ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Computer and Information Sciences

C·O·L

Commonwealth of Learning

INTRODUCTION TO MOBILE ARCHITECTURE

Introduction to Mobile Application Development

1

School of Computer and Information Sciences
Indira Gandhi National Open University

Commonwealth *of* Learning

Introduction to Mobile Architecture

Block

1

INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT

INTRODUCTION TO MOBILE ARCHITECTURE

This course has been developed as part of the collaborative OER Course Development for ICT Skills initiative of the Commonwealth of Learning (COL). COL is an intergovernmental organisation created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

Indira Gandhi National Open University (IGNOU), established by an Act of Parliament of India in 1985, has continuously striven to build an inclusive knowledge society through inclusive education. It has tried to increase the Gross Enrollment Ratio (GER) by offering high-quality teaching through the Open and Distance Learning (ODL) mode.

For the avoidance of doubt, by applying this licence the Commonwealth of Learning does not waive any privileges or immunities from claims that it may be entitled to assert, nor does the Commonwealth of Learning submit itself to the jurisdiction, courts, legal processes or laws of any jurisdiction.The ideas and opinions expressed in this publication are those of the author/s; they are not necessarily those of *Commonwealth of Learning* and do not commit the organisation.

Course Team

Author:

Mr. Shailesh Kumar Shiva
Kumar, Research Scholar
(SOCIS, IGNOU) and Senior
Technology Architect (Infosys
Limited) , Bangalore, India

Content Editor: Dr.P.Venkata Suresh, SOCIS, IGNOU, New Delhi, India

Content Design Team

Dr. P.V.Suresh,
Associate Professor & Director,
SOCIS, IGNOU, New Delhi

Mr. Amit Kumar Saxena,
Chief Technology Officer,
Grasp IO Inc, Bengaluru

Dr. S. Arumuga Perumal,
Associate Professor & Head of the Department, S.T. Hindu College,
Nagercoil, Tamil Nadu

Prof.Parma Nand Astya,
Dean, School of Computing Science & Engineering, Galgotias University,
Greater Noida

Mr. Shashi Bhushan Sharma,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. Akshay Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Dr. Naveen Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. M. P. Mishra,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Dr. Sudhansh Sharma,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Language Editors: Mr. P. Lakshmi Kantham , Faculty in English(Retd.), VSR & NVR College, Tenali, India

CRC Preparation: Tessa Media & Computers, New Delhi

Published by:

Indira Gandhi National Open University, Maidan Garhi, New Delhi-110068, India.

The course has been developed with the support of the Commonwealth of Learning, Canada.

Acknowledgements

Indira Gandhi National Open University wishes to thank those below for their contribution to development of this Course Material:

Staff of School of Computer and Information Sciences, Administration, Finance & Accounts Division and Electronic Media Production Centre, IGNOU, New Delhi, India and all External Experts. Special thanks to Dr.Sanjaya Mishra, Education Specialist(eLearning), Commonwealth of Learning, Canada for his support and guidance through out.

COURSE INTRODUCTION

This course introduces software and hardware architectures of Mobile devices.

The emphasis will be on development of applications for Mobile devices. Not to specially mention, there is a rush for each and every PC / Web based application to be ported to Mobiles. Millions of APPS (Applications) are available in APPSTORE of Apple devices , namely, iPhone and iPad. The same is the case with Android and Windows Store. Hence, Mobile APP Development has become one of the most sought after skills for ICT professionals.

In this course, application development is reviewed for PCs and Web. Its followed by a brief on Mobile based APPS. To develop any application, there is need to have knowledge of underlying architecture. Hence, components of a Mobile application are discussed. Also, any APP designed should consume less resources so that device performance is not adversely affected. Hence, some fundamentals about application design for mobile devices is discussed.

Any APP development will be ultimately based on the underlying operating system also. Hence, Mobile operating systems were also introduced in the course. Separate units are dedicated to basics of Android, iOS, and Windows Mobile. Indeed, everything impacts everything. So, cannot ignore the underlying hardware also. Hence, different types of processors, memories , sensors and I/O devices are introduced.

In the last block of the course, tools to develop applications that are platform specific and platform independent tools are introduced. Also, the topic of monetization was covered in the case of Android, iOS and Windows Mobile APPS

This Course consists of 4 blocks and is organized in the following manner:

Block 1 introduces different types and components of Mobile applications, and basic principles of Mobile Application Design.

Block 2 introduces the most popular Mobile Operating Systems , namely, Android, iOS and Windows Mobile.

Block 3 covers the hardware portion of a mobile device on which the Mobile application development may have impact. It covers different types of processors, memory , sensors as well as I/O devices.

Block 4 covers different types of software development tools for Mobile devices. It includes native software development tools, cross platform development tools , publishing tools as well as issues related to monetization and security.

BLOCK INTRODUCTION

This block introduces the learner to development of Mobile applications (APPS)

Application development is not a new thing and applications are developed for PCs initially, followed for different configurations such as Client/Server and then followed massively for Web. However, the latest phenomenon is APP development for Mobile devices and millions of APPS are already developed and deployed. This block introduces applications based on PC and Web before discussing about Mobile based applications.

Lot of topics are to be studied before starting development of APPS for Mobile devices. Hence, the block covers various components of a Mobile application that includes its architecture. The architecture covers components of Mobile client application as well as Mobile support infrastructure. Then , starts embarking on Mobile application design. As part of it, some issues related to efficiency of Mobile APPS were discussed such as deployment issues, consumption of power etc.

This block consists of three units and is organized as follows:

Unit 1 deals with PC and Web based applications primarily. It also discusses evolution of Mobile based applications.

Unit 2 covers various components of a Mobile Application. It includes components of Mobile client application and support infrastructure.

Unit 3 covers basics of Mobile application design. It includes deployment issues, power consumption, synchronization issues , and patterns and technologies.

UNIT 1 INTRODUCTION TO MOBILE APPLICATIONS

Structure

1.0 INTRODUCTION

As we have witnessed a revolution in the consumer space toward mobility, most analysts have identified that mobile devices are the major gateways to Internet as compared to desktop browsers. Mobile device is replacing all traditional channels to access the information. To align with this trend, enterprises too are designing the digital applications to cater to wide array of mobile devices and platforms.

Mobile application development involves the process of developing the applications for mobile devices such as Personal Digital Assistants (PDA), tablets and smart phones and other mobile devices. Native mobile apps are designed to run on a specific mobile platform, sometimes specific mobile operating system and supported hardware.

Mobile applications are part of main stream digital strategy for Business to Consumer (B2C) enterprises. Most of the enterprises are now adopting "*mobile-first*" strategy wherein the digital applications are designed, developed and tested for mobile devices; mobile users attain the primary focus in the digital strategy. Disruption in mobility space has major impact on the revenues for the enterprises. Mobile apps are shaping user experiences and are providing real-time information and offer more engaging experiences for the users.

Mobility based digital strategy considers various things such as user experience, performance, interactivity, device form factors, device limitations, location needs and personalization.

Key Drivers for Mobile Applications

The following are the key drivers of mobile apps:

- **Innovation** in mobile space such as proliferation of smart phones, higher bandwidths offered by 3G (Third generation) and 4G (Fourth generation) technologies are coupled with higher capacity storage technologies with higher speed chips would keep powering mobile devices.
- **Consumer behavior**: Customers are more used to mobile devices and is easy for them to access information on the move.
- **Personalized content delivery**: Enterprise can leverage the location and sensors to offer more contextualized, relevant and personalized content, offers and advertisements.
- **Mobile ecosystem**: An explosive growth in Mobile Applications stores such as Apple store, Google Play store, Windows marketplace store was coupled with availability of games, utilities and other apps.
- **Social Networking**: With the popularity of web 2.0 and social media technologies such as Facebook, Twitter users are increasingly using the location based features in the social media platforms.

Impact of Mobile Apps on various domains

Mobile apps are impacting various industry verticals and functional domains. Given below are high level changes enabled by mobile apps across industries:

- **Retail and Consumer Packaged Goods (CPG) Industry**: Mobile apps provide location based store locator, targeted promotions/offers/coupons, service reminders, mobile bidding, in-store tools, cross sell/upsell tools and comparator tools. Basically, mobile apps play key role in driving the traffic, increasing the sales and drive the brand loyalty. On B2B front, mobile apps have redefined lead management, CRM functions, efficient tracking, field force automation and such. Mobile apps have also lead to improvement in store merchandize, supply chain and inventory managements.
- **Banking industry**: Mobile apps enable convenient ways to carry out transactions such as account balance, payment, localized alerts, tap-to-pay, branch locator, and payment coupons. Mobile apps would also enable mobile banking, mobile wallet and provide “on-the-go” features.
- **Logistics**: It is easier to track shipments, get updates, manage warehouse, and fleet using mobile apps.
- **Healthcare**: Mobile apps can easily connect patients, doctors and insurance providers as well as provide wellness management solutions.

Besides the above mentioned enterprise scenarios, mobile apps have revolutionized consumer space with various mobile apps related to gaming, utilities, social media, video streaming and many more.

Attributes of Mobile applications

The following are the key attributes of mobile applications:

- **Ubiquity:** Mobile applications are always available and connected and enable users to access information anytime anywhere
- **User friendliness:** Mobile applications provide responsive and interactive user interface with essential information. They utilize the

camera, sensors, media output, touch/multi-touch/voice interface for providing simplified actionable information.

- **Location awareness:** Mobile applications provide location sensitive information using Global Positioning System (GPS) and other sensors.
- **Minimalistic:** The content and features in mobile apps are minimal which are essential for the functionality.

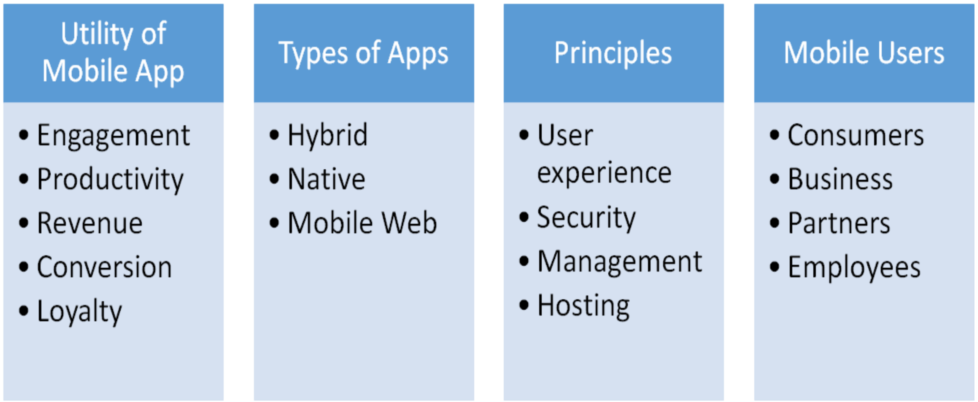
1.1 OBJECTIVES

After going through this unit, you should be able to

- understand key concepts of mobile app development,
- know the opportunities and challenges of mobile apps,
- know the details of PC based apps as well as web based apps,
- know the content and key protocols of mobile apps
- know the evolution of mobile apps, and
- Comparison of mobile apps and web apps and upcoming trends in mobility space

1.2 CONSIDERATIONS AND CHALLENGES FOR MOBILE APP

The main considerations for mobile apps are given in Figure 1.1



Utility of Mobile App	Types of Apps	Principles	Mobile Users
• Engagement • Productivity • Revenue • Conversion • Loyalty	• Hybrid • Native • Mobile Web	• User experience • Security • Management • Hosting	• Consumers • Business • Partners • Employees

Figure 1.1: Mobile App Considerations

The main considerations for mobile app design are listed below:

- Intended utility of the mobile app
  - o Consumer engagement with richer user experience
  - o Productivity through efficient flows
  - o Driving incremental revenue through user stickiness
  - o Customer conversion
  - o User loyalty through targeted and personalized offers
- App Architecture

  - Native vs hybrid vs web based on the requirements
  - Middleware requirement for centralized configuration
  - Offline vs online capability for storing data
- App Development Principles
  - User experience through richer controls and interactive components
  - Compatibility on various devices and platforms
  - Performance for each screen and task
  - Security for data
  - Productivity enhancement tools
- Target users
  - Consumers for B2C apps
  - Business for Business to Business (B2B) apps
  - Partners for B2B apps
  - Employees for Business to Employee (B2E) apps
- Testing
  - Device testing
  - Performance testing
  - Various testing scenarios

The main challenges in mobile app strategy are given below:

- Diversity of devices and heterogeneous technologies: There are various mobile platforms and devices. The app should provide optimal experience in all the scenarios.
- Security: Mobile app should ensure data security during transmission and during storage.
- User experience: Mobile app should provide optimal user experience leveraging the device capabilities to provide highest engagement possible.
- Network: Mobile app should be designed to work in regions with network, latency and bandwidth challenges.
- Compliance to diverse standards, OS, mobile platforms and devices.

1.3 PC BASED APPLICATIONS

Personal Computer (PC) based applications are software programs developed to run on specific operating system and hardware platforms. These were the pioneer applications that were used during the initial days of software development. There are mainly two types of PC based applications, namely, standalone PC applications and client server applications.

1.3.1 Standalone PC Applications

Standalone PC applications are independent software programs which would run on an OS. These applications do not typically use network resources or support multi-user mode. Utility programs such as word processor, calculators, and media players fall into this category. Each of the applications had a good user interface for the PC user to interact.

1.3.2 Client server applications

In client server applications, each terminal PC had a client software which is connected to a centralized server software. The client program would get input from the end user and would submit the details to the server software through a dedicated session established through the network. These applications were also referred to as “thick clients”.

Database software, networked games, banking software, network file system are some of the examples of this category of applications.

☞ **Check Your Progress 1**

1) ……………………. is an attribute related to mobile app availability.

2) …………………………… are independent software programs which would run on an OS.

3) ………………… and ………………….. are two key components of client server applications.

4) Main target users for mobile apps are ………………………. .

5) Native vs hybrid is related to ……………………………….. consideration.

1.4 WEB BASED APPLICATIONS

Internet enabled applications that are mainly rendered on desktop browsers are categorized as web applications. Most of the modern web applications follow layered Model-View-Controller (MVC) architecture which supports loose coupling and flexible modular components.

A typical MVC based web application is depicted in the Figure 1.2:

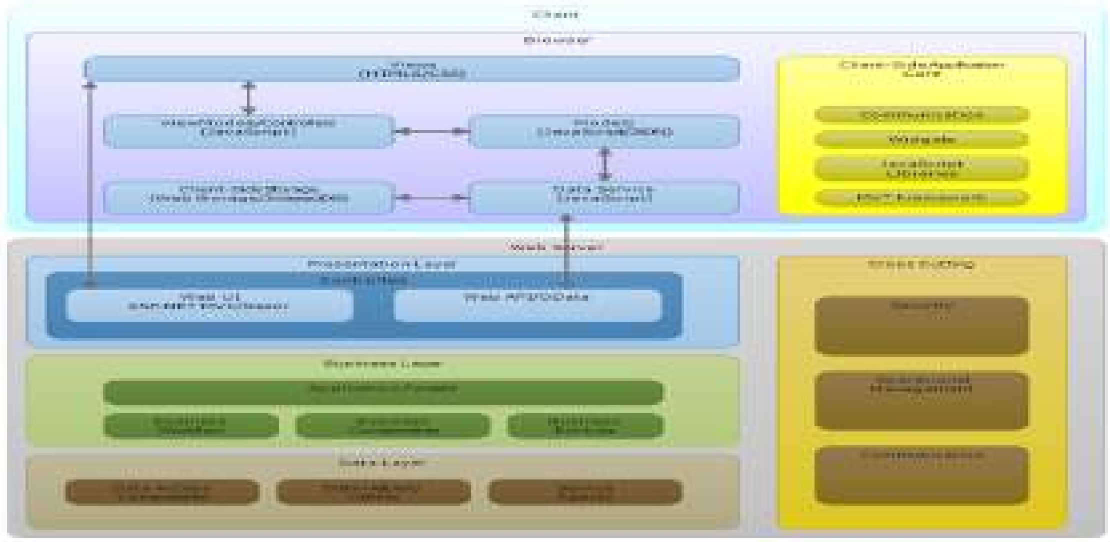


Fig. 1.2: MVC Web Application

The key layers of the MVC framework are shown below:

- **Presentation layer (View layer):** User experience components such as portlets, widgets, pages, User Interface (UI) modules, buttons, and forms are present in this layer. Modern web applications normally use JavaScript components to build the UI modules. The UI modules will mainly render the view portion of the application. *View* components communicate with back end through services. Modern web applications use Representational State Transfer (REST) based light-weight services.
- **Business layer:** This layer consists of business components which implement business logic and business rules. The layer mainly consists of rules engine, search, business objects, workflows, business process management (BPM), caching frameworks and other entities. All business modules expose services to presentation layer.
- **Data layer (model layer)**: This layer mainly consists of persistence handling components such as database access components, Data access objects (DAO), query components, Object Rational Mapping (ORM) frameworks and such.

Besides these layers we also have security components (responsible for authentication and authorization) in security layer and utilities components to handle cross-cutting concerns.

A complex n-tier enterprise web application is depicted in Figure 1.3. We could see various layers for enterprise search, content management, e-commerce with various enterprise interfaces.

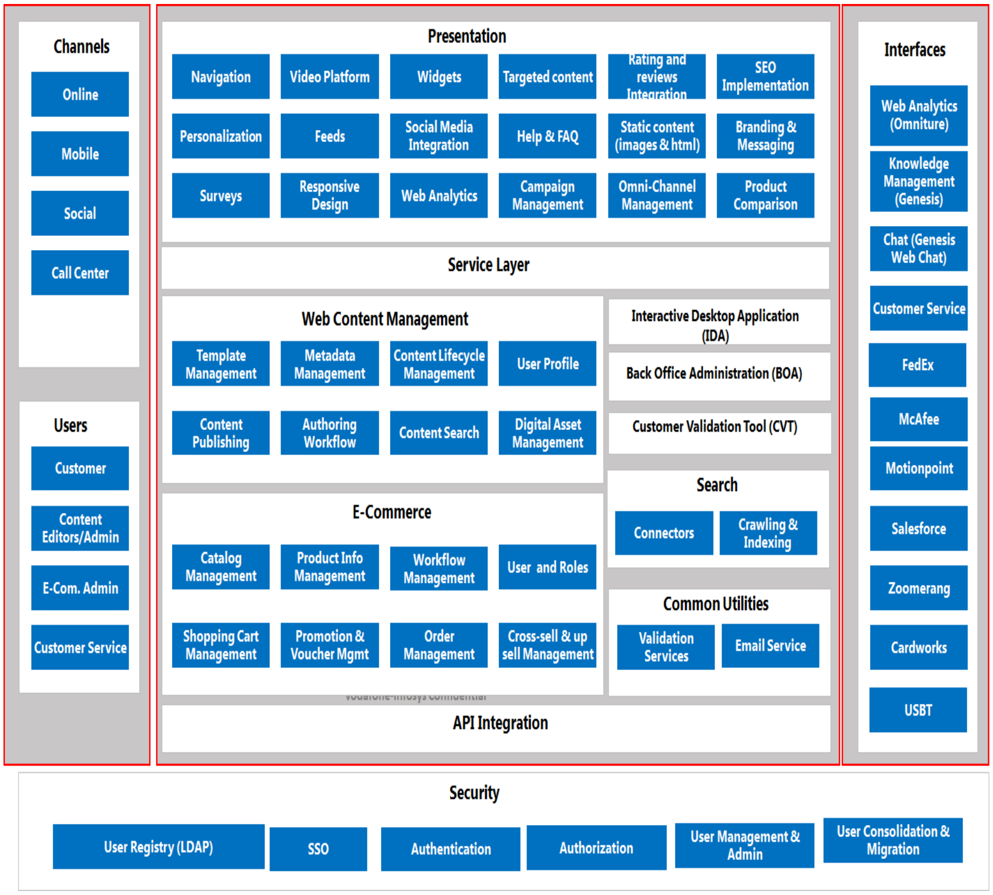


Fig. 1.3: N-tier Enterprise Web Application

The multi-layered architecture is mainly based on MVC architecture:

1.4.1 Presentation Layer

This involves the presentation components like portlets and other user experience components. Key components are explained below:

- **Personalization:** Role-based access and other fine-grained access control to provide personalized user experience.
- **Widgets:** Wherever required jQuery based client-side widgets would be developed to provide client-side functionalities. This would help enhance overall user experience and improve the page performance. Real-time report data display/refresh, pagination, search functionality are typical scenarios where this AJAX-based feature can be employed.
- **Multi-device support:** Responsive design and device recognition features will be leveraged to cater to various mobile devices.
- **Page layouts:** Flexible page layout to cater to various web pages.
- **Information architecture and navigation models:** This consists of context menus, bread crumb, left navigation, site map, site hierarchy for the web site.

1.4.2 Web Content Management Layer

Web content management layer consists of mainly following modules:

- Content authoring using authoring templates.
- Content tagging with relevant metadata and tags.
- Content publishing to various targets and in various formats.
- Asset management of various digital assets, documents and multimedia files.
- Workflow for managing the content approval, publishing and update processes.

1.4.3 E-commerce Layer

Ecommerce modules usually consist of modules related to catalog management, order management, modules for shopping cart, promotion, cross sell and up sell, and product information management.

1.4.4 Integration Layer/Services Layer

The solutions use business service layer for integration with external system. Integration strategy is based on Service Bus Architecture, in which the middleware can act as a service bus linking multiple applications that require services of each other through a central service layer. The service bus becomes a point of data interchange and manages the communication with each peripheral application independently. All service invocations will be done through ESB layer. JMS component will be developed to send and receive the messages to the ESB message destination. Application/System services will be developed and exposed to the ESB layer.

1.4.5 Security Layer

Security layer consists of modules related to authentication, authorization and single sign on.

1.5 EVOLUTION OF MOBILE BASED APPLICATIONS

A brief overview of various stages of mobile app development is depicted in Figure 1. 4.

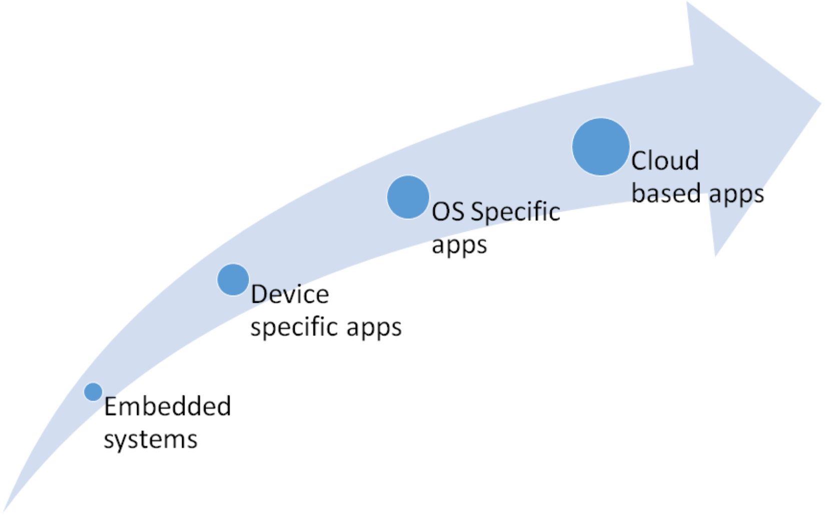


Fig. 1.4: Evolution of mobile app

The core features of the various stages of mobile app evolution are detailed below:

- **Embedded systems:** In this category, we has embedded systems such as calculators on various devices. The embedded systems had limited functionality.
- **Device specific apps:** The applications specific to the device are part of this category. For instance, device specific games and other utility apps fall into this category.
- **OS specific apps**: Operating system specific apps such as games, media players fall into this category.
- **Cloud based apps**: In this category, we have always available cloud based apps.

1.6 COMPARISON OF A MOBILE APP WITH A WEB APPLICATION

A high level comparison of mobile app with web application is given in following table1.1:

Table 1.1 : Mobile App Vs Web Application

Criteria	**Web Application**	**Mobile App**
User experience	Provides good experience optimized for desktop browsers.	Mobile apps can leverage full device capabilities and offer rich experience to users. Mobile apps provide rich branding experience to users.
Performance	Web applications provide good performance based on performance optimizations.	Native mobile apps provide high performance.
Location awareness	Web applications provide relatively less location awareness	Mobile apps provide location sensitive and contextual information
Development cost	We can have single code base and hence relatively lesser development and maintenance cost.	We need to potentially have multiple code bases to cater to various mobile platforms and hence relatively higher maintenance cost.
Access mechanism	Web applications are mainly accessed by desktop browsers and mobile browsers.	Mobile apps are accessed by mobile browsers and mobile devices.
User intuitive features	Web applications use limited amount of device features.	Mobile apps provide intuitive features using device's camera, sensors, GPS etc. and provide notifications to users.
Interactivity	Web applications provide interactive interfaces through widgets.	Mobile apps offer high level interactivity through touch interface.
Applicability	Web applications are normally used as Information display platforms	Gaming, location-specific applications such as car rental apps, store locator apps, reporting apps.
Personalization	Web applications offer personalization features through server side preferences.	Mobile apps provide high degree of personalization through various context parameters such as location, history etc.
Common use cases	News, blogs.	Games, social media and location related services.

1.7 CONTENT AND PROTOCOL IN MOBILITY

The content for mobile apps are designed to be adaptive. The adaptive content uses rules to adapt itself based on the context. Context includes various parameters such as device, location, time, form factor, mobile platform, screen resolution, personalization parameters, preferences and such. Adaptive content hence provides a superior information delivery experience for mobile users.

Let us look at couple of examples of adaptive content in the context of mobile apps. For a regular product page, we would be having product long description and product short description. If the product content is designed to be adaptive, the long description would be rendered on desktop browsers and tables which have higher page real estate and only short description will be delivered on smart phone with lesser screen layout. Similarly, role based content filtering, location and time based content filtering can be done using adaptive rules.

Responsive design mainly caters to the flexible page layouts and screens. A combination of responsive page design and adaptive content would be idea for mobile scenarios.

The main protocols used in mobile apps are as follows;

- **Wireless Application Protocol (WAP)**: It provides specifications and rules for wireless communication devices such as smart phones. WAP is optimized for low-memory , low-bandwidth mobile devices.
- **Representational State Transfer (REST)**: Responsive mobile web applications communicate with server using light-weight REST service calls. Normally, JSON data is used in REST service for data exchange.

1.8 TRENDS IN MOBILITY SPACE

The following are some of the trends in mobility space:

- HTML 5 based responsive apps: HTML 5 technologies would help developers to develop pure responsive web applications that cater to all mobile devices.
- Bring your own device (BYOD): Employees can bring their own mobile devices to work and work simplification.
- Internet of Things (IoT): Provides real-time data through sensors and other connected devices to the mobile. *Wearable* is increasingly gaining traction and the data published from wearable will be consumed by mobile devices.
- Mobile device management (MDM): Managing configuration, security and policies of various devices.
- Mobile application management (MAM): Managing version, delivery and provision of various apps.
- Increased variety of communications such as touch, voice, facial/gesture recognition, video, scanning, Near Field communications etc.
- Location based real-time alerts and notifications and other innovative features such as route optimization.
- Mobility as a service (MaaS): MaaS is to accelerate time to market and reduce complexity.

1.9 BRIEF NOTE ON MOBILE APP PLATFORMS

Let us look at two popular mobile app development platforms: iOS and Android.

Android

Android is an open-source mobile development platform that is based on Java and is maintained by Google. The key features supported by Android are SQLite based light-weight storage, SMS and MMS messaging, multi-lingual support, mobile browser. Other key features are multi-touch support, multi-tasking, voice features, external storage and such.

Android development needs Android SDK, libraries, emulator and Eclipse IDE. Testing can be done using Android testing APIs. We can deploy Android apps to Google Play store.

Apple iOS

iOS is the mobile OS for Apple devices. The development on iOS happens using objective C. iOS supports many features such as iMessage, iCloud, Siri etc. iOS provides in-built apps such as mail, notifications, contacts calendar, bookmarks, sync etc.

iOS development includes iOS SDK (XCode), iOS Simulator, XCode IDE, and other frameworks for building iOS apps.

For mobile web development, we could use numerous JavaScript frameworks such as SenchaTouch, jQuery Mobile, jQTouch and software such as PhoneGap that are used for mobile web development.

☞ **Check Your Progress 2**

1) …………………………….. Layer is responsible for presentation components in MVC architecture.

2) ………………………….. Layer is concerned with centralized services access.

3) Light weight service invocation in mobile apps can be done through …………………………. .

4) ………………………… is popular open source mobile platform.

1.10 SUMMARY

In this unit, we started discussing the main considerations and challenges in mobile app development. We then discussed the PC based applications including stand alone and client server applications. We then had a detailed discussion on web based application which is designed using MVC architecture. We also looked at various stages of mobile app development. We compared the mobile apps with web applications and looked at content and protocols of mobile apps. We summarized brief trends and overview of key mobile platforms.

1.11 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) Ubiquity.
2) Standalone PC applications
3) Client terminal and Server software
4) Consumers, business and employees
5) App architecture

Check Your Progress 2

1) View layer
2) Integration/services layer
3) REST calls
4) Android

1.12 FURTHER READINGS

References

- https://developer.apple.com/
- http://developer.android.com/
- https://play.google.com
- http://phonegap.com/
- http://jquerymobile.com/
- http://www.sencha.com/
- http://www.wirelessweek.com/articles/2011/03/15-trends-matter
- http://www.wirelessweek.com/articles/2009/11/innovation-fragmentation
- http://www.androidauthority.com/android-vs-blackberry-pros-and-cons-31084/
- http://bgr.com/2013/09/19/mobile-app-store-downloads-102-billion-2013/
- http://www.wired.com/images_blogs/gadgetlab/2010/04/android-os-marketshare2.jpg
- http://www.idc.com/getdoc.jsp?containerId=prUS23946013
- http://www.webpronews.com/ios-android-app-store-android-market-millennial-media-mobile-mi-2011-04

UNIT 2 COMPONENTS OF A MOBILE APPLICATION

Structure

2.0 INTRODUCTION

Mobile app consists of various client side components and support infrastructure components. It is important to understand various components of mobile app to design and develop an effective mobile app. The knowledge about the architecture for various kinds of mobile app and their applicability would guide us to select the most appropriate mobile app for a given scenario.

Mobile app architecture also provides a perspective about the user interface design, integration aspects and other such considerations.

In this unit, we closely look at various components of mobile app and have a deep dive study of architecture of various mobile apps. At the end we will also look an end-to-end case study.

2.1 OBJECTIVES

After going through this unit, you should be able to:

- understand key concepts of mobile app architecture,
- know the details of architecture of native mobile app,
- know the details of architecture of hybrid mobile app,
- know the details of architecture of mobile web app,
- know the details of various client side and support infrastructure components of mobile app and
- look at an end to end case study of an Android app.

2.2 ARCHITECTURE OF A MOBILE APPLICATION

Mobile applications can be classified into mainly 3 categories:

- **Browser based mobile web apps**: These are pure web applications that are designed using responsive web design techniques that can cater to variety of devices and form factors.
- **Native mobile apps**: These apps are built for specific mobile platforms (such as iOS, Android) that can fully leverage the device capability. Normally native mobile apps are built using SDKs provided by mobile platforms.
- **Hybrid apps**: These apps can be developed using web technologies such as JavaScript and can also partially leverage the device capabilities using a web to native layer.

Table 2.1: Provides the key differences of three types of mobile app architectures

	Browser based responsive mobile web	**Native apps**	**Hybrid apps**
Development technology	Usually uses responsive web design (RWD) using CSS3, media queries	Native apps are developed using iOS, Android and other mobile platforms	Developed using a single technology (usually JavaScript) and can be ported to any native technology
Features	Only used for responsive and interactive UI. Cannot fully use all mobile device features. Provides Best Portability, Time to Market, maintainability	Can fully exploit all device capabilities such as camera, sensor, contact books and such. Provides best user experience	Limited usage of device native features
Applicability	Can be used for information display without any offline support and for hosted solutions	Can be used for features needing rich user experience, user engagement such as games. Can be used for offline requirements and for features needing device specific features	Can be used for easier portability across mobile platforms and cross-platform compatibility. We could use this for faster time to market scenarios with frequent releases.
Compatibility	Works across mobile devices and platforms. Offline not supported.	Native apps run on specific mobile platforms; offline capabilities supported	Hybrid apps can be ported to various mobile platforms

Development	Fast development and maintenance	Higher cost	Lower development cost due to portability. Build once deploy to any platform model.
Advantages	Standards compliance, lesser development cost.	High performance, rich UI, can exploit native mobile features,	Easy portability
Performance	Relatively lower performance	Optimal performance	Relatively lower performance
User experience	Provides optimal experience for web experience	Provides excellent UI features leveraging the device capabilities to the fullest extent	Provides decent UI experience.
Hosting	Cloud or on premise platforms	App stores or marketplace	App stores or marketplace

2.2.1 Architecture of a native Mobile App

Native Mobile app fully utilizes the device capability. Figure 2.1 and Figure 2.2 depict architectures for iOS and Android native apps:

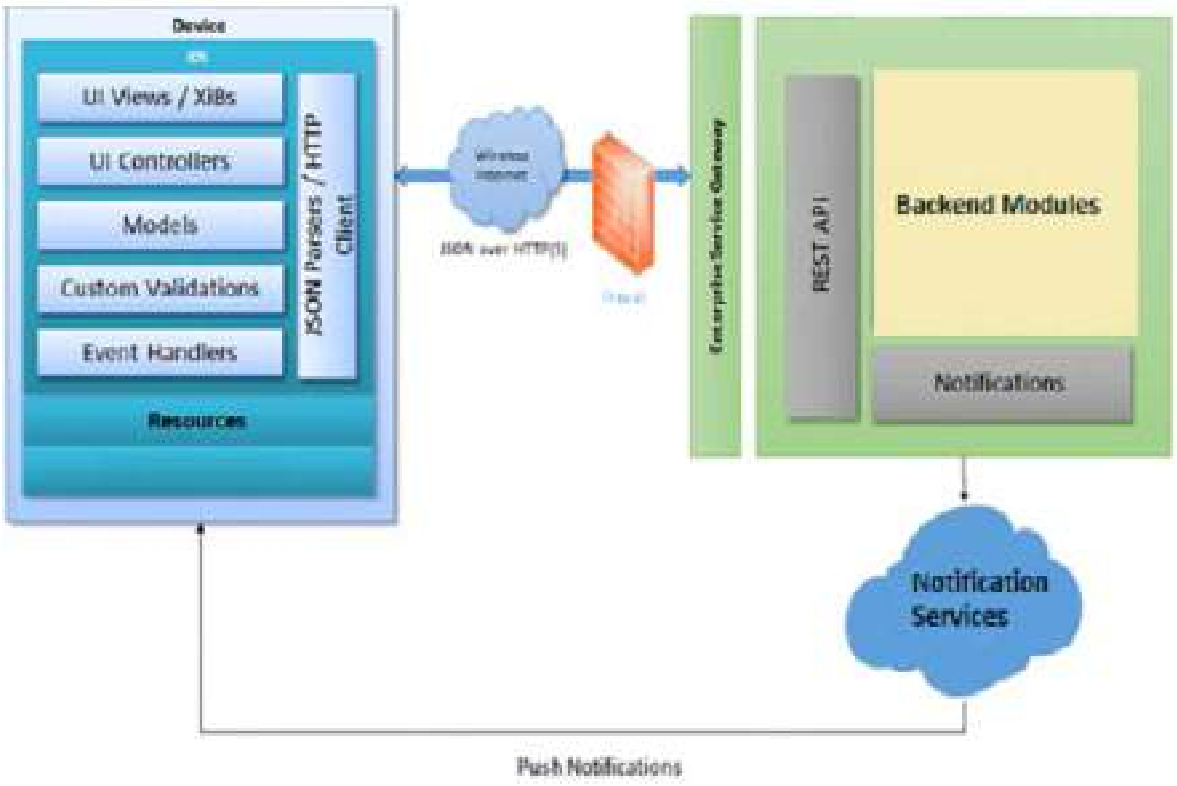


Fig. 2.1: iOS based native Mobile App

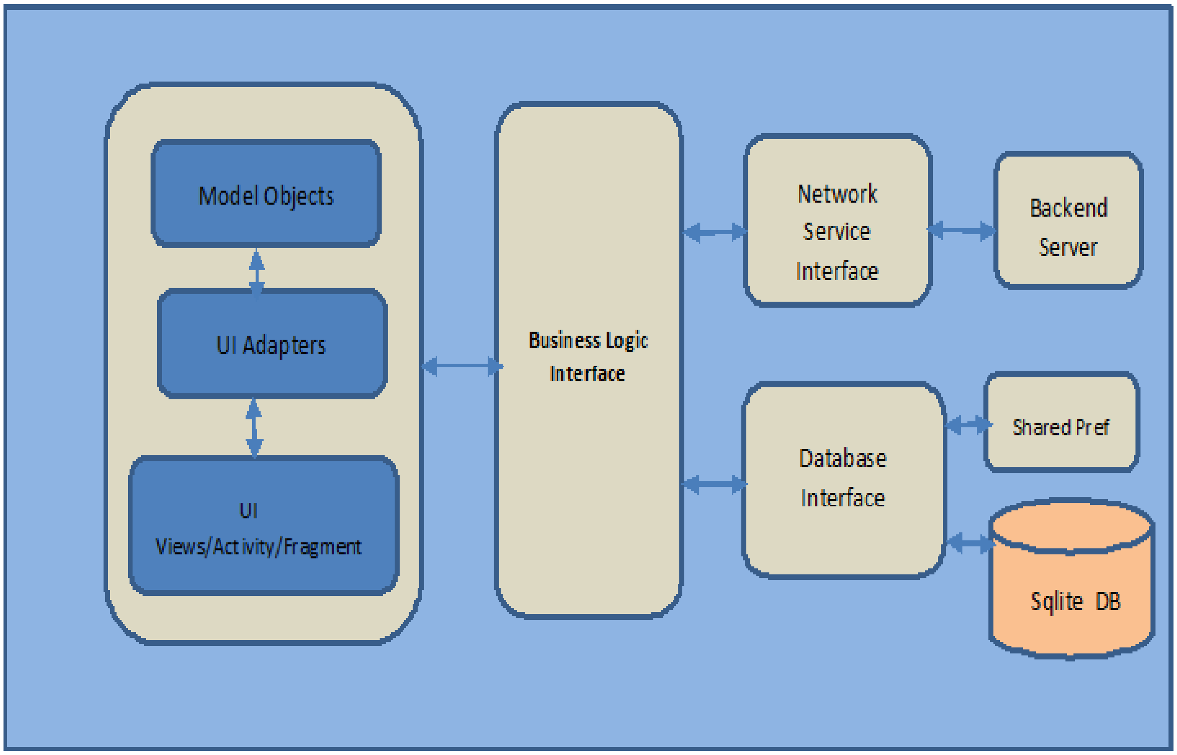


Fig. 2.2: Architecture of Android based native Mobile App

2.2.2 Architecture of a hybrid Mobile App

Hybrid application targets different mobile OS with a very thin application shell with a web view. One time application development is needed with publication to various mobile app stores. Hybrid apps are very popular in developing Single Page Application responsive application for Tablets and Smart Phones.

The key components of the hybrid app are given in Figure 2.3. The Mobile application development using Hybrid app development has the following benefits:

- **Usability:** Ability to use device specific features to improve usability
- **Maintainability:** Easy maintenance of code and ease of future enhancements
- **Extensibility:** Support for multiple platforms
- **Device Diversity**: No additional effort for supporting new devices
- **Portability:** to various mobile platforms
- **Faster time to market:** through quicker deployment to mobile app stores.

We can have mobile specific services which consolidate the required backend services and provides optimized data for the mobile application.

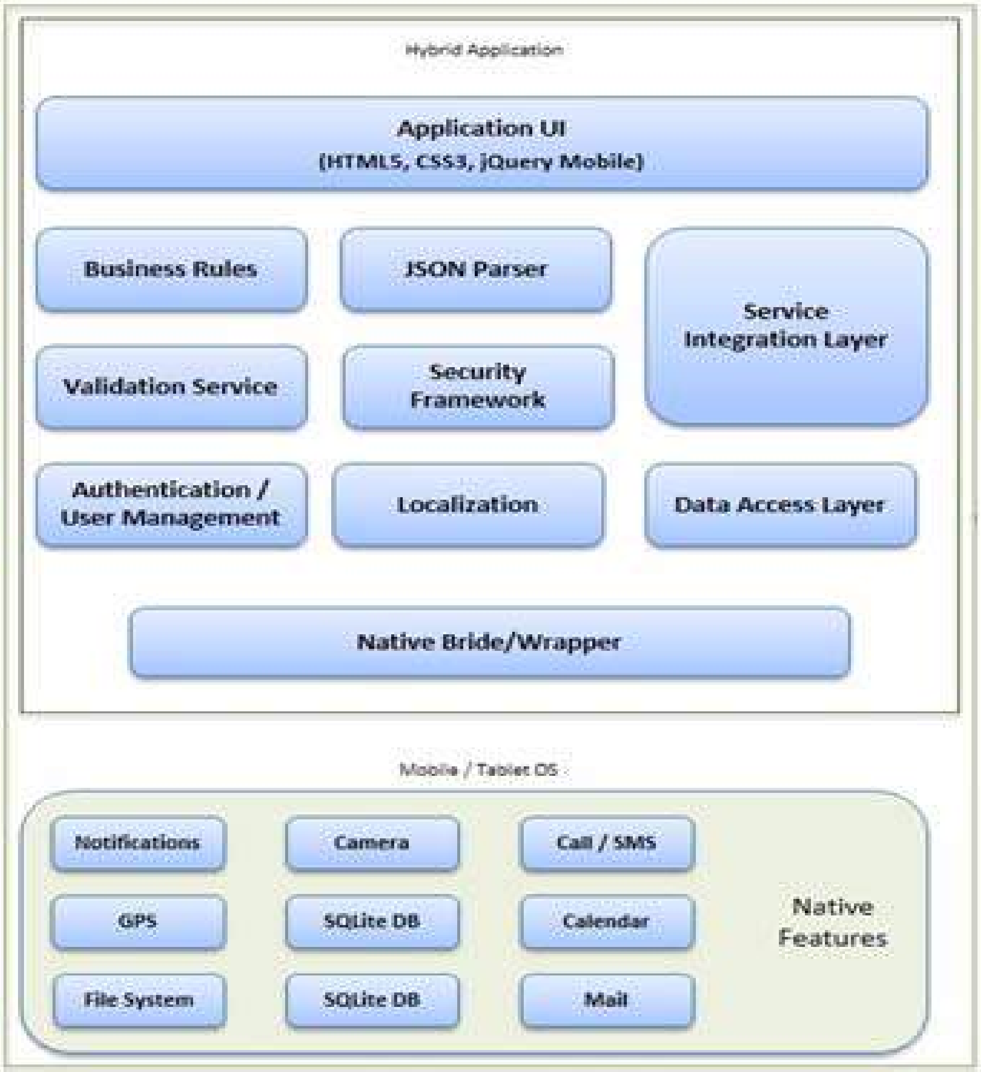


Fig. 2.3: Architecture of a hybrid Mobile App

Figure 2.3 is an expanded view of the mobility solution. It details out the functionalities and the interfaces with the underlying components as given below:

Application UI: All the screens would be developed using HTML5, CSS3 and JavaScript's. User experience would be specifically designed for Mobile Apps with targeted platform compatibility.

Business Rules: are operational procedures for making decisions, identifying workflows and operational choices.

Service Integration Layer: Integration of all the business services with the backend system would be managed by this layer.

Security Framework: All the data persisted within the device will be stored securely using the encryption techniques specific to each platform. Random generated unique key will be used for encryption and decryption. The key used to encrypt will persist in the device securely. Advanced Encryption Standard (AES) is the algorithm used for security purpose. The data sent through the network will be specific to the backend service.

Authentication / User Management: The user specific data management and the business logics will be handled by this layer. The authentication will be done in online / offline mode.

Localization: This is the layer which will provide the multi-lingual capability to the app. All the static literals specific to each language will be maintained locally inside the app. The literals specific to the dynamic content would be fetched from the backend services.

Data Access Layer: All the data stored locally would be accessed through this layer. This consists of data objects and the local storage. SQLite would be used as local storage for the application.

Native Bride/Wrapper - This layer will provide the capability to access the devices native features. This is the combination of the mobile core platform and JavaScript.

The end to end flow of the hybrid mobile app is shown in Figure 2.4 and the corresponding layers explained.

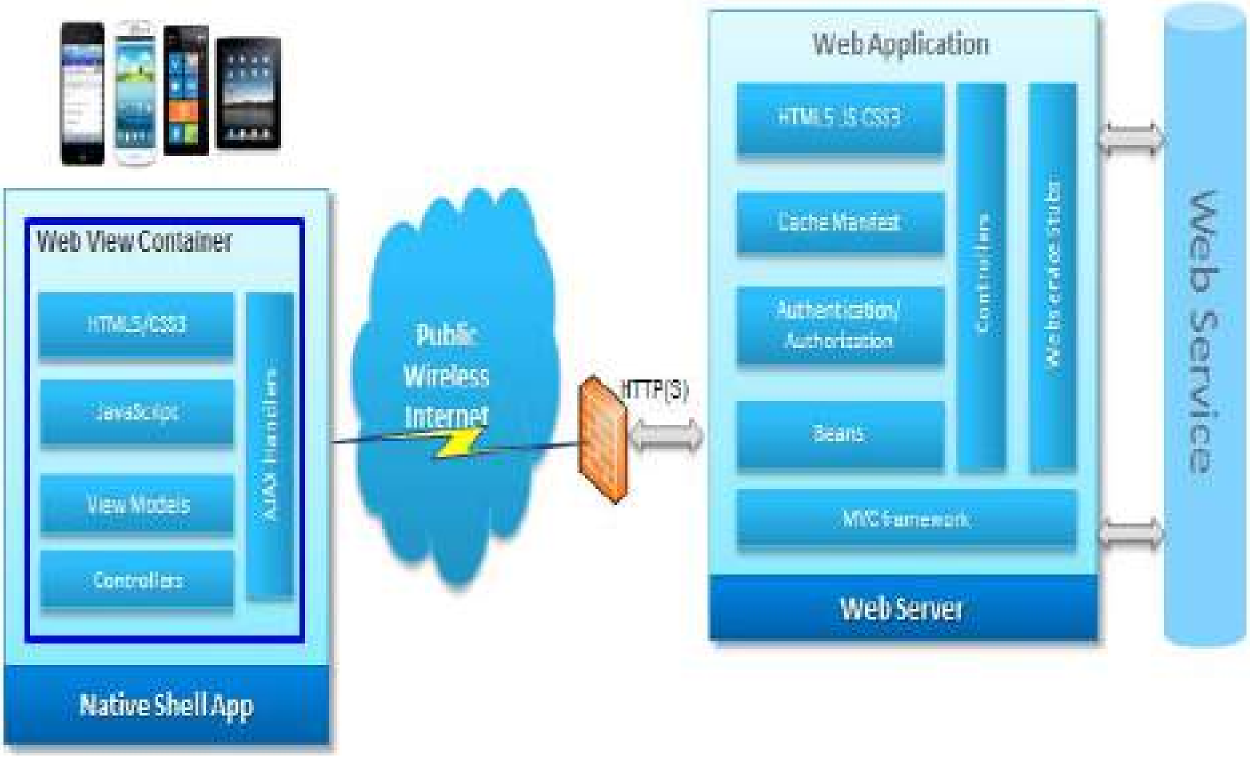


Fig. 2.4: End to End Flow of a hybrid Mobile App

Presentation Layer

- UI will follow a Responsive Web Design (RWD) and will be developed using HTML5.0, CSS3, Twitter Bootstrap with JQuery
- The UI can be packaged using Apache Cordova to create a Hybrid App. REST/JSON Web services will be leveraged.
- Hybrid apps will have two components, the native component and the web component.

Business Layer

- REST/JSON Web services will be leveraged to access server side applications.
- Business layer consists of server side components for Services Integration and other custom business services.

Security Layer

- Authorization to the app user will be leveraged with user registry services.

2.2.3 Architecture of a Mobile Web App

Mobile web apps are mainly developed using responsive design. RWD provides flexible and responsive layouts that automatically adjust themselves for various devices and form factors which include the following techniques:

Fluid Design

contains the page size in relative units.

Flexible image and media

Flexible images would automatically scale based on the screen resolution.

CSS3 media queries

would provide flexible layouts and use various device dependent CSS style rules.

Figure 2.5 shows the responsive layout displayed over multiple devices.

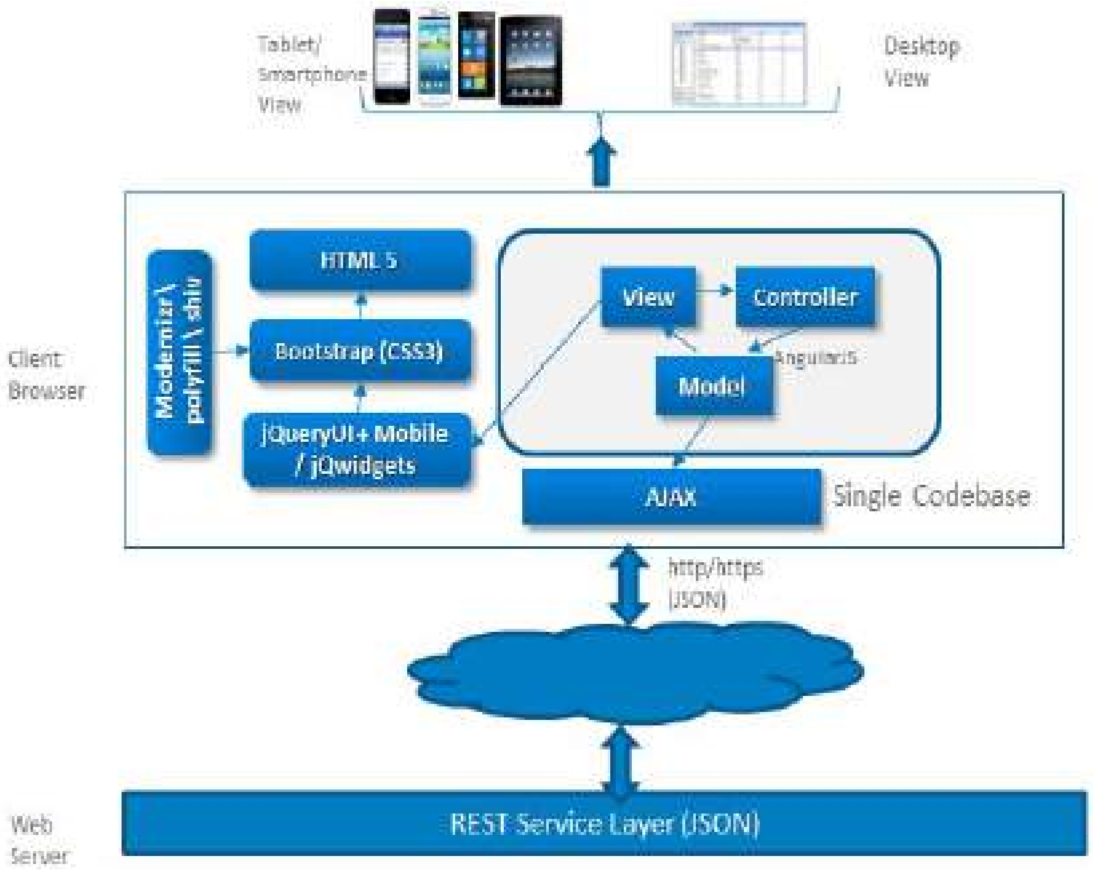


Fig. 2.5: Mobile web app using RWD

The salient features of RWD are given below:

- User Experience - a responsive website is flexible and will adapt screen layout to provide optimal user experience.
- Cost Effective - only one version of the source code and only one content management system to update the content which means that RWD will save both time and money.
- Improved SEO - The URL structure will remain the same on all devices, improving your search engine visibility and rankings. Instead of building links or optimizing content for multiple websites, you will only need to market a single responsive website.
- Increased Conversions - The potential customer can access your website with ease from their preferred device enhancing user experience that will increase sales and improve conversion rates.

The key architectural principles followed while designing a mobile web app are as follows:

- Open standards based technology.
- Layered architecture using MVC pattern
- Modular and extensible component design
- Adoption of services oriented architecture for integration
- Leveraging open-source technologies wherever applicable
- Performance based design
- Continuous build and integration approach for execution

A sample mobile web app architecture with responsive components in a MVC architecture is depicted in Figure 2.6:

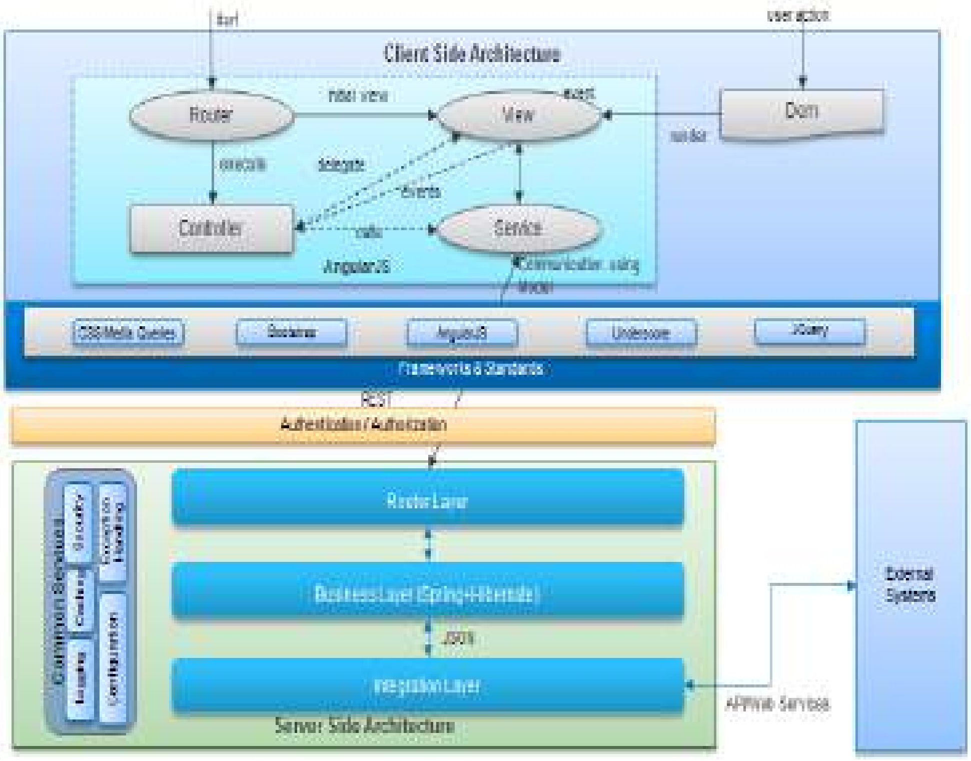


Fig. 2.6: MVC architecture of mobile web app

The MVC architecture depicted above comprise of building server side and client side components where client components will have presentation logic based on MVC design pattern using approved JS (Java Script) libraries whereas server layer will have all the business logic which will be exposed through RESTful services. Client components will interact with server side components using REST services to fetch and persist the data.

Client Layer

- **MVC**: Client-side MVC framework (Angular JS) will be used to structure the application, which will provide:
  - routing support within the application to allow navigation in application;
  - event-driven interaction between views and model;
  - Simplified CRUD (Create, Read, Update and Delete) invocations on RESTful services.

Server Layer

- A router layer maps the URL with a service end point. This layer will be an interface with client side. These REST services will pass the request parameters to business layer.
- Business layer will validate the input data, perform any data manipulation or business logic implementation on the data retrieved from the Integration layer.
- Integration layer will basically integrate with the external system which will be exposing APIs/web services. Data communication will take place using JSON (Java Script Object Notation) format.

Roles of various layers are depicted in Table 2.2:

Table 2.2: Layers in a Web Mobile App

Layer	Client/Server	Description
Router	Client side	Linking URLs to controllers and views.
View	Client side	Consists of Angular directives, templates, CSS and JS files.
Controller	Client side	Controller ties the view with back end data.
Services	Client side	Services interact with back end data.
Router	Server Side	Router maps the URL with service end point.
Business	Server Side	Takes care of business rules, validations, data transformation and such requirements.
Integration	Server Side	Integrates with external system which will be exposing APIs/web services.

A sample technology stack for implementing the mobile web app is depicted in Table 2.3:

Table 2.3: Technology Stock for Mobile Web App

Technology to be used	**Usage side**	**Description**	**Benefits**
AngularJS	Client side	MVC based JavaScript framework	Introduced clean MVC design pattern for client side web applications Built-in dependency injection subsystem Extensible using directives
Bootstrap	Client side	HTML and CSS based front end framework	Makes front-end web development faster and easier
HTML/CSS	Client side	Standard technologies for web development	
Spring	Server side	REST API backend built using Spring or Node.js	Extensive and Scalable framework
REST Services	Server side	Web services based on REST (http/https) URLs	Lightweight, maintainable, and scalable

2.3 COMPONENTS OF MOBILE CLIENT APPLICATION

The following are key components of mobile client application:

- UI components such as widgets, buttons, screens and navigation components
- JavaScript libraries in case of mobile web applications
- Widget libraries to manage the client-side widgets
- UI controllers for client side MVC frameworks
- Validators and event handlers for handling interrupts and notifications
- Model components for handling the data
- Service interface layer to access services
- Light weight database to persist the user data.

2.4 COMPONENTS OF MOBILE SUPPORT INFRASTRUCTURE

The following are main components of mobile support infrastructure are:

- Mobile device management (MDM) is used for configuration and management of mobile devices. MDM is mainly used for distribution of mobile apps, enforce policies. Other key functions of MDM software are:
    - Asset management and device grouping
    - Remote software management
    - Configuration management
    - Performance diagnostics and health check of devices including memory, battery, network information with reporting capabilities
    - Restore and backup of device data such as calendar, contacts, notes and such.
    - Application and service provisioning
    - Firmware upgrades
    - Logging and reporting
    - Troubleshooting
- Mobile security management for enforcing security policies
- Mobile middleware for acting as adaptors and to expose services. Mobile middleware components are also responsible for data transformation, routing, caching, governance.
- Other mobile infrastructure elements are:
    - Spectrums for 3G and 4G

☞ **Check Your Progress 1**

1) …………………………. Apps are normally built using SDK.

2) Mobile web apps pre-dominantly use ……………….. technique to cater to multiple browsers and form factors.

3) ……………………… in Android app makes backend service.

4) Most commonly used light weight DB in a mobile app is ……………

5) In MVC framework, ………………… glues view data with back end data.

2.5 END TO END CASE STUDY OF AN ANDROID MOBILE APP

The following is a detailed case study of an end to end flow of an Android mobile app for retail domain. The detailed case study would help us to understand various components and integration mechanism for building a native mobile app. The high level architecture of Android App is given in Figure 2.7.

The following are key requirements for the app:

- App should scan the bar code for a retail solution and provide mapping support

- App should support multiple languages
- App should support data encryption during transmission
- App should support synchronization features.

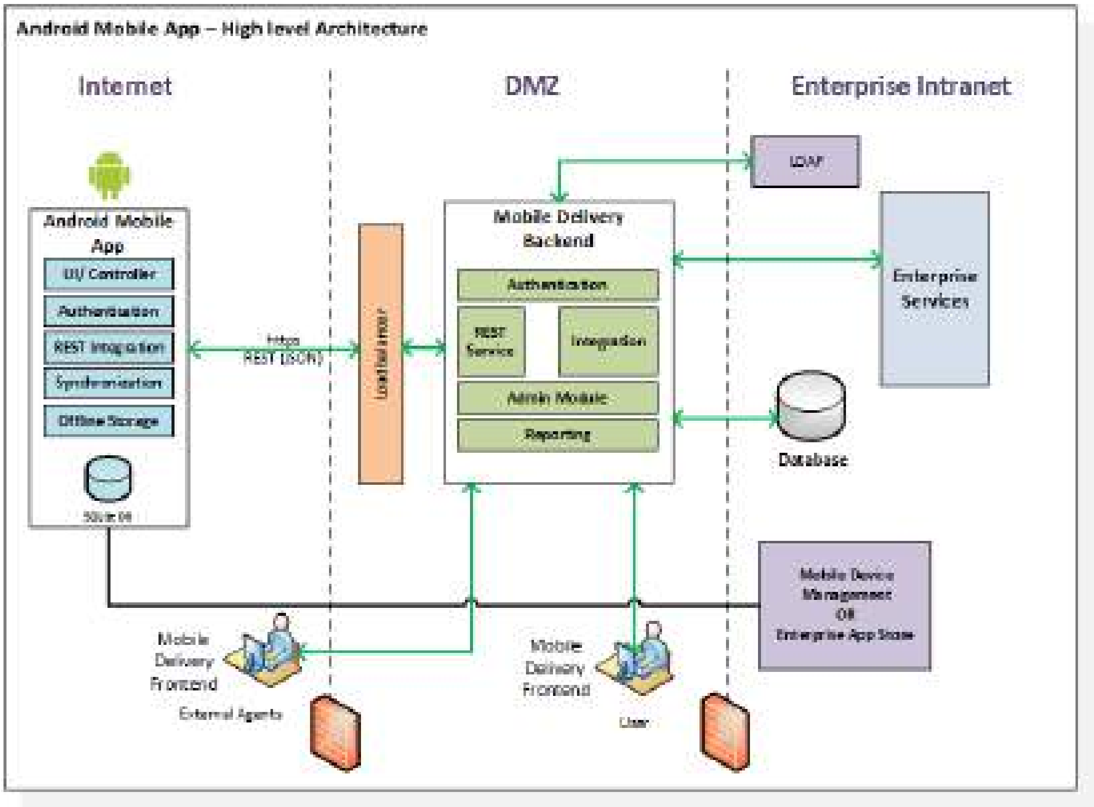


Fig. 3.7: Android App High level architecture

Different components of Android App are described below:

Client Tier

- Native Android application– to work on Android Phones
- Solution uses encrypted storage to securely store sensitive data in device. Data can be stored in the SQLite data store for access in offline mode.
- Application supports multi-language screens. The application will be designed to use language strings that will be added in the properties files. Depending on the logged-in user’s device language, the appropriate language string file is loaded.
- Application leverages the device camera to scan the barcodes.
- Application will leverage native device libraries to support features like Tracking by providing GPS coordinates in latitude/longitude format
- Application will use standard google map APIs for all map specific workflows

Client Integrations

- Application authentication would be done against the Mobile Backend system.
- Application would integrate with the RESTful services exposed by the Mobile Delivery Backend system. Data transmission format will be JSON.

- The communication channel will be secured by using https for all data transmission
- The application uses offline data storage. The application expects the backend system to provide delta data retrieval (for offline sync). The delta data retrieval is needed to reduce the amount of data transmitted across network.
- The data synchronization would be implemented as a manually initiated sync. This is to provide more control to the user and to prevent periodic polling to the backend (which would increase battery usage on the device).
  - o Last synchronized time and number of entities modified on the device which are pending to be synchronized with the backend may be displayed in the header. This will provide a better view on the data freshness to the users.

Figure 2.8 depicts various components in the backend of Mobile App.

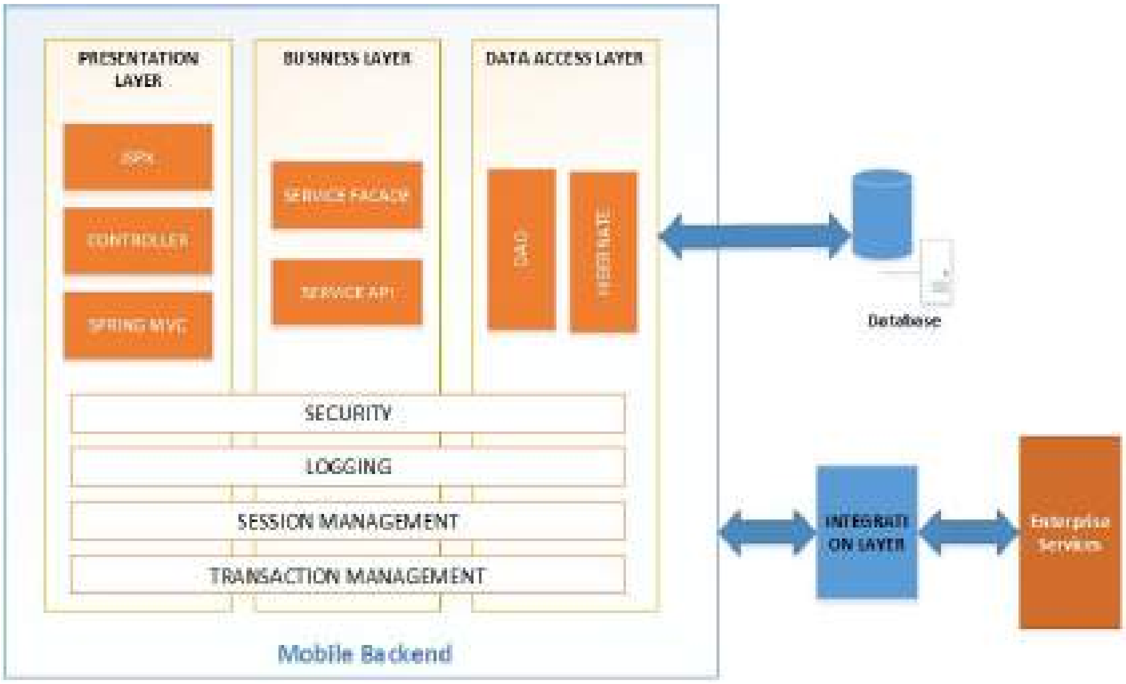


Fig. 2.8: Mobile backend components

The components are described below:

- JSPX web pages developed for Mobile Platform
- Spring MVC for attaining loose coupling between components
- Business Layer that comprises of “Service Façade” and “Service Implementation” patterns
- Data Access Layer would comprises of DAO (Data Access Objects) and would connect to database using Hibernate
- Spring Framework would be used to cater for Logging, Security, Session Management and Transaction Management
- Integration Layer would provide interface with the Enterprise services
- REST services would be exposed to Mobile client Application

☞ **Check Your Progress 2**

1) The communication channel will be secured by using …………………… for mobile app.

2) Offline data sync can happen through ……………….. data retrieval.

3) ………………….. is used for configuration and management of mobile devices.

2.6 SUMMARY

In this unit, we started discussing the high level architecture of various kinds of mobile apps. We then detailed layer-wise components of native moble app, responsive mobile web app and hybrid app. We looked at various components of mobile client application and support infrastructure. We also detailed the roles and technology stack for responsive web app. We saw an end to end case study for Android app.

2.7 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) Native Mobile Web

2) RWD

3) NetworkService

4) SqLite DB

5) Controller

Check Your Progress 2

1) HTTPS

2) Delta

3) MDM

2.8 FURTHER READINGS

References

http://en.wikipedia.org/wiki/Mobile_security

http://csrc.nist.gov/publications/drafts/800-124r1/draft_sp800-124-rev1.pdf

https://en.wikipedia.org/wiki/Mobile_architecture

https://en.wikipedia.org/wiki/Mobile_asset_management

https://en.wikipedia.org/wiki/Mobile_device_management

https://en.wikipedia.org/wiki/Secure_Mobile_Architecture

UNIT 3 BASICS OF MOBILE APPLICATION DESIGN

Structure

3.0 INTRODUCTION

Designing an impactful mobile app involves various aspects. The designer should consider various elements such as user interface, design best practices, optimal integration methodologies, patterns, security, etc. Successful roll out of a mobile app also involves effective testing and robust deployment practices.

In this chapter, we will look at key design best practices, deep dive of user interface design, security aspects and testing elements of a mobile app.

We also provide a checklist for mobile apps and touch base upon power usage and synchronization aspects of a mobile app.

3.1 OBJECTIVES

After going through this unit, you should be able to:

- understand key design considerations and best practices of a mobile app,
- know the anti-patterns of mobile app design,
- deeper understanding of user interface design of mobile app,
- know the deployment and power usage scenarios,
- Know the security standards, and
- Know various kinds of testing

3.2 DESIGN CONSIDERATIONS AND BEST PRACTICES

The key design considerations for mobile applications are as follows:

- Firstly, decide on the type of the mobile app – web, native or hybrid.
- Design the mobile application considering various form factors, screen sizes, orientations and resolutions. Test for all supported devices.
- Design the code to use the device memory, battery and storage optimally.
- Provide mobile friendly controls, navigation and touch enabled actions.
- Provide support for multiple languages and font sizes.
- Design fast, responsive and interactive page layouts.

The following are the key best practices for designing mobile applications:

- Design applications to handle lost network using offline features.
- Test the mobile applications for slow performance scenarios and handle the transactions gracefully.
- Use the least restrictive security model and provide permissions to the mobile apps only when required and when it is permitted by the user.
- Utilize notification features, events, messaging and progress bars wherever necessary.
- Mandatorily encrypt data during transmission.
- Mobile app should leverage offline caching and local caching for rich and interactive apps.
- Given below are the user experience related best practices:
  a) Provide visual feedback using progress bar or steps during processing workflow steps.
  b) Do not provide deep menus. Minimize the depth of menus. Keep simple navigation and minimal page depth.
  c) Provide intuitive UI which is self-explanatory and easy to use.
  d) User should be able to reach the required information with minimal clicks.
  e) Use the points at http://www.w3.org/TR/mobile-bp/ as a checklist for mobile web applications.

3.2.1 Anti Patterns

Given below are some of the common anti-patterns that should be avoided in mobile apps:

- Cluttered information on pages.
- Too heavy and deep page hierarchies with complex navigation.
- Invoking too many services for a given page.
- Not testing mobile apps on all supported devices and platforms.

3.3 CHECKLIST FOR MOBILE APPS

Given below is the checklist that can be used during mobile app development:

- Ensure availability of controls and call to action buttons on all screens.
- Ensure that app handles the crash in a graceful fashion.
- Check for all validations (maximum length, type checking, minimum length etc.) for all forms of input fields.
- Ensure that all kinds of testing is completed to validate the conformance to requirements.
- Check for all alignment, resizing and behavior of UI elements on all supported mobile platforms and browsers.
- Ensure none of the links and screens are broken.
- Ensure that there is no accidental information leakage during exceptions and crashes.
- Ensure that the mobile app is tested for all memory leaks and resource releases once the applications is closed.
- Is the application tested for resource overrun, peak load and other scenarios that lead to app crash?
- Does the app support multi-tasking and multiprocessing?
- Given below are the checklist points from UI point of view:
  - Does the app use standard colors (as per visual specs)?
  - Is the font uniform across pages. Is all text properly aligned?
  - Does text wrap properly around pictures/graphics?
  - Is the error message text spelt correctly on this screen?
  - Does Progress message appear on load of tabbed (active) screens?
- Given below are the checklist points from usability stand point:
  - Verify if the app behaves as desired if device is tilted (portrait/landscape)
  - Verify if the page navigation is smooth
  - Verify if the font size and spacing ensures good readability
  - Verify if the labels and buttons text are clear and concise on every page
  - Verify if the UI elements provide visual feedback when pressed

3.4 USER INTERFACE DESIGN FOR MOBILE APPS

The user interface for a mobile app is one of the critical design item as it plays a major role in user engagement. The main goals of a user interface design for a mobile app are as follows:

- Ease of use: The UI should be easy to learn and easy to use.

- High productivity: User should be able to find the requisite information quickly as well as complete the task quickly
- Easier navigation: User should be able to navigate across the screens easily through intuitive information architecture.
- Minimal error: The app should minimize the error rate for the end user.

3.4.1 Experience Design Process

The key steps in designing the experience (user interface) for mobile apps is given in figure 3.1.

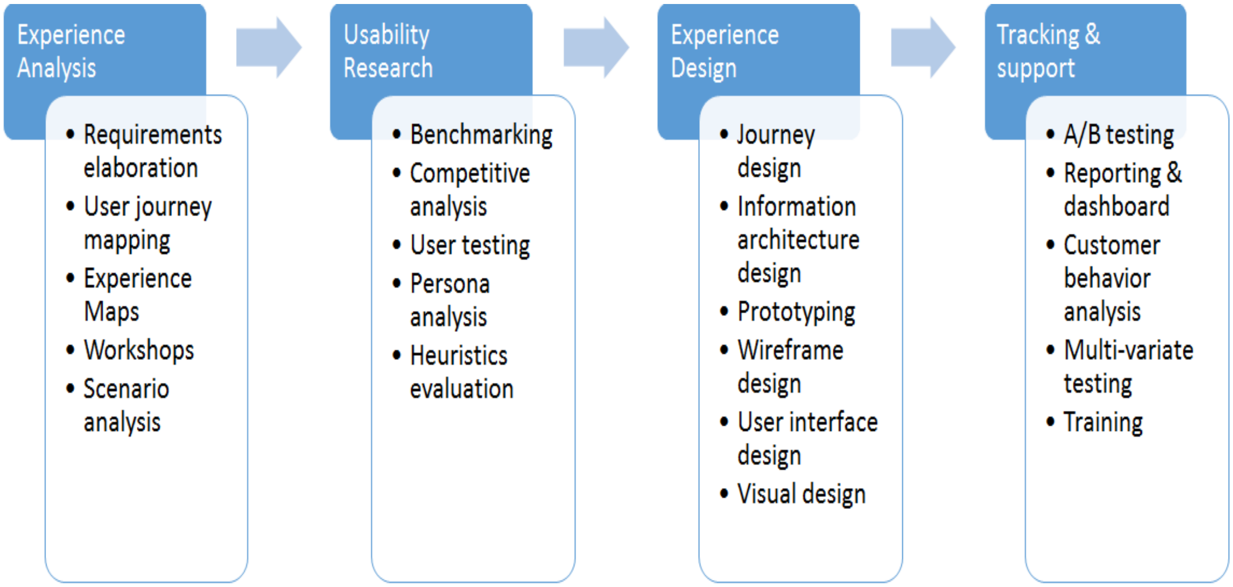


Fig. 3.1: User Interface Design Process for Mobile Apps

Let us look at the key activities in each of the phases:

Experience Analysis phase

In this phase, we would analyze various aspects of mobile app experience. For consumer apps, we always take user-centric approach wherein we place high user engagement and user satisfaction above all other goals. Main activities in this phase are detailed below:

- Requirements elaboration by interviewing key stakeholders and conducting workshops with all users.
- Map the user journey across various user groups and identify key touch points.
- Identify any pain points or challenges in the as-is scenario.
- Develop experience maps for user groups.
- Perform scenario analysis for key user groups.

Usability Research phase

During this phase, we will design for each user journey. Key activities in this phase are as follows:

- Persona analysis: Identify distinct user groups (personas), and map their user journey, tasks, goals and needs.
- Benchmarking and competitive analysis: Benchmark the user design with competitors.
- Heuristics evaluation: Run the key heuristics to ensure that the design conforms to all specified heuristics. Heuristics related to usability, simplicity, user controls, consistency, branding, error handling, flexibility, efficiency, help, aesthetics standards will be tested with help of experts.
- User testing: Test the design with intended users and experts.

Experience Design phase

During design we strive to create simple to use app user interface that enhances various touch points of user journey. Main activities in this phase are given below:

- Wireframe design: We create low-fidelity wireframes (such as sketches, videos) based on journey and persona analysis
- Prototype: Mockups and prototypes are developed to get user feedback. HTML and JavaScript will be used to develop interactive prototypes. Prototype will be tested with stakeholders and users.
- Visual design: Develop the specifications for UI elements such as page layout, fonts, images, buttons, videos and provide design guidelines. Design will be validated through eye-tracking tests.
- Information architecture and navigation model will be defined in this phase. Also, user navigation journey will be defined for various user personas.
- All personalization and contextual features related to mobile app will be designed.

Tracking and Support phase

In this phase, we mainly track the user behavior on the new design to understand the effectiveness of the design. Given below are the key activities.

- A/B testing: We carry out testing with two variants to understand the effectiveness
- Multivariate testing: We test multiple variants of the design.
- Customer behavior analysis: We use analytics to track and analyze the customer behavior.
- Dashboard reporting: We report all the findings and insights in an intuitive dashboard format.

User interfaces of a Mobile app for authentication and dashboard for an insurance company is shown in Figure 3.2.

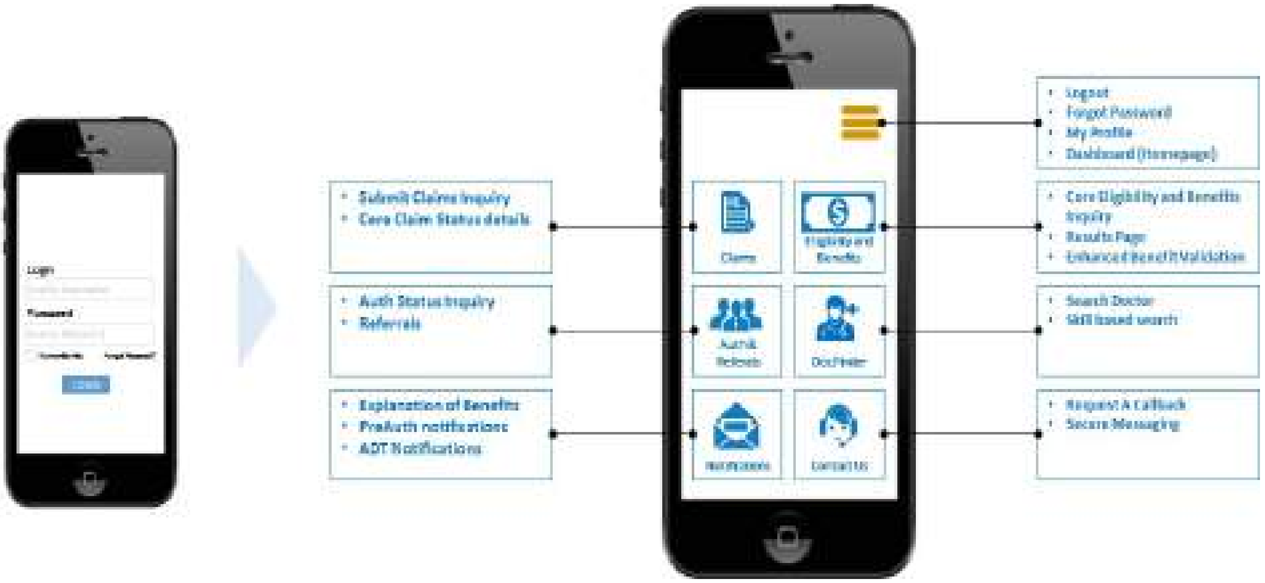


Fig. 3.2: Mobile app user interface of an Insurance Company.

3.5 DEPLOYMENT

Figure 3.3 shows end to end steps involved in the lifecycle of a mobile app deployment.

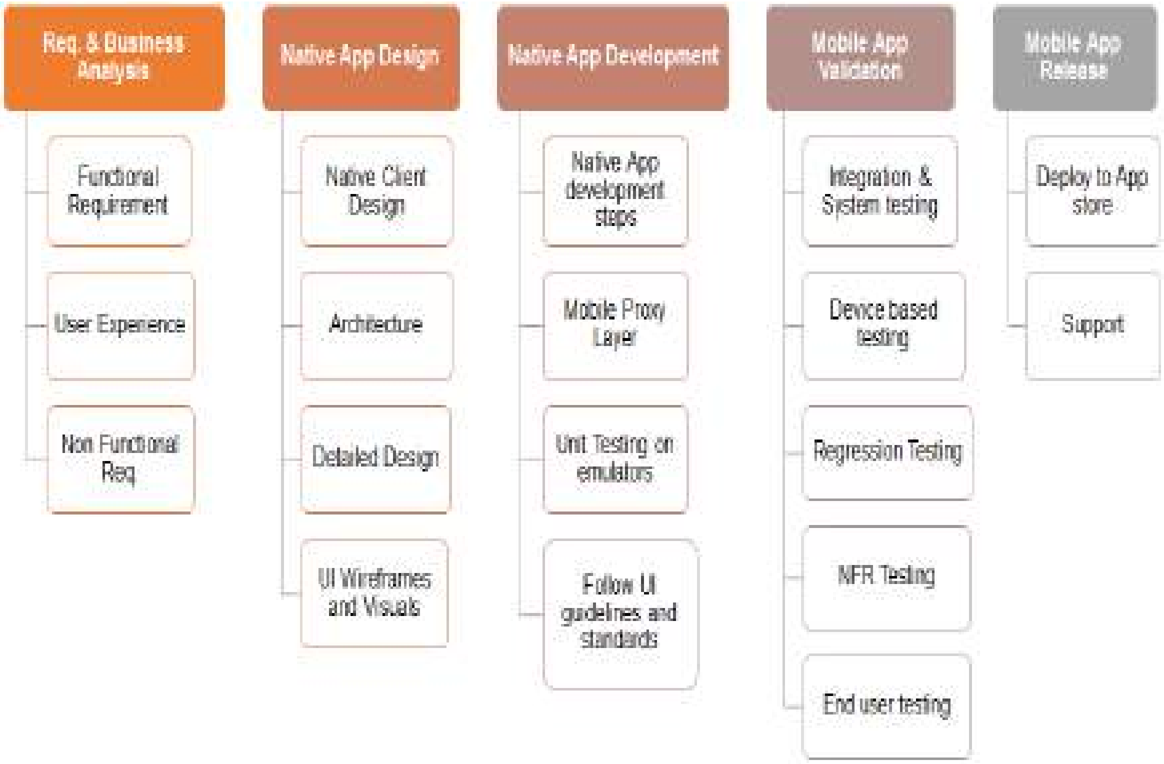


Fig. 3.3: Steps involved in Mobile app deployment

- **Requirements and Business Analysis**: In this phase, we will analyze detailed user requirements, journey mapping, and persona analysis and compile all functional and nonfunctional requirements. We discussed the detailed steps as part of experience design process. We will also understand the requirements related to security, performance, accessibility, localization, standards, modularity, etc.

- **Native App Design:** During this phase, we will design the user experiences, information architecture needed for the mobile app. The end-to-end architecture and detailed design would be carried out for the mobile app. Other key deliverable, in this phase are wireframes, visual design, and HTML prototypes.
- **Native app development:** In this phase, we will follow all the specified design guidelines to develop the native mobile app for the specific platform. Wherever needed we will also develop proxy layer for interaction with other systems and services. The developed app will be tested on various emulators.
- **Mobile app validation:** Various forms of testing such as device testing, integration testing, system testing, security testing, regression testing and NFR (Non-Functional Requirements) testing would be carried out.
- **Mobile App release:** After the testing is completed, mobile app shall be deployed to the corresponding app store and support for future releases shall commence.

☞ **Check Your Progress 1**

1) The process by which we analyze needs of users is known as ……………………………….. .
2) Comparing design with other existing designs is done through ……………………………….. .
3) Developing interactive mockups is done through ………………….. .
4) Comparing two variants of the design is done using ………………. .
5) ……………………………. involves testing app at peak load.

3.6 POWER USAGE

As the battery life for most of the mobile phones is limited, it is important to judiciously use the battery. Hence, mobile apps must be designed and tested for optimal battery usage. Each mobile platform provides its own set of guidelines for optimal power consumption. Given below are some of the generic guidelines for optimal power usage:

- The mobile app should optimally use the CPU
- Usage of disk, Bluetooth and other networks should be minimized
- Disable all unnecessary background services
- Reduce the frequency of app updates
- Regularly monitor the battery usage of all the apps and test the app with peak load

3.7 SYNCHRONIZATION

Mobile apps use synchronization to sync app data with server to refresh the data. A sample sync architecture for iOS mobile app is depicted in figure 3.4

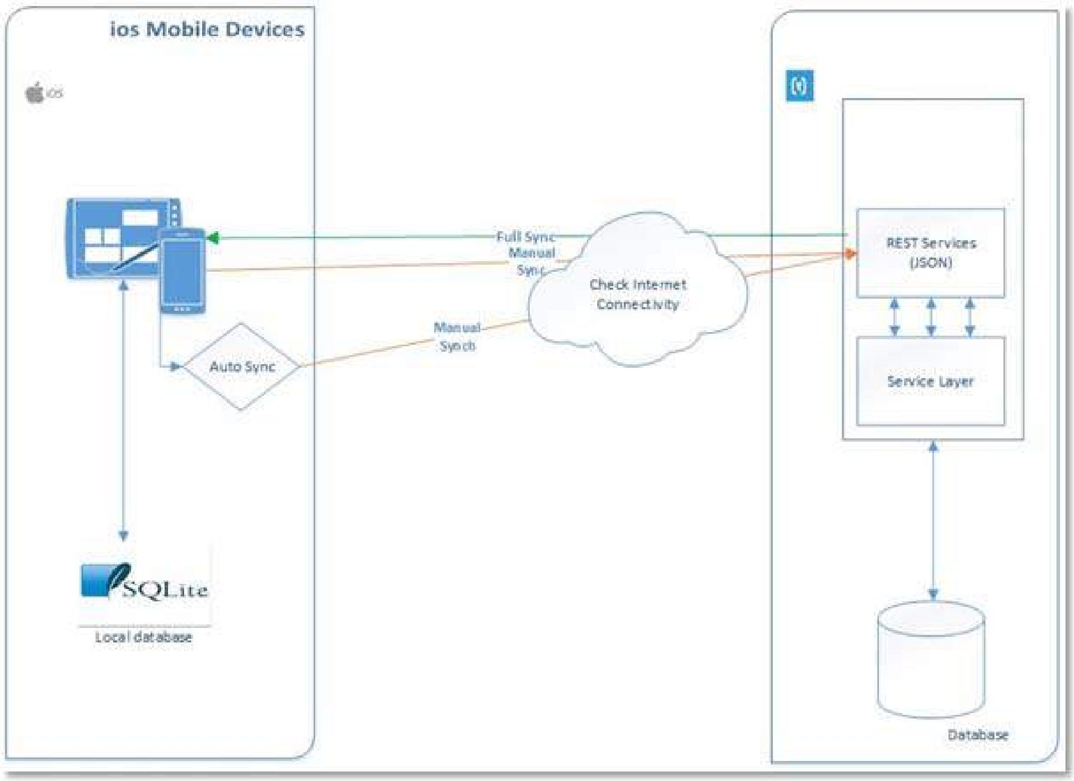


Fig. 3.4: Process of App synchronization

The key steps in the sync process of iOS mobile app are as follows:

i) iOS Offline Mobile application will have two modes of data synchronization:

a) Full Sync

- A full sync involves importing of all the objects. Hence, all the objects in the local store will get refreshed
- Data initialization will happen using full sync upon logging in for the first time

b) Delta Sync

- A delta sync will import only changes and does not import any unchanged record
- Subsequent automatic or manual sync will fetch the data modified after last sync date and time. Offline mobile application will take care of keeping the last sync date on successful sync operation

ii) Data synchronization will happen in both directions from the offline mobile app

- Inbound → for incoming data
- Outbound → for outgoing data

iii) Back end application will expose REST web services to perform Full Sync or Delta Sync

iv) Offline Mobile application will invoke all the required REST web services individually during full or delta sync. This will enable a feature in mobile application of having options for sync of selected entities

v) OAuth (Open Authentication) authentication module will be used for REST based web services.

3.8 PATTERNS AND DESIGN ELEMENTS

Mobile apps are built using platform recommended architecture design patterns to support refactor ability, extensibility and code organization. The following principles of architecture are considered while building mobile apps:

- **Decorator Design Pattern** – Decorator design pattern extensively using Categories, Delegates, and Protocols.
- **Singleton** – Singleton pattern is used for class initiations.
- **Memento** – Memento is used for application state management.
- **Command Pattern** – Command pattern is used for network layer optimization.
- **Façade (Abstract Class)** – Web service API utilization.

In addition to these patterns, following aspects are considered while designing mobile apps:

- **Performance -** Optimized mobile app code and use of best practices will ensure better performance.
- **Security & Standards Compliance –** Security aspects including the following are considered:
  - Sensitive information in platform is encrypted using secure Keychain (AES 256-bit encryption) only
  - Clearing in-memory information at session timeout
  - No Caching of APP data
  - Secure logging
- **Usability -** Allows human interface guidelines, standardized look and feel, navigation, HCI standards etc.
- **Maintainability** - Follows modular approach, platform recommended design patterns, high reusability and extendibility.
- **Code quality** - Strict code reviews and analysis of both static and dynamic nature will ensure high level of code quality.
- **Compatibility -** Follow platform recommended best principles and tools to ensure that app UI is aligned to the recommended form factors to provide completeness.

3.9 SECURITY STANDARDS AND BEST PRACTICES

During mobile application design, it is important to evaluate and take pro-active security measures to mitigate the security risk. The following are some of the practices that will lead to development of a secure Mobile App:

- Security assessment: It is always recommended to assess mobile devices for known security risks and vulnerabilities.

- Security policies:
    - All mobile devices must be password protected.
    - All key applications (such as banking apps) should be password protected.
    - All confidential and user data should be encrypted.
    - All mobile apps should present the privacy policies, data sharing policies, legal policies through end-user license agreements.
    - Password policies should include length restriction (to at least 8 characters), complexity (usage of special characters and alphanumeric), password change frequency and such.
    - All major events such as failed login attempts, apps crashes, and system events should be logged.
    - For secured applications and secure functionality, the system should use multi-factor authentication or mobile device management (MDM) capability.
    - Data at rest and in motion should be encrypted using appropriate encryption standards.
- Various Authentication mechanisms for mobile apps:
    - **Single factor authentication:** Here, user is asked to enter a password every time the application is started or any secured activity is being initiated.
    - **Two factor authentication:** The authentication is performed twice, once with the user credentials and second using the OTP (One time password).
    - **Single Sign on (SSO):** Mobile app is integrated with enterprise SSO solutions to seamlessly access all secured enterprise applications.
    - **Other modes of authentication:**
        - Authentication using popular social channels like Google+, Facebook, Twitter, etc.
        - Authentication using Biometrics for e.g. facial features, speech patterns, fingerprints, etc.
- The following Information Risk Management (IRM) policies should be applied:
    - Periodic threat and vulnerability assessment of all the applications.
    - Remote data wipe methods should be enabled for administrators.
    - Mobile device disposal policies should be devised and enforced to protect the confidential and business critical data.
    - Virus and malware scan should be carried out on periodic basis.
    - Filters and scanners should be installed to prevent vulnerabilities such as phishing, data leakage, cross-site scripting, etc.
    - Screen locking policies should be enforced.
    - Restrict the installation of applications such as Jailbreak to prevent unauthorized usage.

3.9.1 Mobile Platform Security

The following are the security related practices for securing mobile platform:

- **Data Transmission Security -** Provide secure connection for end to end mobile communications.
- **Operational Data Security -** Minimize exposure of data across all end points.
- **Communication Channel Security** - Provide secure communication channels for transmitting confidential data.
- **Application Security -** Provide role based access to all functionality and data and provide access only for authorized roles.
- **On-Device Data Security** - Encrypt data that resides on the device for native and hybrid apps.
- **Encryption Standards**: Use encryption standards such as SHA2 (Secure Hash Algorithm) to encrypt sensitive information.
- All mobile apps should use filters, validations and other secure mechanisms to address following vulnerabilities:
  - ➢ Invalidated input
  - ➢ Broken access control
  - ➢ Broken authentication and session management
  - ➢ Cross site scripting (XSS) flaws
  - ➢ Buffer overflows
  - ➢ Injection flaws (e.g., SQL injection)
  - ➢ Improper error handling
  - ➢ Data under-run / overrun
  - ➢ Application denial of service
  - ➢ Insecure configuration management
  - ➢ Improper application session termination
  - ➢ Insecure storage and transmission
  - ➢ Insecure configuration management
  - ➢ Viewing instructions or code in the server script
  - ➢ Modification by web page users
  - ➢ User-entered input used for script code injection
  - ➢ Access via other non-web-based services
  - ➢ Dynamic generation of other server-side scripts
  - ➢ Dynamically generating executable content (beyond HTML)
  - ➢ Not running as a user ID with least privilege (Running with system level privilege)
  - ➢ Running in a system shell context
- Use secure transport layer (SSL/HTTPS) for secured data transmission.

- App should not log any sensitive data.

  The following practices may be followed for source code security:

  - Regular security review of code.
  - Secured access controlled source code repository.
  - Proper version management and release management processes.

3.10 MOBILE APP TESTING

As testing is an important aspect to ensure quality of the mobile apps, let us look at various testing scenarios for mobile apps. Mobile testing is challenging considering the variety of form factors, hardware, unreliable wireless network, latency issues, etc. Given below are some of the key testing categories that are carried out for mobile app testing.

3.10.1 UI (User Interface) Testing

In UI testing category, we would test the user experience for various form factors, resolutions and on various browsers. The testing normally includes testing look and feel, font, color, controls, touch controls, pinch controls, ease of use, control features, zoom in and zoom out features, etc.

3.10.2 Unit Testing

In this testing category the logical unit of testing would be a module or a screen. We would test the module functionality, screen features, navigation features, flows, business rules as part of it.

3.10.3 Integration Testing

This testing is carried out with integrated set of modules. We will mainly test the data flow, performance, dependencies.

3.10.4 System Testing

System testing involves end-to-end testing for the entire system. This includes testing all the scenarios end to end to check if the application meets the requirements.

3.10.5 Compatibility Testing

This includes testing the mobile app on all supported devices and hardware platforms. We would also test the mobile app for various networks, browsers and carriers. Leverage automated tools to test on various combinations of form factors, devices and mobile platforms.

3.10.6 Performance Testing

The mobile app will be tested at various loads and various bandwidths for response times. Identifying bottlenecks is one of the key activities in this phase of testing. The resource usage such as battery usage will be tested along with scalability and reliability.

3.10.7 Security Testing

This includes testing various security scenarios of authentication, authorization, authorized port access, checking user permissions check, etc. We will also verify potential vulnerabilities such as SQL injection, input validation, account lockout scenarios, session handling, communication, data encryption, information leakage, etc.

3.10.8 Synchronization Testing

This testing involves testing synchronization scenarios between mobile device and the server. This also includes testing data integrity during synchronization process and handling network failure scenarios.

3.10.9 Usability Testing

The testing is carried out from end user stand point of view to test the ease of use, efficiency, recall, ease of navigation, etc.

In addition to the above mentioned tests, we would also conduct other tests based on the application needs:

- Installation testing to test the ease of installation on various platforms
- Recovery testing to understand the ease with which the app recovers from failure.
- Battery consumption testing to test the usage of battery for extended durations
- Network testing to check the app behavior on various networks with differing bandwidths.
- Interruption testing to check the behavior of app during interrupts such as incoming call, message, flash message, system events etc.
- Compatibility testing is carried out mainly for mobile web apps to test the experience and behavior on various devices and mobile platforms.
- Localization testing to test the app behavior for various languages, translation needs and cultural requirements.

☞ **Check Your Progress 2**

1) For optimal power usage, frequency of app updates should be ……………. and unnecessary background services should be …….. .
2) …………………… pattern is mainly used for class initiations.
3) ……………………… design concern deals with look and feel and interface guidelines.
4) ……………………………………..… authentication forces users to authenticate multiple times.
5) …………………………… involves end-to-end testing for the entire system.

3.11 SUMMARY

In this unit, we started discussing the key design considerations, anti-patterns and best practices in mobile app design. We provided a checklist and then provided deep dive concepts of user interface design for mobile apps. We also looked at the power usage and synchronization aspects of mobile app design. We then looked at various patterns, security standards and various forms of mobile app testing.

3.12 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) Persona analysis
2) Benchmarking and competitive analysis
3) Prototyping
4) A/B testing
5) Regression testing

Check Your Progress 2

1) Minimized and disabled
2) Singleton
3) Usability
4) Two factor
5) System testing

3.13 FURTHER READINGS

References

- https://en.wikipedia.org/wiki/Mobile_application_testing
- https://en.wikipedia.org/wiki/User_interface_design
- https://www.owasp.org/index.php/OWASP_Mobile_Security_Project

ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Computer and Information Sciences

C·O·L

Commonwealth of Learning

INTRODUCTION TO MOBILE ARCHITECTURE

Mobile Operating Systems

2

Introduction to Mobile Architecture

School of Computer and Information Sciences
Indira Gandhi National Open University

Commonwealth *of* Learning

Block

2

MOBILE OPERATING SYSTEMS

INTRODUCTION TO MOBILE ARCHITECTURE

This course has been developed as part of the collaborative OER Course Development for ICT Skills initiative of the Commonwealth of Learning (COL). COL is an intergovernmental organisation created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

Indira Gandhi National Open University (IGNOU), established by an Act of Parliament of India in 1985, has continuously striven to build an inclusive knowledge society through inclusive education. It has tried to increase the Gross Enrollment Ratio (GER) by offering high-quality teaching through the Open and Distance Learning (ODL) mode.

For the avoidance of doubt, by applying this license the Commonwealth of Learning does not waive any privileges or immunities from claims that it may be entitled to assert, nor does the Commonwealth of Learning submit itself to the jurisdiction, courts, legal processes or laws of any jurisdiction. The ideas and opinions expressed in this publication are those of the author/s; they are not necessarily those of *Commonwealth of Learning* and do not commit the organisation.

Course Team

Author:

Mr. Sumit Srivastava, MCA
Technical Document Writer,
Indian Patent Office,
New Delhi,
India

Content Editor: Dr.P.Venkata Suresh, SOCIS, IGNOU, New Delhi, India

Content Design Team

Dr. P.V.Suresh,
Associate Professor & Director,
SOCIS, IGNOU, New Delhi

Mr. Amit Kumar Saxena,
Chief Technology Officer,
Grasp IO Inc, Bengaluru

Dr. S. Arumuga Perumal,
Associate Professor & Head of the
Department, S.T. Hindu College,
Nagercoil, Tamil Nadu

Prof.Parma Nand Astya,
Dean, School of Computing Science &
Engineering, Galgotias University,
Greater Noida

Mr. Shashi Bhushan Sharma,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. Akshay Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Dr. Naveen Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. M. P. Mishra,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Dr. Sudhansh Sharma,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Language Editors: Mr. P. Lakshmi Kantham , Faculty in English(Retd.), VSR & NVR College, Tenali, India

CRC Preparation: Tessa Media & Computers, New Delhi

Published by:

Indira Gandhi National Open University, Maidan Garhi, New Delhi-110068, India.

The course has been developed with the support of the Commonwealth of Learning, Canada.

Acknowledgements

Indira Gandhi National Open University wishes to thank those below for their contribution to development of this Course Material:

Staff of School of Computer and Information Sciences, Administration, Finance & Accounts Division and Electronic Media Production Centre, IGNOU, New Delhi, India and all External Experts. Special thanks to Dr.Sanjaya Mishra, Education Specialist(eLearning), Commonwealth of Learning, Canada for his support and guidance through out.

BLOCK INTRODUCTION

This is the second block of the Course. This block introduces the learner to Operating Systems that were running Mobile devices.

Operating System plays a major role on the entire functionality of a device irrespective of the fact that it is a PC, Laptop , Tablet or a Mobile phone. There are numerous operating systems for each category of devices. In the case of Mobile devices also, there are large number of operating systems. In this block, we mainly discuss three popular Mobile operating systems, namely, Android, iOS and Windows Mobile. Initially, we discuss operating systems in general and then discuss specific Mobile operating systems.

The units in this block will demonstrate the running of some apps on some of the Mobile operating systems as well as configuring the concerned devices for some of the most important functionalities.

This block consists of four units and is organized as follows:

Unit 4 introduces Mobile Operating Systems (MOS). It covers basic functions of an Operating System followed by specifics of MOS. It also briefly discusses about the MOS that are currently in market and those that are popular but discontinued by the concerned developers.

Unit 5 introduces the basics of Android. The unit specifically discusses interface of Android devices, applications supported by them, the way the memory is managed and support for advanced features such as Virtual Reality.

Unit 6 introduces iOS which runs iPhone and iPAD. The accessibility features in addition to its flagship app SIRI are discussed in addition to other features of iOS.

Unit 7 introduces Windows Mobile operating system. The unit covers various versions of Windows operating system that was used to run on Windows based mobile devices including the latest Windows 10 Mobile operating system.

UNIT 4 INTRODUCTION TO MOBILE OPERATING SYSTEMS

Structure

4.0 INTRODUCTION

An Operating System may be defined as Resource Manager. An Operating system (OS) handles all hardware & software resources of the Computer. It works as a communication medium between various components of Computer as shown in Figure 4.1

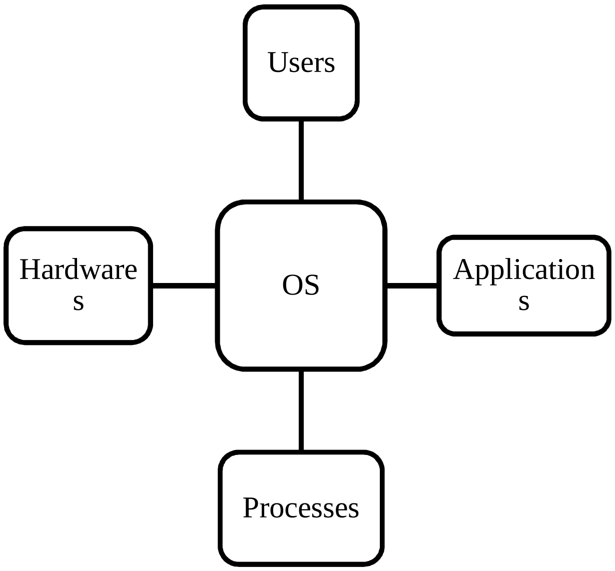


Figure 4.1 : Operating System and other elements

An Operating System not only manages but also allocates resources. The following are some of the basic functions of an Operating System:

- Peripheral Device Management;
- Data File Management;
- Memory Management; and
- Process Management.

4.1 OBJECTIVES

After going through this unit, you should be able to know

- meaning of Mobile OS;
- basic functions of a Mobile OS;
- architecture of a Mobile OS;
- about various Mobile Operating Systems; and
- existing and discontinued Mobile Operating Systems.

4.2 BASIC FUNCTIONS OF AN OPERATING SYSTEM

The following are some of the basic functions of an Operating System:

4.2.1 Peripheral Device Management

An OS handles communication between various devices with the help of its associative drivers. It keeps tracks of every connected device in the system. The unit which keeps track of devices is known as the Input-Output Controller or I/O controller which identifies which assigns processes to devices for specific duration. It allocates and de-allocates processes to devices in an effective way.

4.2.2 Data File Management

For smooth navigation and use of data files; a file system is generally organized into directories/folders which will consist of files and sub directories. Operating System maintains a log of information with respect to every file and folder such as location of the file, users of the file, date of creation, access rights etc. This information will enable OS to allocate/de-allocate resources efficiently.

4.2.3 Memory Management

Memory management involves management of Primary Memory which is a collection of data or bytes where each datum or byte has its own memory address or location. Primary memory provides a quick repository which can be accessed straightaway by the Central Processing Unit (CPU). Operating System keeps tracks of primary memory. That is, which portion of the memory is in use and by whom. Also, it keeps track of portions that are not in use.

4.2.4 Process Management

An OS also manages the allocation of the processor to various processes. The assignment is for a fixed period. This is known as Process scheduling.

Operating System keeps track of processor and status of each process in the memory. The unit which is responsible for executing this activity is known as traffic controller which allocates the processor to a process and also de-allocates processor when no longer required.

4.3 MOBILE OPERATING SYSTEMS

Mobile operating system is the system software which operates upon mobile devices. It is the software that provides the base upon which applications or programs can be run effectively on mobile phones, smart phones, tablets and hand held devices. Apart from the basic functions of an operating system, the mobile OS also handles cellular & Wi-Fi connectivity as well as access to device itself.

Now a days, mobile phones are used frequently to check emails, play games, watch news and make video calls etc. This development has brought latest functions to mobile phones that were previously only available on our Computers. There are hundreds of thousands of apps available each with its own purpose. You may download a news app that tells you the current news or updates in your city or around the globe or a game app. Indeed, we can find an app for any function that we are looking for.

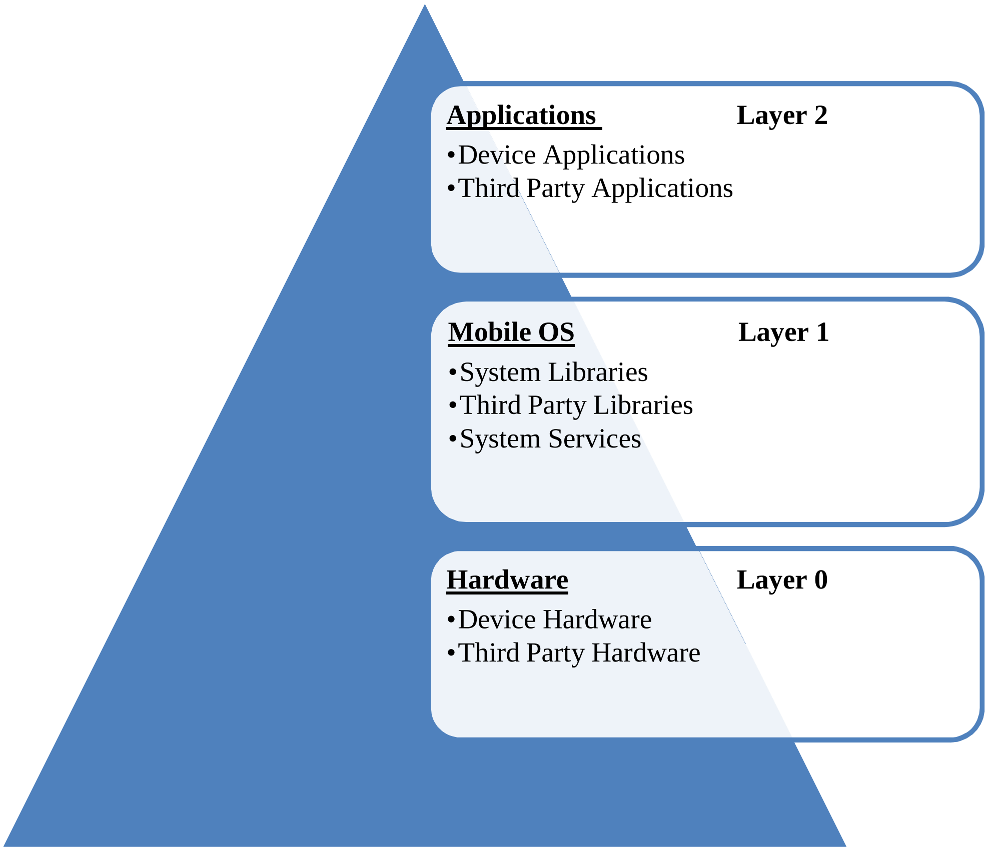


Figure 4.2 : Layers of a Mobile OS

The following are different layers of a Mobile Operating System (Figure 4.2):

4.3.1 Layer 0

It includes device hardware and third party hardware.

Device Hardware

Device Hardware refers to the identification of all physical components which are essential to develop a complete product in terms of its usability. The hardware components of the mobile include circuits, IC's, screen, and speakers which are needed to make a complete mobile phone. The hardware is basically designed while keeping the software requirements in mind, software or application will run on designated hardware only like Apple iOS can be configured on apple manufactured devices.

Third Party Hardware

It includes all physical components of the mobile devices which are manufactured by companies besides the original mobile manufacturer like using Samsung earphone in Apple iPhone. In other words, the hardware or physical component of the system which is not supported by the original manufacturer is known as third party hardware. For example, you buy anASUSlaptop and then upgrade it by installing third party hardware, like an Intel video card and a hard drive ofwestern digital technologies. Since the hardware is not pre-installed with the laptop and are bought from companies other than ASUS, they are called third party hardware.

4.3.2 Layer 1

It includes system libraries, third party libraries and system services.

System Libraries

These are In-built software resources which are basically developed for smooth functioning of system applications. It includes configuration, message templates, help data etc. In other words, System Libraries are the programs which are developed for providing the base, with the help of which application program can be run smoothly on the device. For example, playing a game on mobile phone required a graphics library for rendering different scenes, similarly for smooth functioning of system, system libraries are needed.

Third Party Libraries

These are those software resources which are required for running third party applications like flash player is required for running SWF (Small Web Format) file on the web browser. Similar to third party hardware, third party libraries are offered by the companies besides the original manufacturer of the application. For Example, you might require plug-in for running windows applications in Macintosh environment. We use Microsoft office in android operating system with the help Microsoft libraries designed for android so we can say third party libraries are used to support third applications.

System Services

These are the services which are actually required for running a mobile OS like notification manager, message organizer, contacts, dialing services etc. No mobile can work without system services; they have to be initiated or started first before using any phone. For example, you cannot write on a paper without a pen similarly all system services should be loaded first; boot manager is required for starting the operating system. Wifi service is needed for establishing a wireless connection etc.

4.3.3 Layer 2

It includes device applications and third party applications.

Device Applications

These are the pre-installed applications which come up along with the purchase of your smart phone;these are pre-loaded in the smartphones. For example, if you buy an apple iPhone, then you can find Apple iTunes, Apple Store, iWork etcare pre-installed in your phonesimilarly android phones come up with Google play store, google drive, YouTube, Gmail etc.

Third Party Applications

These are the extra applications which are not bundled along-with device applications. Basically, these applications are used to extend the features of your device. They not pre-installed applications, a user has to download and install it from the app store or play store. Some of the most commonly used third applicationsare MX Player, WhatsApp, Twitter etc.

4.4 Architecture of Android

Figure 4.3 depicts architecture of Android mobile operating system.

The architecture of Android includes the following four layers:

- Application
- Application Framework
- Libraries or Android Runtime
- Linux Kernel

Applications

This is the top most layer of the android operating system that categorized the applications or apps in to two parts, the first one manages the native or device applications and the second one manages the third part applications.

Application Framework

This layer provide the environment or base upon which different application can be run smoothly, like activity manager keeps tracks of the activities which are being carried by different applications, location manage uses the GPS to identify the location of the device, window manager maintains a cache or record of already opened applications etc.

Libraries or Android Runtime

This layer is used to identify the use of different resource libraries, which are required by different applications, like SQLite, which is database for storing data of mobile applications, Free type library is used to identify the font to be loaded, SSL maintains a secure connections if payment related app is functioning etc.

Linux Kernel

This layer handles different physical component of the device, by managing the processes, memory, power and drivers. It acts as an interface between the hardware and applications.

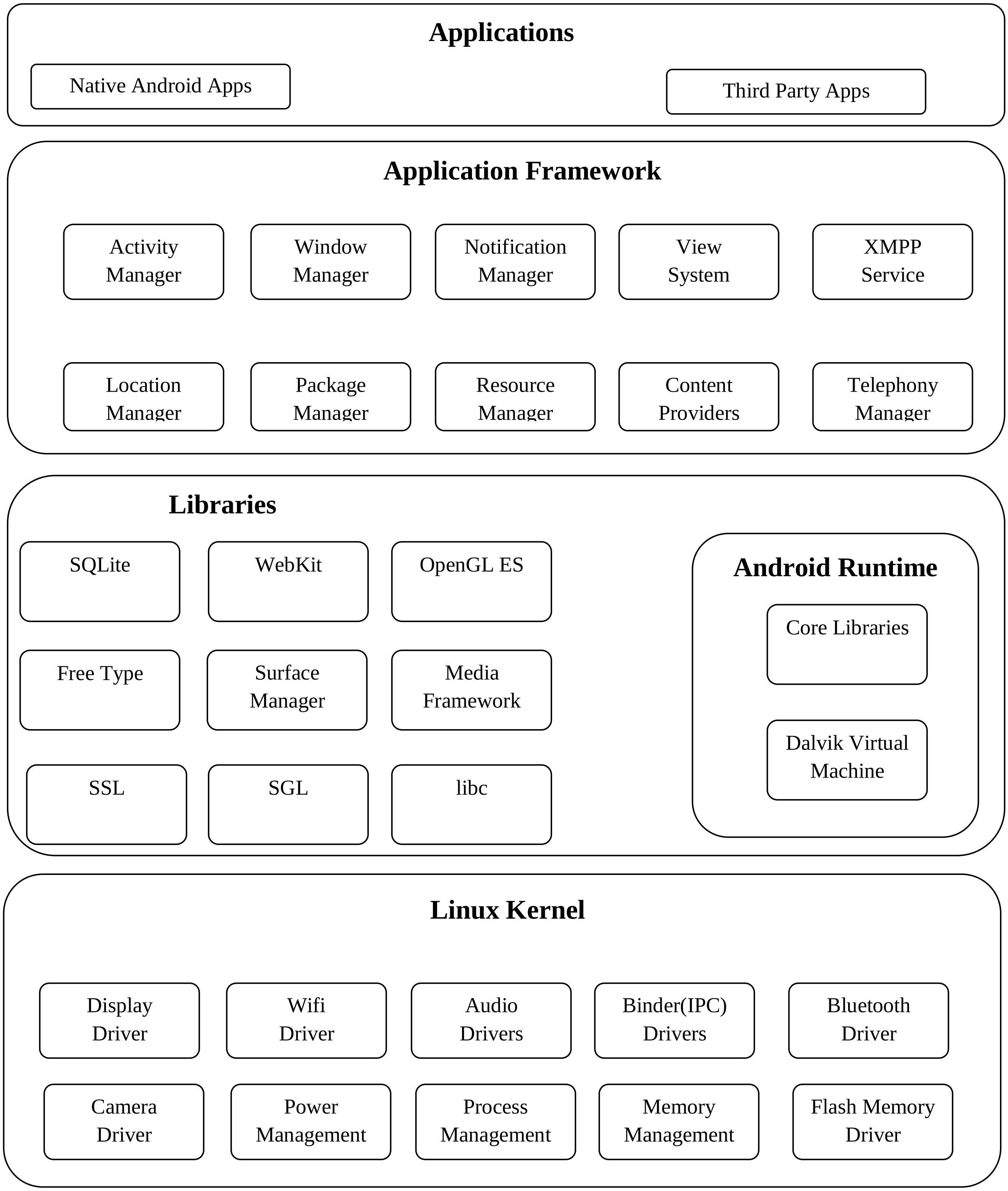


Figure 4.3 : Architecture of Android Mobile Operating System

4.5 KNOWING THE OPERATING SYSTEM OF A MOBILE PHONE

Mobile operating systems are tightly integrated with appearance, interface and function of a mobile phone that the name and version of its operating system software can be found only from concerned phone’s *Settings* as shown in figure 4.4. There are some examples of mobile operating systems which includes Apple iOS, Google Android, and Microsoft's Windows Phone OS.

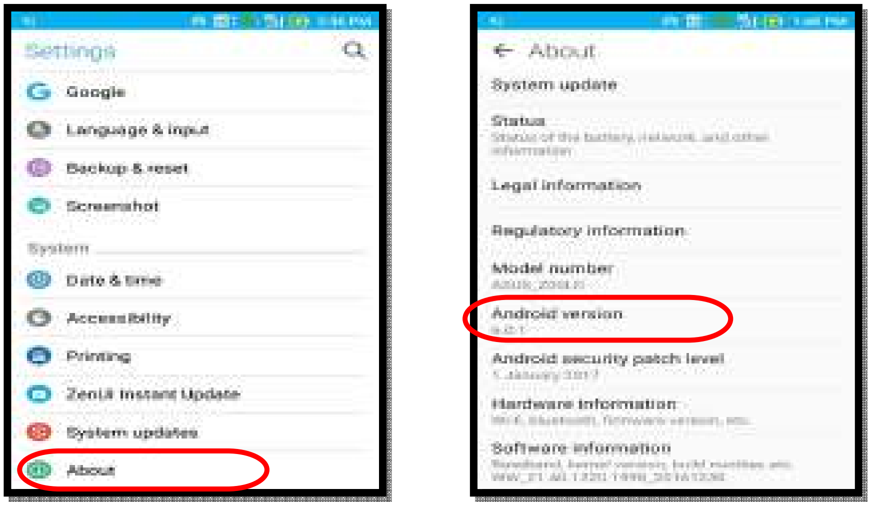


Figure 4.4 : Knowing the operating system of the mobile phone

Most of the mobile operating systems are bound to particular hardware that they come up with little flexibility and have some restrictions. For example, Apple’s iOS permits users to install applications only from Apple’s *Appstore*. In the case of Android, its *Play Store*.

Here's the list of commonly used mobile operating systems:

- Apple’s iOS
- Google’s Android
- Microsoft’s Windows Phone OS
- Nokia’s Symbian
- BlackBerry OS (Research in Motion)

Check Your Progress 1

1) An Operating System may be defined as ………………………….

2) In ………………….. Device Management, An OS handles communication between various devices with the help of its associative drivers.

3) ………….. layer of a Mobile OS includes device applications and third party applications

4) ActivityManager and WindowManager are part of …………….. layer of Android architecture

5) ………….. menu includes information about the operating system and its version in Mobile phone

4.6 DISCONTINUED MOBILE OPERATING SYSTEMS

The following are the discontinued Mobile Operating Systems. Though they are discontinued, some of them are still used. Discontinuation means that there will be no further upgrades or releases of new versions. Table 4.1 describes them briefly:

- Firefox OS
- Bada
- Symbian
- MeeGo
- webOS
- BlackBerry OS

Table 4.1 : Discontinued Mobile Operating Systems

Firefox OS	Firefox OS was introduced by the Mozilla Foundation for tablet PCs, smartphones, smart TVs and other handheld devices.It was an open source operating system that supports all standard web based technologies (HTML, CSS and JavaScript). It was publically launched in February 2012 and by the end of December 2014 it was offered from 14 operators in 28 countries. In September 2016, Mozilla formally declared that it would stop further development of Firefox OS smartphones.
Bada	Bada is a discontinued proprietary mobile Operating System of Samsung Electronics which wasintroduced in 2010. Samsung Wave was the first smartphone to use Bada OS. Bada supports mobile features like 3D graphics, multipoint-touch and of course, application downloads and installation. It was publically launched in February 2010 in Barcelona. In February 2013, Samsung publically announced that it will stop developing Bada.
Symbian	Symbian mobile operating system was initiallyfocused at mobile phones that provide a high-level functionality for exchanging information and managing personal information. This mobile OS acts as a middleware with transmission of wireless signalsusing integrated mailbox and the integration of Java and personal information managementfeature. It was the biggest collaboration between software and mobile manufacturing companies like Psion Software, Ericsson, Motorola and Nokia. The Symbian foundation shattered in late 2010 and only Nokia managed to control the development of the OS.

	The last Symbian smart phone from Nokia was Nokia 808 Pure View.
MeeGo	It is an open source discontinued mobile operating system hosted by the Linux foundation and was mainly targeted at mobile phones and information gadgets in the electronics market. In February 2010, Nokia and Intel announced the launch of MeeGo T01. In September 2011, it was discontinued in favor of Tizen (a new Mobile OS).
webOS	Palm Inc, a US based company was the originator of the webOS mobile operating system. It was the successor to Palm Operating System but later HP acquired Palm and now known as webOS in HP repository. HP deploys webOS in a variety of its devices including smartphones and HP Touchpad. webOS runs on the Linux kernel.
BlackBerry OS	Research In Motion introduced Black Berry OS for its deploymentin company’s most popular Black Berry phone which was popular with corporate group users as it provides compatibility with Microsoft Exchange, Lotus Domino, Novell Group Wise email and other business software.Now it comes under the category of discontinued mobile operating systems. It was discontinued after the release of BlackBerry 10.

4.7 EXISTING MOBILE OPERATING SYSTEMS

The following are some of the existing Mobile Operating Systems. Table 4.2 describes them briefly:

- iOS
- Android
- Windows 10 Mobile
- Sailfish OS
- Tizen
- Ubuntu Touch

Table 4.2 : Existing Mobile Operating Systems

iOS	iOS mobile operating system was initially introduced by the Apple Inc for its deployment in iPhone devices. Now it is compatible with variety of Apple devices such as iPhone, iPad, iPad 2, iPod Touch etc. You can see iOS mobile operating system on Apple's own manufactured devices becauseapple does not grant license for its deployment on third-party hardware. iOSmobile operating system is extendedusing Mac OS X operating system of Apple Inc.

Android	Android is the most commonly known mobile operating system which was introduced by the Google under the category of Google's open and free software repository.The free software repositoryconsists of an operating system, middleware and also commonly used applications on mobile devices. The development of android version is inspired from the word “dessert” every version is coming up in alphabetical order with further updates and enhancements. The existing android version names are • Cupcake • Donut • Eclair • Gingerbread • Honeycomb • Ice Cream Sandwich • Jelly Bean • KitKat • Lollipop • Marshmallow • Nougat
Windows 10 Mobile	Windows 10 Mobile is the newest version of the windows operating system where Microsoft is trying to unify their desktop computer, tablet and mobile operating system in to a single OS. Windows 10 Mobile will carry many of the same features as it desktop version. In November 2015, Microsoft launched windows 10 mobile OS with Lumia 950, Lumia XL and Lumia 550.
Sailfish OS	Sailfish OS is the product of Jolla ltd which is based on Linux kernel. It is coming up with Jolla smartphones and tablets. Also targeting controlling and smart building equipment’s. It is an extended version of MeeGo OS which was developed by Nokia and Intel.
Tizen	Tizen mobile operating system is an open and flexible operating system whose aim is to serve different industry requirements including mobile operators, device manufacturers, and software developers. It is mainly focusing on UI/UX development in order to meet the requirements of specific user segments.Now Samsung is the only Tizen member who is incorporating and further developing the operating system. In January 2015, Samsung released the Tizen-based Samsung Z1 mobile phone in India.
Ubuntu Touch	This Mobile OS is developed by Canonical Ltd as a mobile version of Ubuntu operating system. It is aimed at touch screen devices like smartphones, tablet computers and other handheld devices. In October 2011, it was announced for tablets, smartphones smart TVs and smart devices like smart watches, head units in car, smart wrist bands etc. It is coming up with Samsung Galaxy S4 Google Edition and Nexus 4 phones.

4.8 TYPES OF MOBILE OPERATING SYSTEMS

As with Software, there are open source and closed source mobile operating systems.

The following are some of the open source mobile Operating Systems:

- Tizen
- Plasma Mobile
- Firefox OS
- Sailfish OS
- Ubuntu Touch

The following are some of the closed source mobile operating systems:

- iOS
- BlackBerry OS
- Symbian
- Bada
- Palm OS

Figure 4.5 depicts the usage trend of various Mobile / Tablet operating systems during January , 2015 to December, 2016.

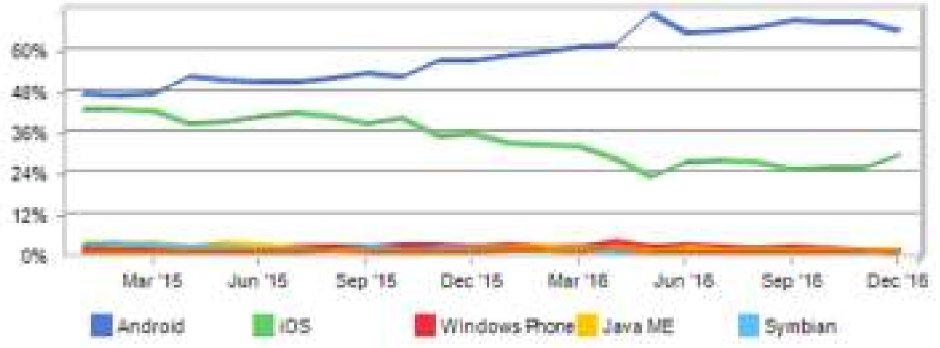


Figure 4.5 : Usage trend of various Mobile/Tablet operating systems

Courtesy: https://www.netmarketshare.com

The Mobile OS is an Amazing platform for developers for developing mobile apps innovatively and publish in market instantly. Now a days, mobile OS is also diverging in areas like tablets, smart TVs, Cameras, Smart watch etc. The most common OS's for mobile devices are Apple's iOS and Google's Android which still show growth. Mobile OS's like Research in Motion's (RIM's) BlackBerry OS and Microsoft's Windows Phone are down the ranks. As far as uses of mobile phones are concerned, India has become the second highest country after china to have maximum number of mobile users.

4.9 SUMMARY

From last few decades, smart phones have changed our lifestyle and now, tablets have entered into the market as well. These smartphones are carrying a notable impact on our lives and are in fact re-constructing the way we get information and communicate with others. This is not because of the device hardware but the specialized software that these devices run and most crucial, their operating systems. Similar like conventional operating systems (like Windows, Linux, BSD etc.) or other versions of the same operating system (like Windows XP, Windows Vista, Windows 7 etc.), most of the smartphones can also run varied versions of the operating system they were made for and in some cases, they might even be capable to run operating systems they weren't made for. In general however, a smart phone with Google Android will only run a version of Google Android where as an Apple iPhone will only run an iOS version.Manysmartphone manufacturers use their own proprietary mobile operating system for their phones and tablets. A known example is Apple, with iOS being the operating system developed for Apple iPod, Apple Touch, iPhone and iPad devices. Similarly, RIM(Research in Motion) who use their proprietary BlackBerry OS for all BlackBerry phones and tablets, and HP, use their proprietary Palm Web OS for their Palm smartphones and tablets. A features of such operating systems is that they all have a muchuniformed look and feel across all devices that they run on, the way Mac OS X looks and act the same way on a MacBook Pro as it does on an iMac or a MacBook Air.

In this unit, we studied the basic architecture of a Mobile operating system, and architecture of most widely used Android. Also, discussed were some of the discontinued Mobile operating systems that were popular in the past, still some of being used, but with no further upgrades or releases of versions by respective companies that developed them. We also discussed some Mobile operating systems that were currently used in the market. After studying the unit, it's also possible to find the operating system along with version information of your mobile phone. The unit was concluded with sharing the usage trends of some of the mobile operating systems.

4.10 SOLUTIONS/ANSWERS

1) Resource Manager
2) Peripheral
3) Top (Layer 2)
4) Application Framework
5) Settings

4.11 FURTHER READINGS

- Mobile Operating Systems and Programming: Mobile Communications by Dr. ArashHabibiLashkari, Publisher: VDM Verlag Dr. Müller (July 7, 2011) ISBN-10: 3639369173, ISBN-13: 978-3639369175
- Mobile Phone Operating Systems: Symbian OS, Android, Mobile Operating System, Webos, S60, Cyanogenmod, Symbian Platform, Blackberry OS by Books LLC, Publisher: General Books LLC, 2010, ISBN:1157464408, 9781157464402
- http://www.techotopia.com
- http://www.wikipedia.com
- https://developer.android.com
- http://www.apple.com
- http://www.windowscentral.com
- https://www.netmarketshare.com

UNIT 5 BASICS OF ANDROID

Structure

5.0 INTRODUCTION

An Android is one of the most widely used mobile operating system that supports variety of Smartphone's and tablet computers. Initially, Android was developed by the Android Inc. which Google bought in 2005. In 2007, Google released the first beta version of the Android Software Development Kit (SDK), whereas the first public version, Android 1.0, was announced in September 2008. Android source code is available under free and open source licenses. For mobile devices, Android provides a unified approach to application development that means mobile developers develop apps only for Android, and their applications should be able to run on different devices powered by Android. It is a standalone software base that can be deployed easily on any type of hardware configurations.

Features of Android

Android is a robust operating system that challenges Apple 4GS and supports good number of features.. Table 5.1 lists some of the features of Android.

Features	Description
Interface	Quite simple and attractive user interface
Database	SQL Lite, a lightweight RDBMS(Relational Database Management System)
Multi touch	Multi touch is available, two or more objects can be touched
Multi-Tasking	Two or more applications can be run at the same time
Multimedia Support	H.264, H.263, , AMR, MPEG-4 SP, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, WAV, MIDI, JPEG, PNG, BMP and GIF
Connectivity	Bluetooth, Wi-Fi, LTE, NFC, GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, and WiMAX.
Adjustable Widgets	User can adjust widgets to show and hide contents, can resize them
Messaging Service	SMS(Short Messaging Service),MMS(Multimedia Messaging Service) and GCM(Google Cloud Messaging)
Internet Browser	Chrome, Firefox with support for HTML5 and CSS3

5.1 OBJECTIVES

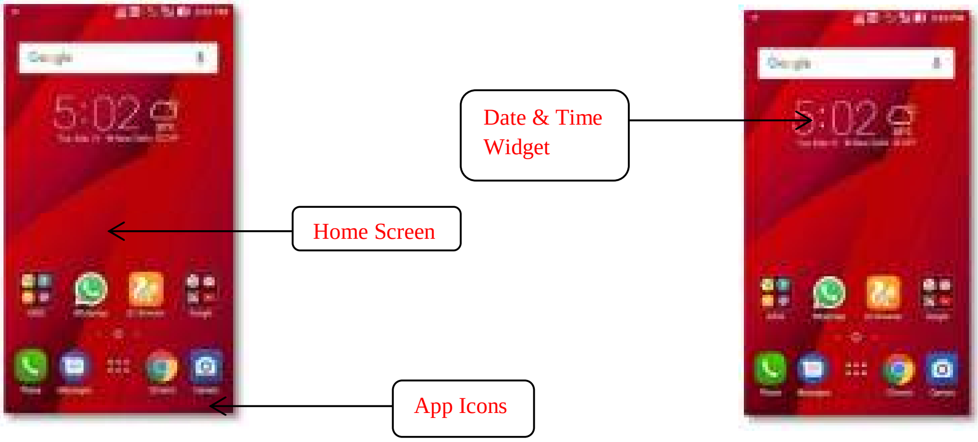
After going through this unit, you should be able to

- know the features of Android OS;
- understand interface &memory management of Android OS;
- develop a simple Android app; and
- understand the concept of virtual reality.

5.2 INTERFACE

The default interface of Android mobile OS is quite simple and easy, it is based upon human–computer interaction approach such as squeezing, tapping and swiping to control on-screen objects, using a virtual keyword. With the help of Bluetooth or USB, we can link-up keyboards, portable speakers; pen drives etc with our mobile phones. The reaction to any information is intended to be brisk and gives a liquid touch interface, regularly through the vibration feature of the gadget to give reply to the user. To react to extra user activities In-built equipment, for example, accelerometers, gyrators and proximity sensors are utilized. For instance altering the screen from portrait to landscape or empower the user to direct a vehicle in a racing game by pivoting the gadget, recreating control of a steering wheel. Home screen is the underlying route on any Android devices which is like the desktop found on PCs. Android home screens are made out of application symbols and widgets. App icons are used to start the relevant app whereas widgets are used to display the live auto-updating content such as a weather forecast, the user's email inbox, news widgets etc. On top of the mobile Home screen there is a status bar that shows information about the Android phone and its network. Status bar can be pulled down to uncover a warning region where applications show critical data or updates. You can drag any application on to the home screen and recent screen lets users to switch between recently used apps.

Figure 5.1 shows some the interfaces of Android OS.



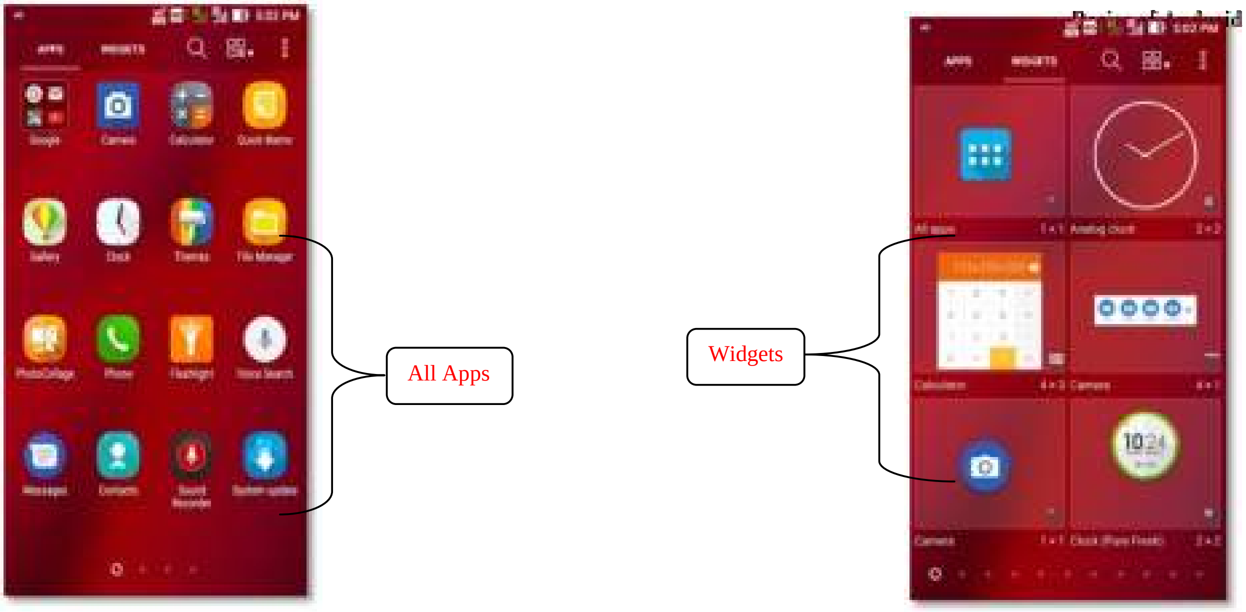


Figure 5.1 : Interfaces of Android OS

5.3 APPLICATIONS

Applications which are commonly known as (“apps”) expand the functionality of the device and are written using the Android software development kit (SDK) and, often the Java programming language. The SDK bundled with a set of development tools like debugger, programming libraries, a handset emulator, documentation, instructional exercises and test code. Android has turned into a developing determination of third-party applications, which can be acquired by downloading and installing the APK (Android application package) file, or by downloading them with the help of an application store which assists users to install, update, and remove applications from their gadgets. Google Play Store is the most ordinarily known application store introduced on Android phones that consent to Google's similarity prerequisites and enables users to peruse, download and update applications which are made accessible by Google and other third-party app developers. Open nature of Android, allows various third-party application stores likewise accessible for Android, either to give an alternate for gadgets which are not pre-stacked with Google Play Store and applications that can't be offered on Google Play Store because of policy infringement or because of different reasons. Amazon Appstore, GetJar, and SlideMeare some of the commonly known third-party app stores.

Now let’s learn to develop a “Hello World” app, First of all you have to download and install the Android Studio from https://developer.android.com/studio/index.html

Start your Android studio and then in the Welcome window, Click on *Start a new Android project* as shown in Figure 5.2.

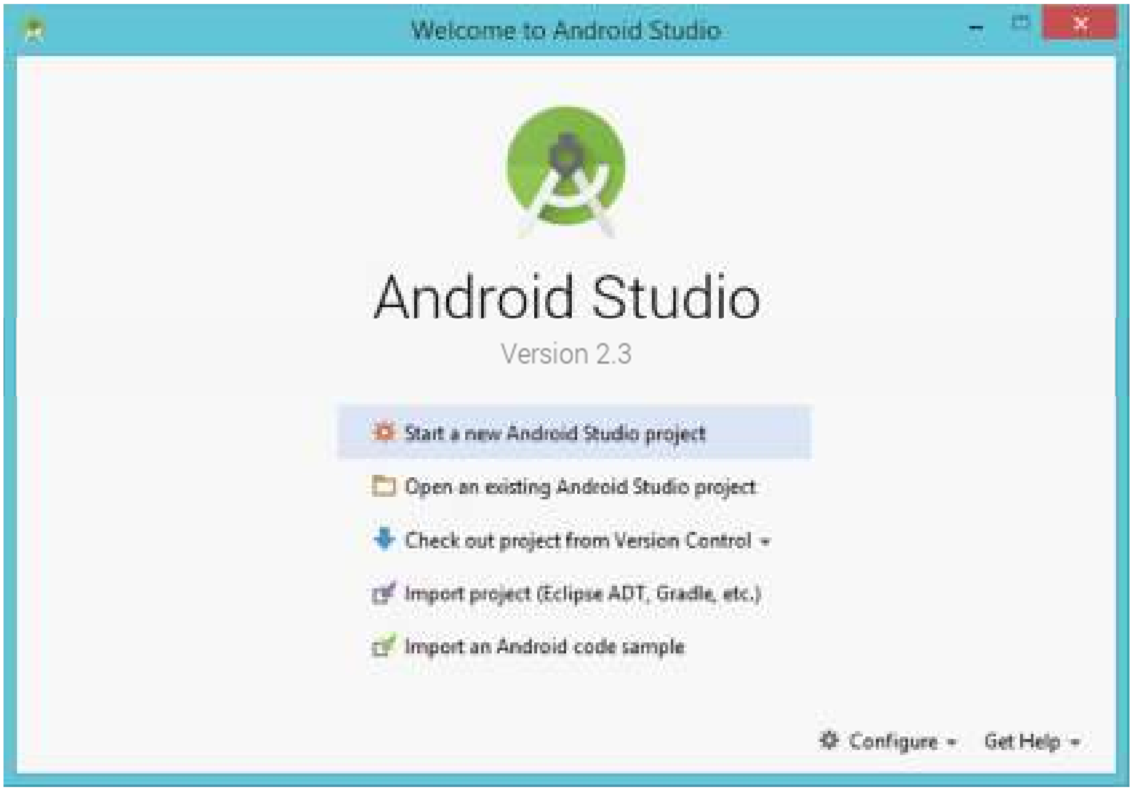
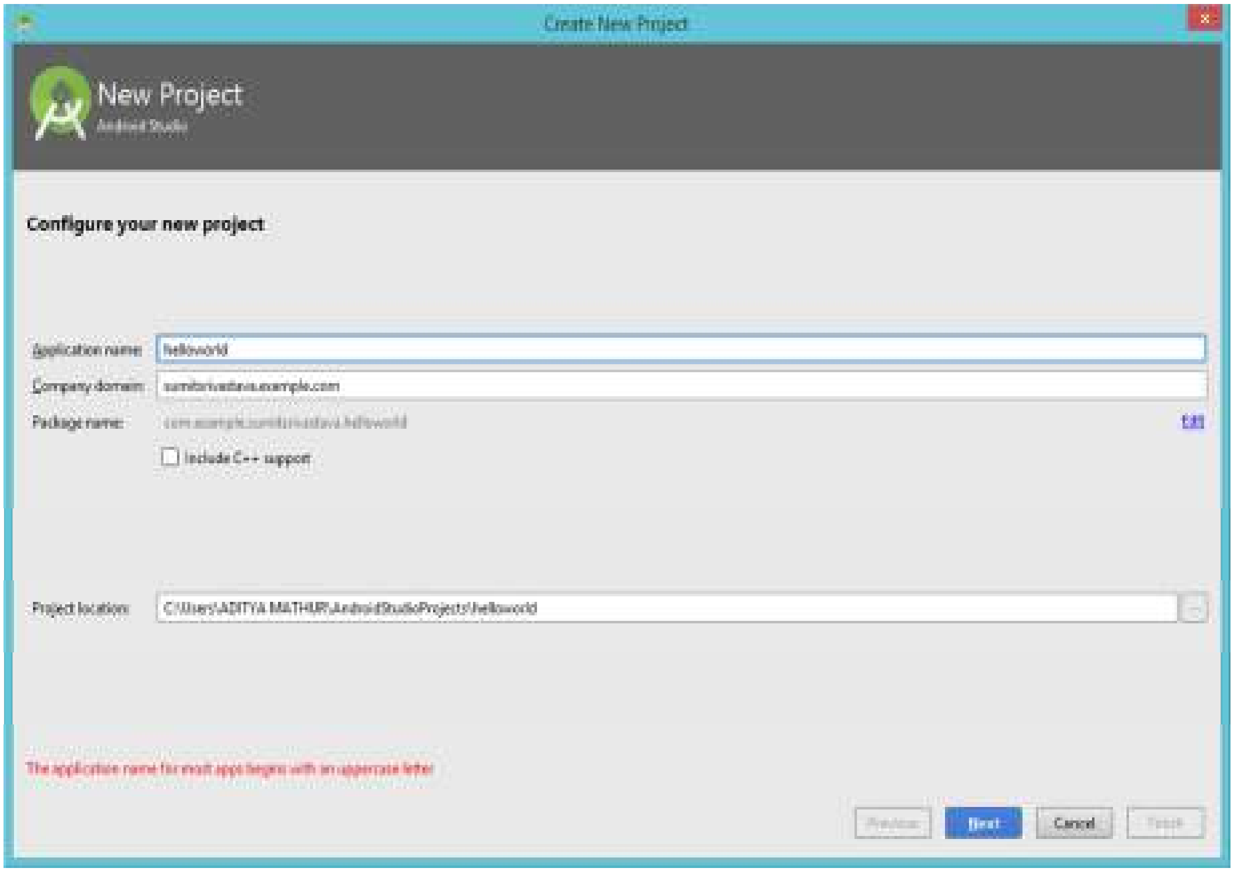

Welcome to Android Studio
Android Studio
Version 2.3
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Get Help

Figure 5.2 : Android Studio



Configure your new project

Application name: helloworld

Company domain: sumitsrivastava.example.com

Package name: com.example.sumitsrivastava.helloworld Edit

□ Include C++ support

Project location: C:\Users\ADITYA MATHUR\AndroidStudioProjects\helloworld

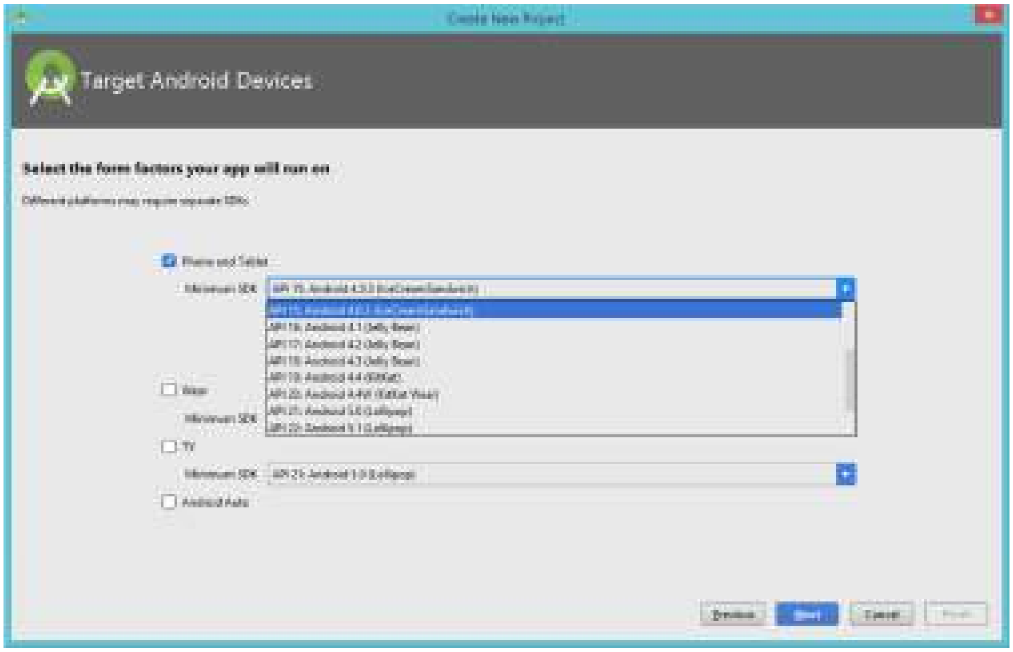
The application name for most apps begins with an uppercase letter

Figure 5.3 : Configuring new Project in Android Studio

In the New Project window, enter application name as *helloworld* as shown in Figure 5.3.

You can also change the location of your project, but, don't select any other option. Now, click *Next*.

In the next window titled *Target Android Devices*, go with default values and then click Next as shown in Figure 5.4.



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

- API 15: Android 4.0.3 (IceCreamSandwich)
- API 16: Android 4.1 (Jelly Bean)
- API 17: Android 4.2 (Jelly Bean)
- API 18: Android 4.3 (Jelly Bean)
- API 19: Android 4.4 (KitKat)
- API 20: Android 4.4W (KitKat Wear)
- API 21: Android 5.0 (Lollipop)
- API 22: Android 5.1 (Lollipop)

☐ Wear

Minimum SDK

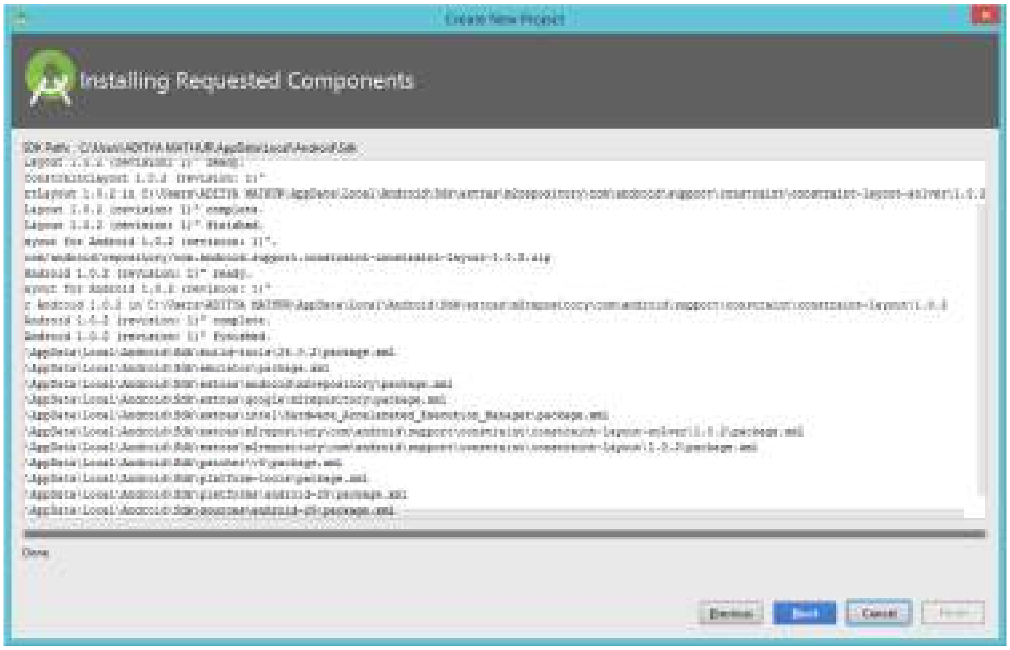
☐ TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

☐ Android Auto

Figure 5.4 : Selecting the Target Android Devices

After clicking *Next* in *Target Android Devices*, requested components shall be installed as shown in Figure 5.5. Click *Next*.



Create New Project

Installing Requested Components

SDK Path: C:\Users\ADITYA MATHUR\AppData\Local\Android\Sdk

```
Layout 1.0.2 (revision: 1)" ready.
ConstraintLayout 1.0.2 (revision: 1)"
ntLayout 1.0.2 in C:\Users\ADITYA MATHUR\AppData\Local\Android\Sdk\extras\m2repository\com\android\support\constraint\constraint-layout-solver\1.0.2
Layout 1.0.2 (revision: 1)" complete.
Layout 1.0.2 (revision: 1)" finished.
ayout for Android 1.0.2 (revision: 1)".
com/android/repository/com.android.support.constraint-constraint-layout-1.0.2.zip
Android 1.0.2 (revision: 1)" ready.
ayout for Android 1.0.2 (revision: 1)"
r Android 1.0.2 in C:\Users\ADITYA MATHUR\AppData\Local\Android\Sdk\extras\m2repository\com\android\support\constraint\constraint-layout\1.0.2
Android 1.0.2 (revision: 1)" complete.
Android 1.0.2 (revision: 1)" finished.
\AppData\Local\Android\Sdk\build-tools\26.0.2\package.xml
\AppData\Local\Android\Sdk\emulator\package.xml
\AppData\Local\Android\Sdk\extras\android\m2repository\package.xml
\AppData\Local\Android\Sdk\extras\google\m2repository\package.xml
\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager\package.xml
\AppData\Local\Android\Sdk\extras\m2repository\com\android\support\constraint\constraint-layout-solver\1.0.2\package.xml
\AppData\Local\Android\Sdk\extras\m2repository\com\android\support\constraint\constraint-layout\1.0.2\package.xml
\AppData\Local\Android\Sdk\patcher\v4\package.xml
\AppData\Local\Android\Sdk\platform-tools\package.xml
\AppData\Local\Android\Sdk\platforms\android-26\package.xml
\AppData\Local\Android\Sdk\sources\android-26\package.xml
```

Done

Figure 5.5 : Installation of Requested Components

In the next step i.e. *Add an Activity to Mobile*, click on *Empty Activity* thumbnail and then click *Next* as depicted in Figure 5.6.

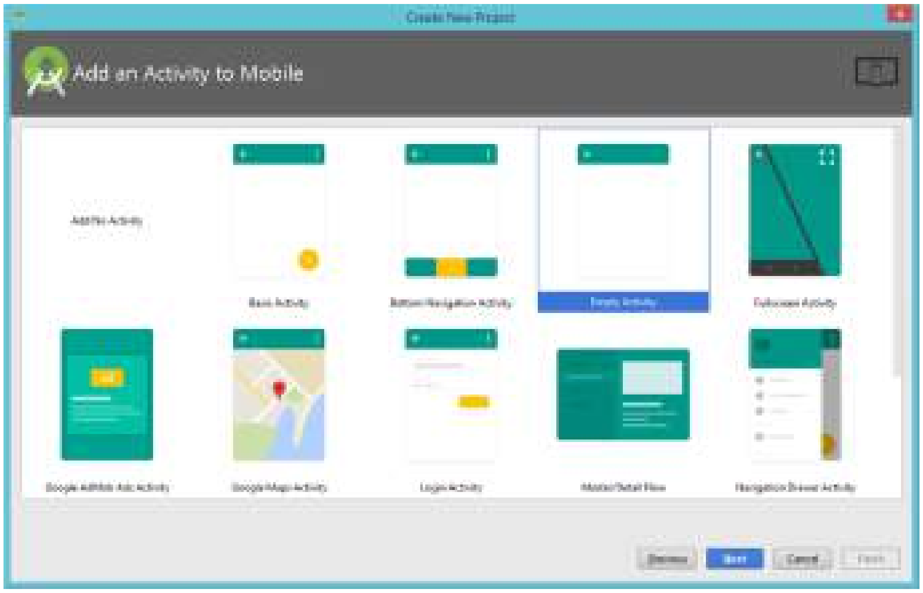
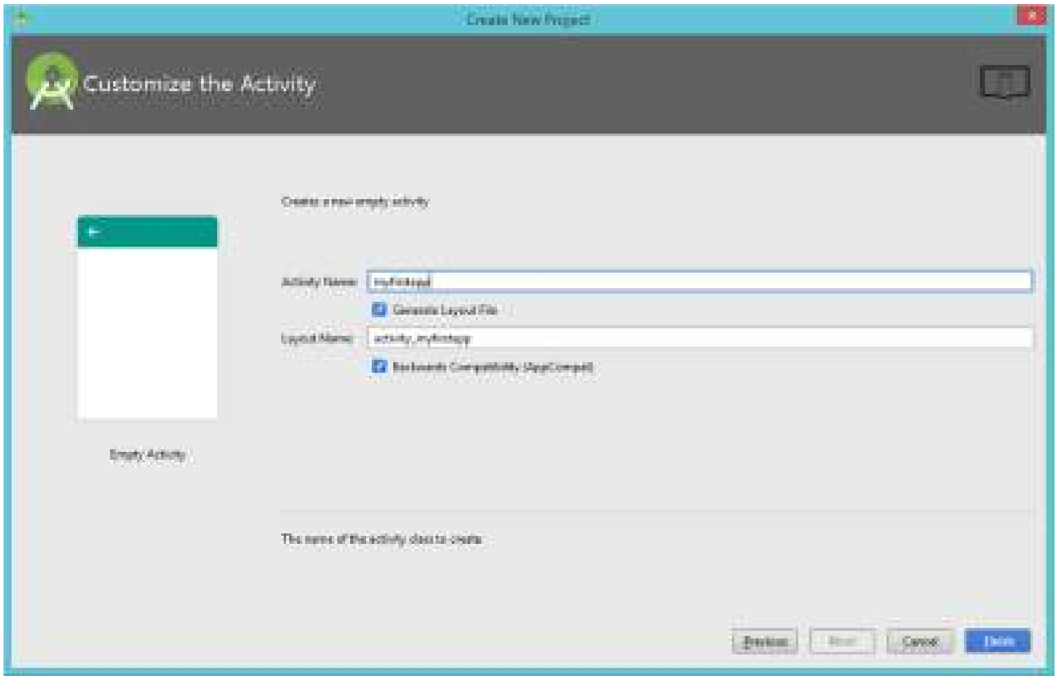


Figure 5.6: Adding an Activity to Mobile

In the next step, i.e. *Customize the Activity*, go with the default values or can give activity name and click *Finish* as shown in Figure 5.7.



Creates a new empty activity

Activity Name: myfirstapp

☑ Generate Layout File

Layout Name: activity_myfirstapp

☑ Backwards Compatibility (AppCompat)

The name of the activity class to create

Figure 5.7: Customizing the activities in Android Studio

After processing, Android Studio will start the IDE (Integrated Development Environment) as shown in Figure 5.8 which builds the project information.

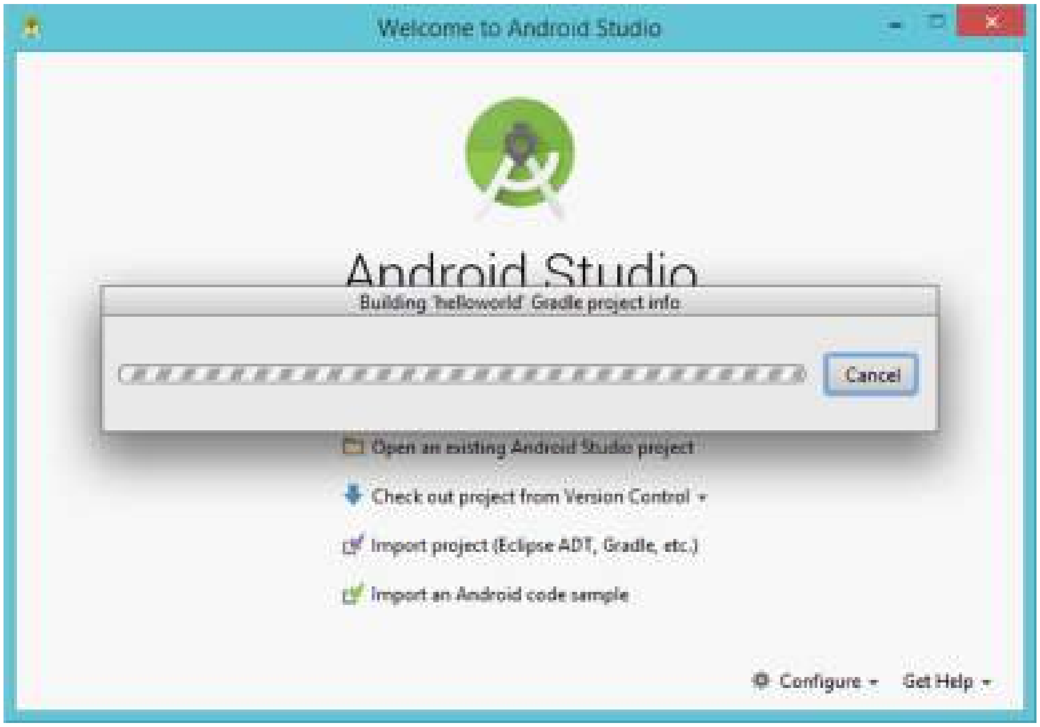
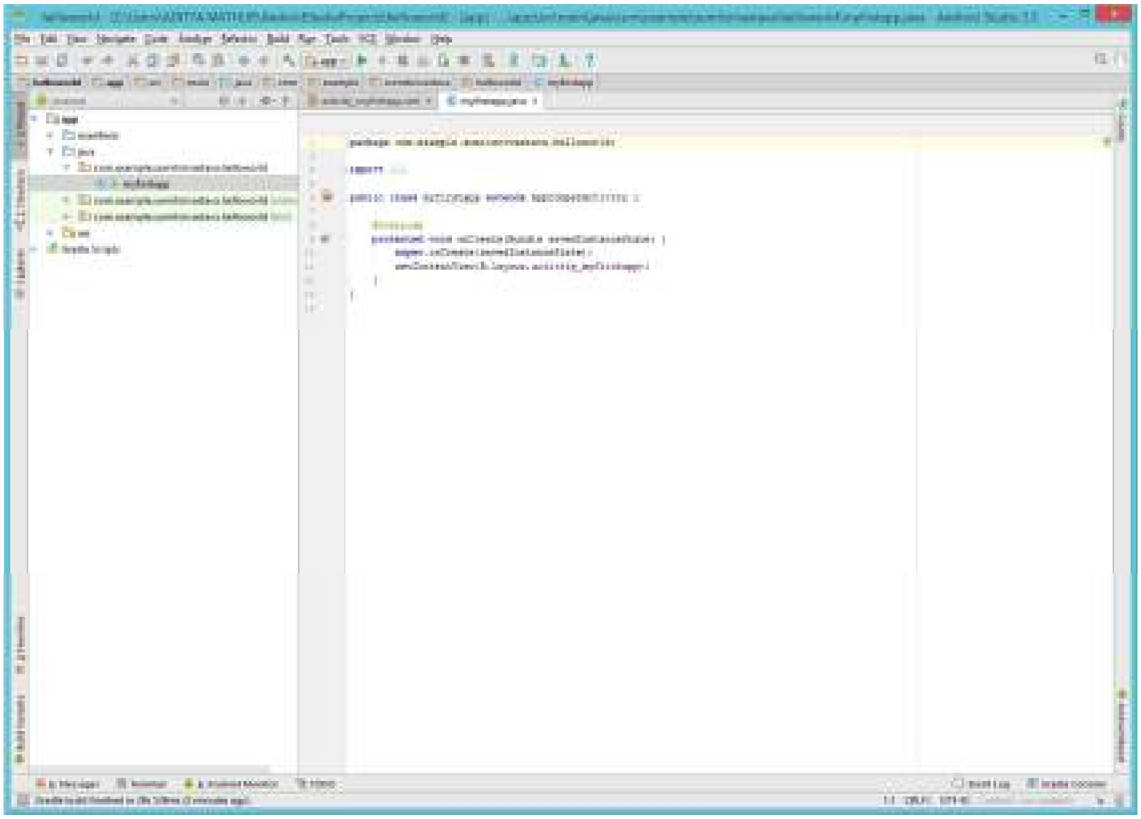

Welcome to Android Studio
Android Studio
Building 'helloworld' Gradle project info
Cancel
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Get Help

Figure 5.8: Building the project information

The code that was generated for the project is shown in Figure 5.9.



```
package [illegible];

import ...;

public class [illegible] extends [illegible] {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_myfirstapp);
    }
}
```

Figure 5.9: *helloworld* application

Now from the left navigation tree view, go to app > res > layout > activity_myfirstapp.xml as shown in Figure 5.10.

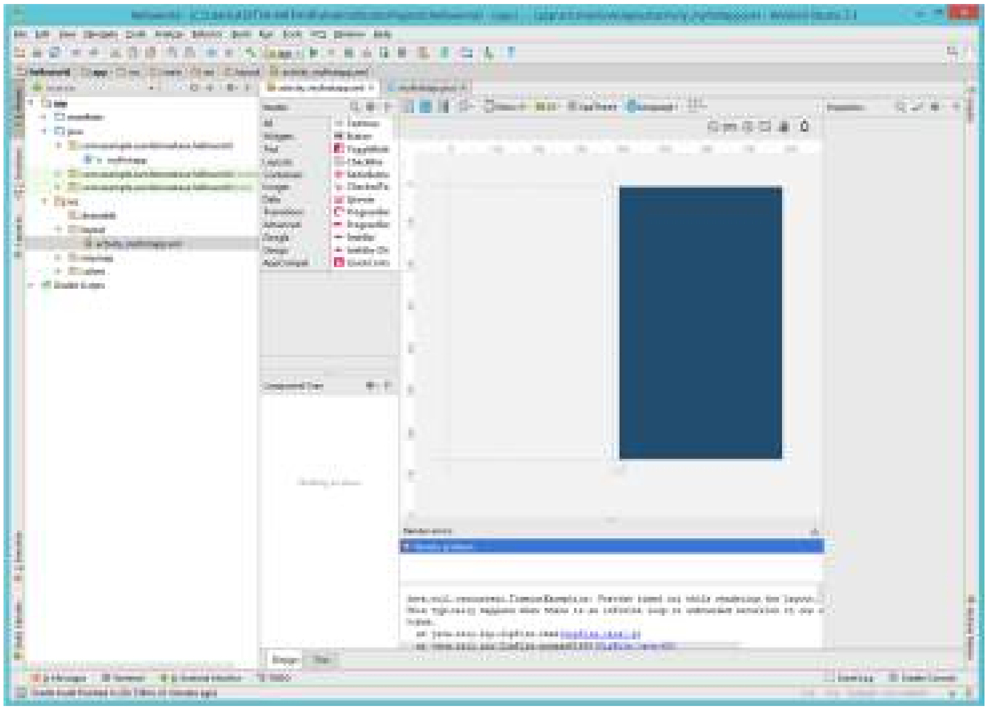


Figure 5.10: Palette in Android Studio

Now from *Palette* , drag the *TextView* element onto the layout as shown in Figure 5.11.

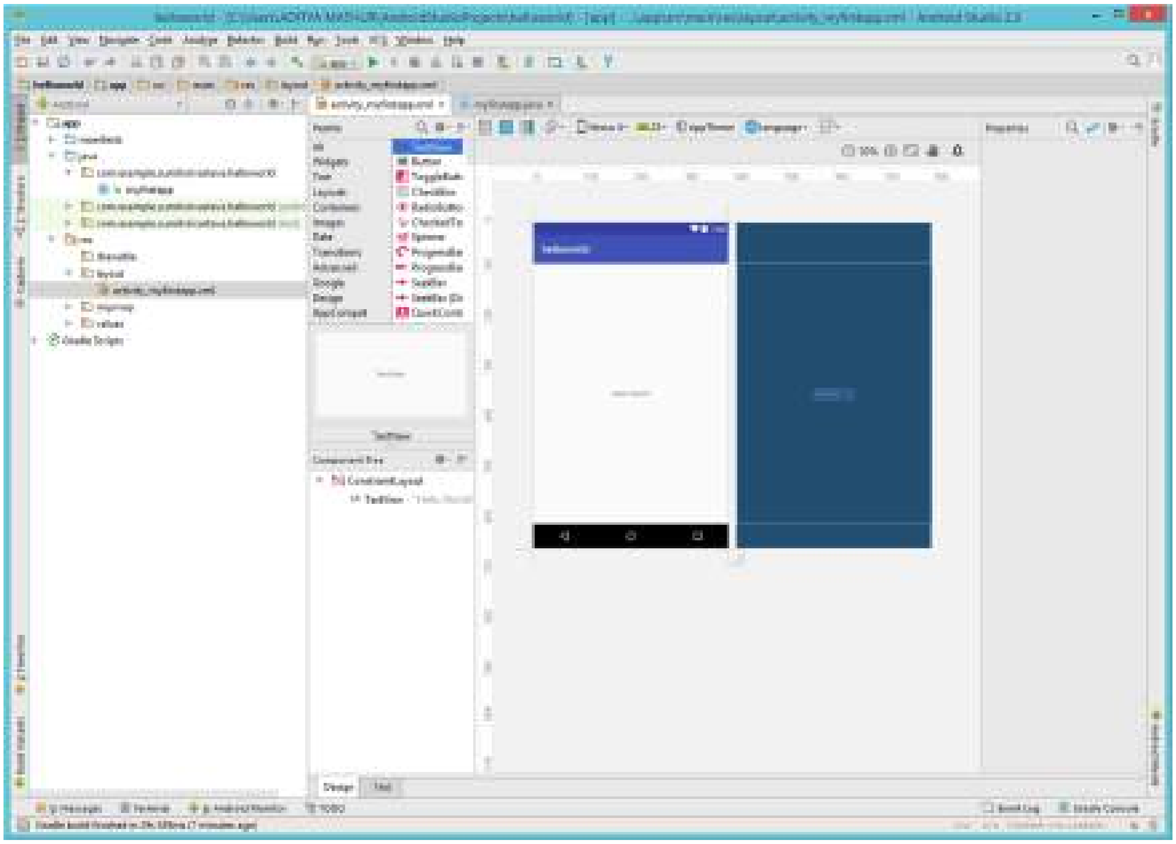


Figure 5.11: TextView in Palette

The following are the steps to setup your device:

- Associate your phone to your PC with a USB link. In case you're dealing with Windows, you have to introduce the USB drivers.
- Now, you have to enable USB debugging mode; Tap to Settings > Developer options.

In Android 4.2 and newer versions, by default developer option is hidden. To make it visible, go to *Settings>About phone> Tap Build number 7 times*. Now go back to the previous screen to find Developer options.

The following are steps to start app:

- Tap the app module given in the Project window and after that click Run > Run (or select Run from the toolbar).
- From the Select Deployment Target Screen, pick your device, and afterwards click OK.
- Now app will get installed on your device and afterwards it gets started automatically

That's *helloworld* app working on your device! (Figure 5.12).

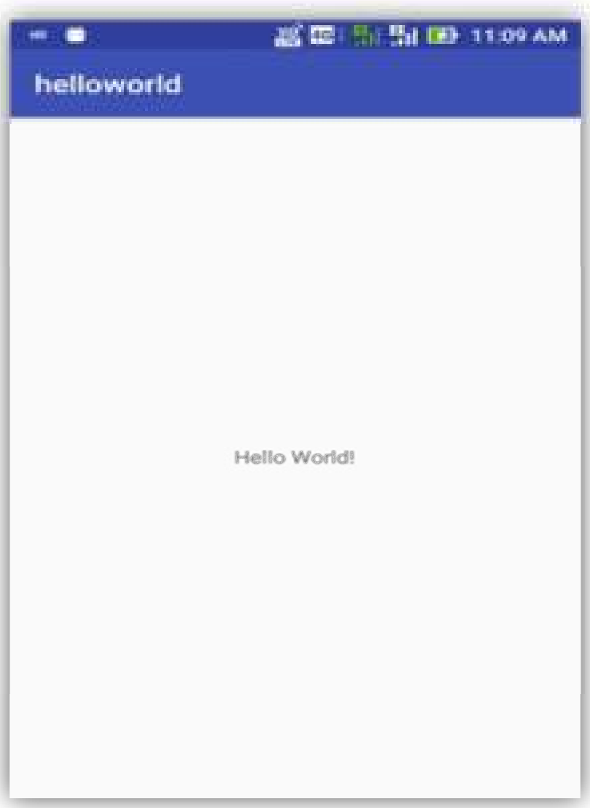


Figure 5.12: helloworld app runs

Table 5.2 lists some important files that are essential to run the app.

Table 5.2: Important files to run an app in Android

Files	Description
app > java > com.example.hello world> MainActivity.java	This is the primary point for your app. After build, when you run your app, the system initializes an instance of the current Activity which loads its layout.
app > res > layout > activity_ main.xml	This XML (Extensible Mark-Up Language) document characterizes the design for the UI movement's which contains a TextView component with the content "Hello World"
app > manifests > AndroidManifest.xml	This file depicts the essential elements of the application and furthermore characterizes each of its segments.
Gradle Scripts > build.gradle	You see two files i.e. one file for the project and one file for the "app" module. Every module has its own build.gradle file, but here you will find only one build.gradle file as but this project has just one module. Most of the time, you'll work with the module's build.gradle file to configure the development of your app.

5.4 MEMORY MANAGEMENT

Memory management includes managing RAM (random access memory) by moving processes back and forth between primary memory and Hard-disk at the time of execution of the program. It keeps an eye on each and every place of the memory, regardless of either it is granted to process or it is free to use. It verifies the amount of memory is to be granted to processes and also decides which process will obtain memory at what time. It keeps trail whenever some memory gets released or un-allocated and accordingly it updates the status.

Android Memory

Android is derived from a Linux based OS with 2.6.x kernel which is capable enough to handle most tasks in very well-mannered as it is deployed on native open source C libraries that have powered Linux machines for years. Basic operations of OS like process management, I/O management, memory management, and so on, are managed by the native stripped-down Linux kernel.

Using memory for each application

Memory for each application in android is managed by its own run time and virtual machine which is little unusual. Android run time also manages the process lifetimes and ensures application responsiveness by stopping and killing processes as necessary to free resources for higher-priority applications.

Every app runs in a separate process within its own Dalvik instance by giving responsibility for memory and process management to the Android run time.

Android run time stops and kills processes which are necessary to manage resources. Dalvik and the Android run time sit on top of a Linux kernel to handle low-level hardware interaction, while a set of APIs provides access to all of the under- lying services, features, and hardware.

The Dalvik Virtual Machine

It is defined as one of the crucial element of Android. Rather using a traditional JVM (Java Virtual Machine) like Java ME (Java Mobile Edition), Android utilizes its own customized VM(Virtual Machine) designed to manage multiple instances on a single device. Following are some function of the Dalvik VM:-

- It uses the Linux kernel to manage low-level functionalities like threading, process, security and memory management.
- As middle tier, it is used for managing hardware and system service access of android devices.
- It provides an abstraction layer to developers that ensure they never have to worry about a particular hardware implementation.
- It executes executable files (a compressed format) to ensure minimal use of memory.

Application Priorities and Process States

The sequence in which processes are killed to reclaim resources is basically determined by priorities given to hosted applications. The priority of an application is equal to its highest-priority component; the process that has a lower priority will be killed first. Priorities are also affected by inter-process dependencies; an app that has a dependency on another app will have at least as high a priority as the application it supports. All apps will remain running and in memory until the android needs its resources for other applications. Figure 5.13 gives details of states of the application states.

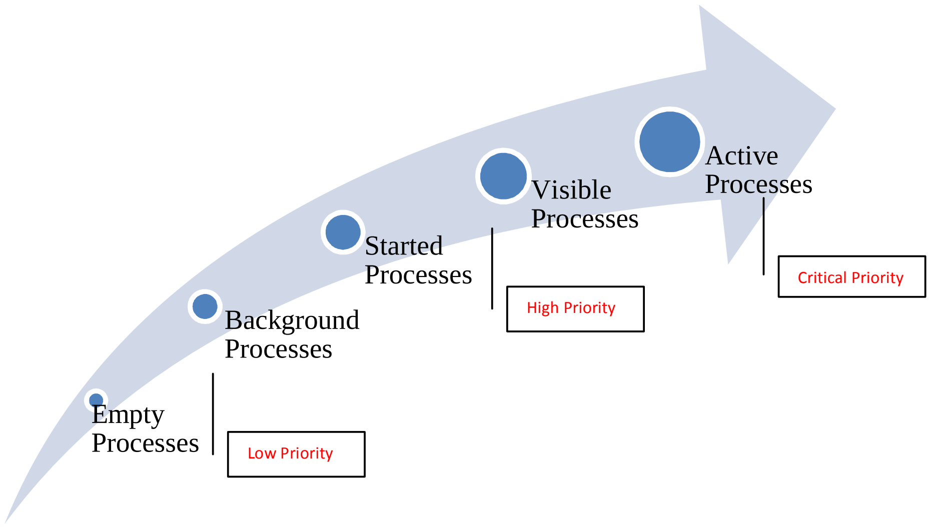


Figure 5.13: Details of Application States

Table 5.3 describes various states of processes:

Table 5.3: States of processes

Processes	Description
Active processes	Processes are active if they are in the foreground and responding to user events. Services which are executing an OnStart, OnCreate, or OnDestroy event handler.
Visible Processes	Processes are visible if they aren’t in the foreground or responding to user events. When an activity is only partially obscured like a non-full-screen or transparent Activity. Usually there are very few visible processes, and they’ll only be killed in extreme conditions to support active processes to continue.
Started Processes	These are the services that should be continued without a visible interface as services don’t interact directly with the user, they get a slightly lower priority than visible processes. They are still kept in the category of foreground processes and won’t be killed unless resources are required for active or visible processes.
Background Processes	Hosting processes which aren’t visible and that don’t have any services that have been started are considered background processes. Usually there will be a large number of background processes that will be killed using a last-seen-first-killed pattern to reclaim resources for foreground processes.
Empty Processes	For improving system performance, Android usually holds apps in memory after they have attained the end of their lifetimes. It maintains this cache to improve the start-up time of applications. These processes are regularly killed as needed.

Android also oversees opened applications which are running in the background. Iit closes the applications when the system needs memory. However, most of the android users are not fully satisfied with how it does its things because sometimes it keeps too many processes running which reduces the system performance. We can use task manager which does its job very well. I think, now, you might be clear with the concept of memory in Android.

5.5 VIRTUAL REALITY

A Computer technology that uses software programs for generating the realistic images, sounds and other sensations are known as virtual realities. In other word, an imaginary environment which simulates user physical presence is termed as virtual reality. It has been characterized as a reasonable and immersive simulation of a 3D domain, setup utilizing intuitive programming and equipment. The made virtual condition can be experienced or controlled by the action of the body or as an "immersive, intelligent affair created by a PC".A user having virtual reality gadget is normally ready to "glance around" the imitated world, move about in it and connect with the environment things that are portrayed on a screen or in goggles. In today’s world, virtual realities are depicted either on a computer monitor, a projector screen, or with a virtual reality headset which is also known as head-mounted display or HMD. Now, advanced haptic (capacity "to grasp something”) systems include tactile information i.e. sense of touch, usually known as force response in video gaming, medical and military training applications. Couple of virtual reality frameworks utilized as a part of computer games can transmit vibrations and different sensations to the user with the assistance of an amusement controller. Virtual nearness of users can likewise be accomplished through telexistence, telepresence or the utilization of a virtual artifact (VA).The artificial condition can be indistinguishable to the present world for making a genuine life experience. Recently, Google developed a virtual reality (VR) platform which is known by the name “Daydream” for *Android Nougat*, the 7th version of the Android mobile operating system. In May 2016, Google uncovered Daydream, a VR platform that works on at Smartphone and furthermore gives VR abilities through a virtual reality headset and controller composed by Google itself.

5.6 SUMMARY

Android is a robust Operating System that supports a large number of apps in Smart Phones. These apps make life easier and advanced for the users. Hardware that supports Android is mainly based on ARM architecture platform which is shown in Figure 5.14.

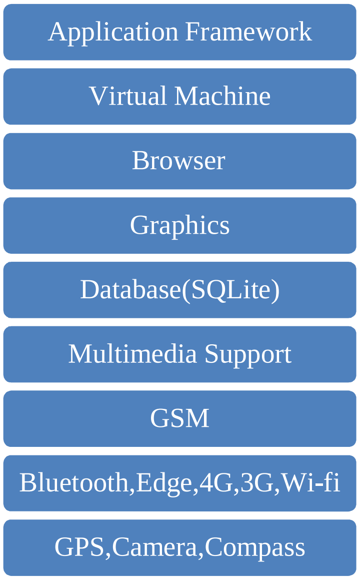


Figure 5.14: ARM architecture

Android mobile OS comes up with a pre-installed app store which is commonly known as Google Play Store. It allows Android mobile users to select, and download applications developed by third party developers and use them. There are around 2.8 million apps available on the market for users. Android apps are developed using Java. Android mobile OS is available as open source software for developers to develop apps. Android applications are made out of at least one application segments like activities, services, content suppliers, and broadcast receivers. Each segment has its own part to determine the general conduct of the application, and each one can be enacted individually (even by different applications). All components in the application must be declared by the manifest file including declaration of all application requirements, like minimum version of android and any hardware configurations.

Check Your Progress 1

1) Explain Android OS and its features?

……………………………………………………………………………..

……………………………………………………………………………..

2) Define memory management in android OS?

……………………………………………………………………………..

……………………………………………………………………………..

3) Briefly explain about android interfaces& applications?

……………………………………………………………………………..

……………………………………………………………………………..

4) What is virtual reality?

……………………………………………………………………………..

……………………………………………………………………………..

5) Develop an android app to display your bio-data?

……………………………………………………………………………..

……………………………………………………………………………..

5.7 FURTHER READINGS

- Hello, Android: Introducing Google's Mobile Development Platform Paperback – 3 Mar 2011 by Ed Burnette, ISBN-10: 9350232928, ISBN-13: 978-9350232927
- Android Essentials by Chris Haseman, ISBN: 1430217367, 9781430217367
- Android Recipes: A Problem-Solution Approach by Dave Smith (Engineer), Jeff Friesen, Publisher: Apress L. P., 2011, ISBN: 1430234156, 9781430234159
- https://developer.android.com/studio/index.html
- https://en.wikipedia.org/wiki/Android_Studio
- https://android-developers.googleblog.com/2017/10/android-studio-30.html

UNIT 6 BASICS OF iOS

Structure

6.0 INTRODUCTION

iOS is a mobile operating system developed by Apple. It was previously known as iPhone OS. This operating system runs on various devices such as iPhone, iPad, and iPod Touch etc. and it is available in 40 languages. It is second most popular operating system for handheld devices with 27% share globally. Most popular operating system is Android. iOS was originally developed for phones in 2007, further it is extended to support other devices such as the iPod Touch in September, 2007 and the iPad in January 2010. There are numerous versions of iOS which are released till date. The current version is iOS 10 which was released on September 13, 2016. It is supported on the iPhone 5 and above. It is also supported on iPad (4th generation), iPad Mini 2, iPad Pro and the 6th generation iPod Touch.This platform is written in C, C++, Objective-C, and Swift. It is very secure platform and it utilizes many security features both in hardware and software. Below are the security features of iOS:

- Secure boot
- Passcode
- Touch ID
- Non executable memory
- Encryption
- App Security
- Network Security
- Two-Factor Authentication

In iOS, there are four layers of abstraction as shown in Figure 6.1.

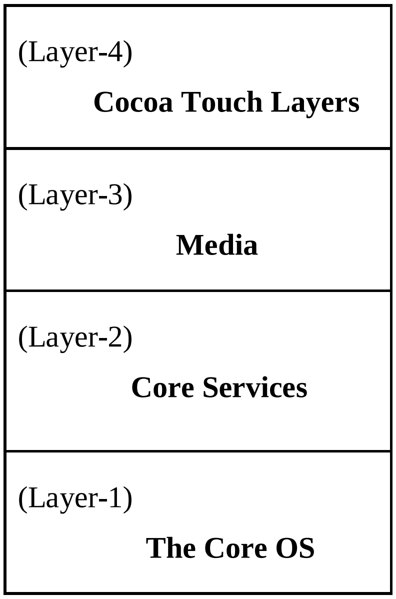


Figure 6.1: Layers of iOS

The user interface of iOS is somewhat different as compared to other Operating Systems of phones. It is based upon direct manipulation, using multi-touch gestures. Multi-touch gestures include swipe, pinch, tap, and reverse pinch. All of these have specific meaning and specific use within the context of iOS and multi-touch. There are various other control elements such as switches, buttons, sliders etc. Some applications in iOS use *accelerometers* to respond to shaking the device and three-dimensional rotating.

Table 6.1 lists some features of iOS

Features	**Description**
Home Screen	The Home Screen of iOS is given by **SpringBoard.** The Home Screen contains icons of different applications installed in the phone and it also contains a dock where a user can easily pin the most frequently used apps. There is a status bar at the top of the screen which displays current time, battery level of the phone, network strength of the connection, etc. Whenever the Home button is pressed, Home Screen will start to appear. If the device is protected with the passcode, then one has to enter that passcode first to unlock the device and after that only, the access to the Home Screen is granted. In iPhone OS 3, a new feature added named '**SpotLight**'. SpotLight allows users to search apps, contacts, media, messages, remainders, emails, etc. Later in iPhone 7, this feature of SpotLight is accessible by just pulling down anywhere on the home screen. In iOS 9, there are two ways by which we can use spotlight. In the latest version of iOS i.e., iOS 10, SpotLight is at the top of the "today" panel.
System font	System font of iOS is *Helvetica*. When iOS 3 is switched to iOS 4, its font changed to Retina Displays. With the release of iOS 7, users were provided with the ability to scale text or they can switch to Neue Bold. In iOS 9, the

	font was changed into San Francisco, which was originally developed for the Apple Watch.
Folders	The concept of Folders was first introduced with the release of iOS 4, in which two or more apps can be jiggled into one folder. Maximum number of apps that can be collected together in a folder is 12 on iphone 4S, 16 on iPhone 5, and 20 on iPad. The name of the folder is selected by the category of apps/files inside it, and user can also rename the name of the folder according to their ease. There can be nested folders also. Nested Folders means there can be a folder inside a folder.
Notification Center	Before the introduction of Notification Center, notifications were delivered in a modal window and cannot be viewed after being removed from the screen. With the release of iOS 5, Notification Center was introduced in which user can view the history of notifications. To view a notification, a user can simply tap on it and the corresponding app will be open in the phone.

6.1 OBJECTIVES

After going through this unit, you should be able to know

- basic features and functionalities of iOS;
- issues related to accessibility of iOS;
- the way iOS does multitasking; and
- about Siri and its usage.

6.2 ACCESSIBILITY

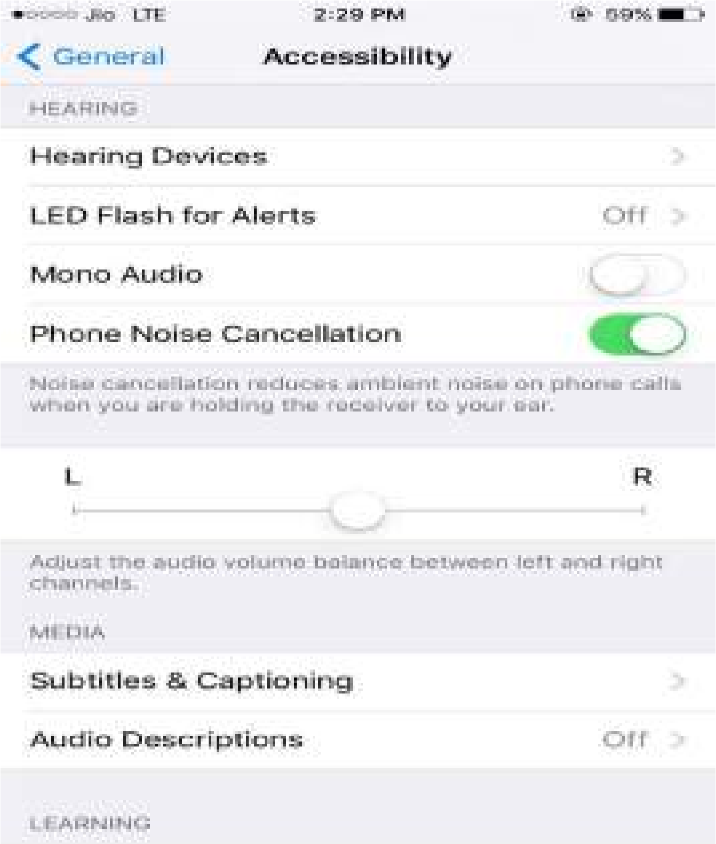
iOS enables users with vision and hearing disabilities to access its features. For example, visually impaired can communicate with their friends or relatives from overseas.

Table 6.2 gives some of the features of iOS for hearing impaired.

Features for Hearing impaired	
Features	**Description**
Facetime	This feature is a video conferencing system which is used on iOS as well as on Mac OS. With the help of this feature, users can do video call on two iOS devices like iPad, iPhone etc. There are many third party tools for video conference but FaceTime is the built in software of iOS and it is very well supported. It has the ability of lip reading and can converts to the text. For example - one user can use Facetime and on the other side second user can use braille display to read the text.

Closed Caption	It displays the text on screen while playing videos. There are many third party players for closed captioning but iOS has in-built player also that plays closed captioning. User has to turned it on and then video has to be closed captioned format that player understands.
Headphone Jack	Those users who need a higher volume level, can use headphone jack to add speakers. It is very useful because the built-in speakers are not loud enough.
Bluetooth Audio	User can also use Bluetooth Audio to listen in higher volume. Through Bluetooth speaker user can add amplification to the sound. Built-in speakers are not loud enough so user can increase the volume through Bluetooth speaker.
Messaging	Messaging is a great lifeline for those who have the hearing impairment. They can easily read the messages.
Visual Alerts	For different applications, user can save the different settings for visual alerts. Alert pop up right in the middle of the screen. Those who has hearing impairment, they could have problem in listening the notification sound, to solve this problem visual alerts is a great way to display the notifications.

The accessibility (Hearing) feature settings are shown in Figure 6.2.



•○○○○ Jio LTE 2:29 PM 59%

< General Accessibility

HEARING

Hearing Devices >

LED Flash for Alerts Off >

Mono Audio ○

Phone Noise Cancellation ●

Noise cancellation reduces ambient noise on phone calls when you are holding the receiver to your ear.

L R

Adjust the audio volume balance between left and right channels.

MEDIA

Subtitles & Captioning >

Audio Descriptions Off >

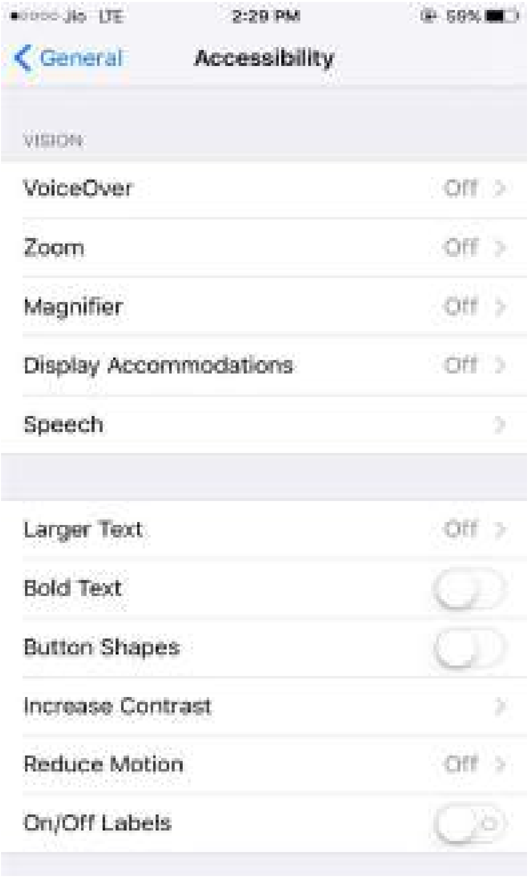
LEARNING

Figure 6.2 : Accessibility (Hearing) feature settings in iOS

Table 6.3 lists features for visually impaired.

Features for Visually impaired	
Features	**Description**
Voice over	If a user is completely blind or partially blind then this technology is a boon for him. It reads the content of device and can navigate the user that where he is onscreen. For example- if user receives a message then it reads back, so that user can listen the message.
Siri	Siri is based on technology that is called Speech to Text. User has to touch the Siri button and speak something then Siri translates voice into the text and then executes the command accordingly. With the help of Siri, user can launch the applications, ask about the weather and get bring it back in voice form. This feature is very helpful for people with disabilities or without disabilities. Many people use Siri on daily basis.
Zoom	User has to turned on this feature, by default it sets to off. Once it is turned on, then any portion of the screen can be zoom in. If there is enough content on the screen then user can zoom it and can read easily. This is very useful for people who are partially blind.
Black on White	This feature has to be turned on as it sets to off by default. It is also called as invert colors. The normal screen is of the white background and black color text on it. In Black on White feature screen background will be of Black color and white color text on it. This is very helpful for those who need high contrast screens.
Speak Selection	User can highlight the text, then a button appears “Speak” and it reads back the selected content to the user. It speaks in the default device language.
Large Text	User can enlarge the text up to 56 points. It is helpful in reading the text easily. It is difficult to use it on small screen devices.
Braille Display	Braille Display is like a Braille Printer. iOS device sends the some text to the display and then it displays the first line of text. Braille displays can have 20 characters or 80 characters. It is a very good interface for braille proficient. This is very helpful for those who are deaf or blind. Braille displays connected to iOS devices via Bluetooth i.e. wirelessly.

The accessibility (Vision) feature settings are shown in figure 6.3.



Jio LTE 2:29 PM 59%

General Accessibility

VISION

VoiceOver Off

Zoom Off

Magnifier Off

Display Accommodations Off

Speech

Larger Text Off

Bold Text ○

Button Shapes ○

Increase Contrast

Reduce Motion Off

On/Off Labels ○

Figure 6.3 : Accessibility (Vision) feature settings in iOS

6.3 MULTITASKING

Multitasking is a concept of performing multiple tasks (also known as processes) over a certain period of time by executing them concurrently. New tasks start and interrupt already started ones before they have reached completion, instead of executing the tasks sequentially so each started task needs to reach its end before a new one is started. As a result, a computer executes segments of multiple tasks in an interleaved manner, while the tasks share common processing resources such as central processing units (CPUs) and main memory. In iOS based devices, user can use more than one app at a time with the help of multitasking. With the release of iOS 4, the functionality of Multitasking was introduced in June 2010. This functionality was limited to fewer devices such as iPhone 4, iPhone 3GS, iPod Touch (3rd generation). Now, multitasking is supported on almost all latest devices like iPhone 3GS+, iPod Touch 3rd generation +, and all iPad models.

After the release of iOS 4, multitasking is supported in seven background APIs:

- Voice over IP
- Background audio
- Background location
- Local notifications
- Push notifications

- Task completion
- Fast app switching

After the release of iOS 5, three more APIs were added:

- Newsstand
- Bluetooth Accessory
- External Accessory

With the release of iOS 7, Apple introduced a new feature of enabling the apps to perform background updates. Most frequently used apps of the device are given high preference to update and it is preferred to use Wi-Fi network instead of cellular network to update the apps.

6.3.1 Switching Applications

In iOS, user can easily switch from one app to another. Following are steps to switch from one app to another.

- Double-Click on the Home Button to see recently used apps.
- Swipe Right or Left to find the app that you want to use.
- Tap the desired app.

Switching in iOS gets updated with every new version released. Like in iOS 4 to iOS 6, the application switcher is activated by double-clicking the Home Button. After this, a scrollable dock-like interface starts appearing in the bottom that let the contents of the screen moving upwards. User can choose any icon to switch to that particular application.In iOS 7, the application switcher gets activated by double-clicking the Home Button. There are some changes now as compared to previous versions. Now, the screenshots of open applications are displayed on the top of the icon and user can browse through the apps by horizontal scrolling, and user can close any application by dragging them up.With the release of iOS 9, there is a significant change in the visualisation of application switcher but the concept of showing screenshots is same as that of iOS 7. In iOS 9, the size of the application icon is relatively small, and it appears above the screenshot, and each application card overlaps the other.

6.3.2 Ending Tasks

User can easily end the tasks running in the phone. In iOS 4 to iOS 6, to quit/end the applications running, user have to hold the icons appearing in the application switcher, then tap on the red minus circle that appears at the corner of the app’s icon. With the release of iOS 7 and further, the process of ending tasks become faster and easier. In iOs7, user don’t have to hold the icons to close them, indeed they can swipe up the icons to close them. At Most three apps can be cleared at a time in iOS 7.

6.4 SIRI

Siri is Apple’s personal assistant that takes input in voice, processes the function and returns back the output in voice format. User can ask the questions to Siri, or can ask Siri to do things. Siri is an essential part of iOS since iOS 5. On 14 October, 2011, it was first introduced as a feature of the iPhone 4S. Now, it has been included on all iOS devices after October 2012. It is also integrated with Apple Watches which can be activated just by saying “Hey, Siri”. It is also integrated into Apple’s TV and in cars also.

6.4.1 Setting Up Siri

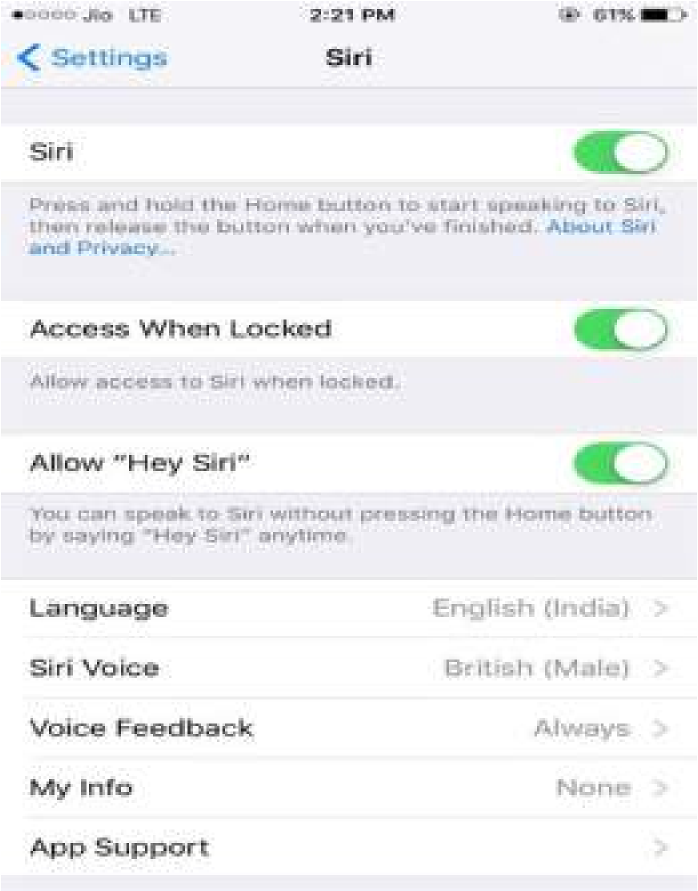
The following are steps to enable Siri in iOS devices:

i) Open the setting in the iOS device.

ii) Scroll down and click on Siri.

iii) Then, user can toggle the switch on or off and user will be asked to confirm to enable or disable Siri.

In *Settings*, Siri needs to be enabled (Figure 6.4).



●○○○○ Jio LTE 2:21 PM 61%

< Settings Siri

Siri

Press and hold the Home button to start speaking to Siri, then release the button when you've finished. About Siri and Privacy...

Access When Locked

Allow access to Siri when locked.

Allow “Hey Siri”

You can speak to Siri without pressing the Home button by saying "Hey Siri" anytime.

Language English (India) >

Siri Voice British (Male) >

Voice Feedback Always >

My Info None >

App Support >

Figure 6.4 : Enabling Siri

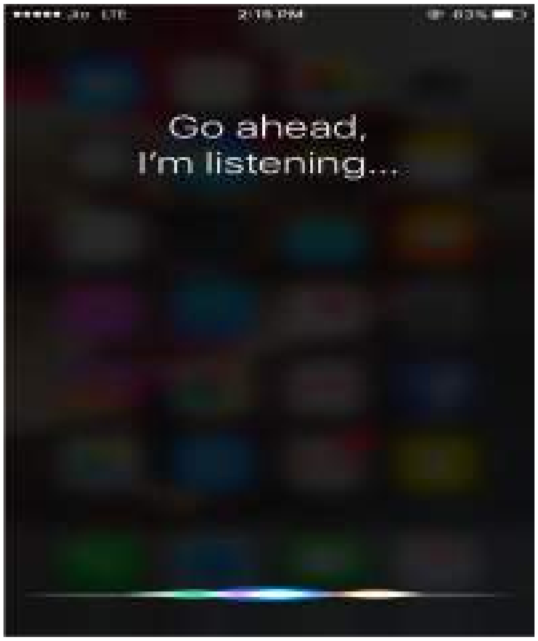
6.4.2 Launching Siri

The following are steps to launch Siri:

i) First, press and hold the Home button on iOS Device, and say "Hey, Siri" to activate Siri.

ii) Ask Siri to do something or give some command.

iii) Then, Siri will perform the action (Figure 6.5)



Go ahead,
I'm listening...

Figure 6.5: Siri in action

6.5 GAME CENTER

Game center is a social gaming network that supports online multiplayer. Players can share the games online. On July 10, 2008 Apple Inc. launched the App Store but it had no unified multiplayer and social system on that platform. Then, many third party apps like OpenFeintPlus AGON Online and Scoreloop came into the light and control over the online gaming environment.Apple announced Game Center on 08 April, 2010. It features turn based gaming, player profile photos, friend suggestions and achievement points. With iOS 6 it updates the game center and some features were added like game challenges, beat leaderboard scores etc.

Players can connect with friends, can send friend requests, playing games and organize multiplayer games. 500 friends can be connected to a single Game Center. Some games contains achievements, some have certain tasks to be completed and player earn some reward points also. In some games, players can see the leaderboards so that they can compare their score with their connected friends.

Table 6.4 lists some features of Game Center.

Features	Description
Achievements	Players can earn points as reward by meeting specific game-in challenges. It was basically developed to socialize and create competition between players.
Leaderboard	Some games have features the leaderboard through which players can compare their score with their connected friends.
Multiplayer	Few games host the matches between the players. Players can be their friends or can be anyone from around the world.

Users have to set their profile in Game Center. Following are the steps to setting up the profile:

Settings > Game Center > Sign In

Then, user should sign in with their Apple ID.

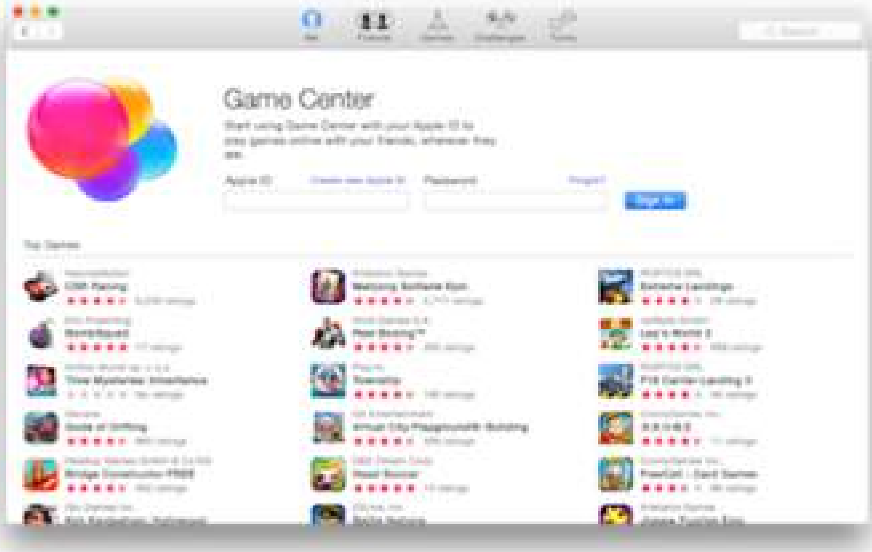


Figure 6.6 : Interface of Game Center

6.6 SUMMARY

iOS is an operating system specially developed for mobile devices. Apple Inc. first launched it in 2007 and then many releases were rolled out. iOS has many smart features, visually and hearing impaired person can use it in a very ease manner. Disabled people can operate the iOS devices by their voice only and it has the great feature of Braille Display through which blind person can operate the phone.

iOS is compatible with many devices like:

- iPad
- iPad Mini
- iPad Pro
- iPhone
- iPod

There is an online store, developed by Apple Inc. called *App Store*. In *App Store*, users can find the apps which are compatible with their iOS device.

Check Your Progress 1

1) Describe security features of iOS.

…………………………………………………………………………………

…………………………………………………………………………………

2) Enlist various changes made in the home screen of iOS with different versions.

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

3) Explain the features of iOS which enable a visually impaired person to use this platform.

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

4) Explain the features of iOS which enable a hearing impaired person to use this platform.

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

5) What is switching? Explain it in the context of iOS.

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

6) List different features of Game Center.

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

7) Mention various steps needed to set up Siri.

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

………………………………………………………………………………..

6.7 FURTHER READINGS

- *Take Control of iPhone Basics, iOS 4 Edition by Karen G. Anderson, publisher: TidBITS(2011), ISBN: 1615422013, ISBN:9781615422012*
- Apple iOS and iPhone Basics: Expert Advice, Made Easy, by Chris Smith, James Wallace, Contributor: Mark Mayne, Publisher: Flame Tree Publishing (2015), ISBN: 1783613947, ISBN: 9781783613946.
- https://upload.wikimedia.org/wikipedia/en/4/44/Game-center-macapp.png
- https://en.wikipedia.org/wiki/IOS
- https://developer.apple.com/game-center/
- https://en.wikipedia.org/wiki/Game_Center

UNIT 7 BASICS OF WINDOWS MOBILE

Structure

7.0 INTRODUCTION

Microsoft developed *Windows Mobile* Operating System for Smart phones and Pocket PCs so that those windows users can access the windows features on their handheld devices. It was originally developed as Windows CE in 1996. However, Windows Mobile first appeared in 2000 as Pocket PC 2000. 'Pocket PC' came to be known as Windows Mobile in 2003 after which it started coming in different versions as that of Windows and was mostly used by business and enterprise consumers. It became the most popular smart phone in U.S. until 2007.But afterwards; it started losing its popularity. After the launch of other operating systems like iOS and Android, in February 2010, Microsoft announced Windows Phone to supplant Windows Mobile. After that, Windows Mobile has been deplored. The last version of Windows Mobile was released after the announcement of Windows Phone. That last version is 6.5.5. But, this version was not able to run in newly developed Windows Phone because Windows Phone was not compatible with Windows Mobile devices and software. Microsoft finally stopped developing the Windows Mobile and started developing Windows Phone only. Windows Phone basically aimed at the consumer market instead of enterprise market.

The following is the list of different versions of Windows Mobile and Windows Phone:

- Windows CE
- Pocket PC 2000
- Pocket PC 2002
- Windows Mobile 2003
- Windows Mobile 2003 SE
- Windows Mobile 5
- Windows Mobile 6
- Windows Mobile 6.1

- Windows Mobile 6.5
- Windows Phone 7
- Windows Phone 8
- Windows Phone 8.1
- Windows 10

7.1 OBJECTIVES

After going through this unit, you should be able to know

- features of Windows Phone;
- about the development for Windows Phone;
- about evolution of Windows Phone; and
- about various releases of Windows phone.

7.2 DEVELOPMENT

Windows Phone is successor of Windows Mobile and Zune (It was the digital media product of Microsoft). Windows Phone shows a new user interface which is developed in *Metro Design Language (MDL)*. Metro is the geometry-focused design language created by Microsoft especially for user interfaces. Windows Phone is specially designed for Consumers instead of enterprise market. It was first come in the market in October 2010 with its first version as Windows Phone 7. Now, the latest version of windows OS in phone is Windows 10 which was released in the end of year 2015.

7.2.1 Evolution of Windows Phone

In 2004, a major update have begun in windows mobile which was named as 'Photon', but the development of this update was very slow and ultimately, the project got stopped. In 2008, Microsoft re-constructed the windows mobile team and started working for the development of a new mobile operating system. In 2009, fresh operating system was released as Windows Phone but due to unexpected shortfalls or delays, Microsoft released Windows Mobile 6.5 as an interim version. Within small span of time, Windows Phone was developed. But, new operating system was not congruent with Windows Mobile apps. According to a Product Manager of Microsoft, they were lacking with the time and resources, if they would have sufficient time and resources they might be able to solve the issues of compatibility. He said, they were attempting to make the Windows Phone available for both end users and enterprise network and also trying to go ahead with capacitive touch screens rather using stylus based screen. On February 15, 2010, Mobile World Congress in Barcelona, Spain announced Windows Phone 7 which was officially released on November8, 2010in the consumer market of United States. In 2011, Microsoft announced Windows Phone 7.5 *Mango* that has the power of Internet Explorer 9; which provides the similar web standards and graphical interface as the desktop versions. In 2012, Microsoft released a minor update which was known by the name *Tango.* This minor update fixed the existing bugs and reduced the hardware requirements; which supported the devices with lesser configuration to run windows mobile operating system. In February 2011, a partnership was signed between Microsoft CEO Steve

Ballmer and Nokia CEO Stephen Elop at a press event in London. In this partnership, it was decided that the windows mobile operating system would be the primary operating system for the Smart phones which are being developed by the Nokia. This partnership replaced the Nokia Symbian OS with windows mobile OS. Both Nokia and Microsoft were aimed at setting up a new mobile platform to compete with Android and iOS. In October 2012, Microsoft introduced Windows Phone 8 as a new generation of the mobile operating system. It was replaced by the previous Windows CE–architecture; which was derived from Windows NT Kernel with various modules shared with Windows 8.In April, 2014, Windows Phone 8.1 was being released. There were some new features added via a notification center , separate volume controls, Internet Explorer 11.Similar to Apple iOS , Siri and Google Now in Android**,** Microsoft introduced anew called CORTANA in its Windows 8.1 Mobile Phone. In January 2015, Microsoft announced Windows 10 Mobile phone where Bing search feature was replaced by the CORTANA, a voice assistant. The Operating System of this mobile is for smart phones and tablets running on ARM Architecture (Figure 7.1).

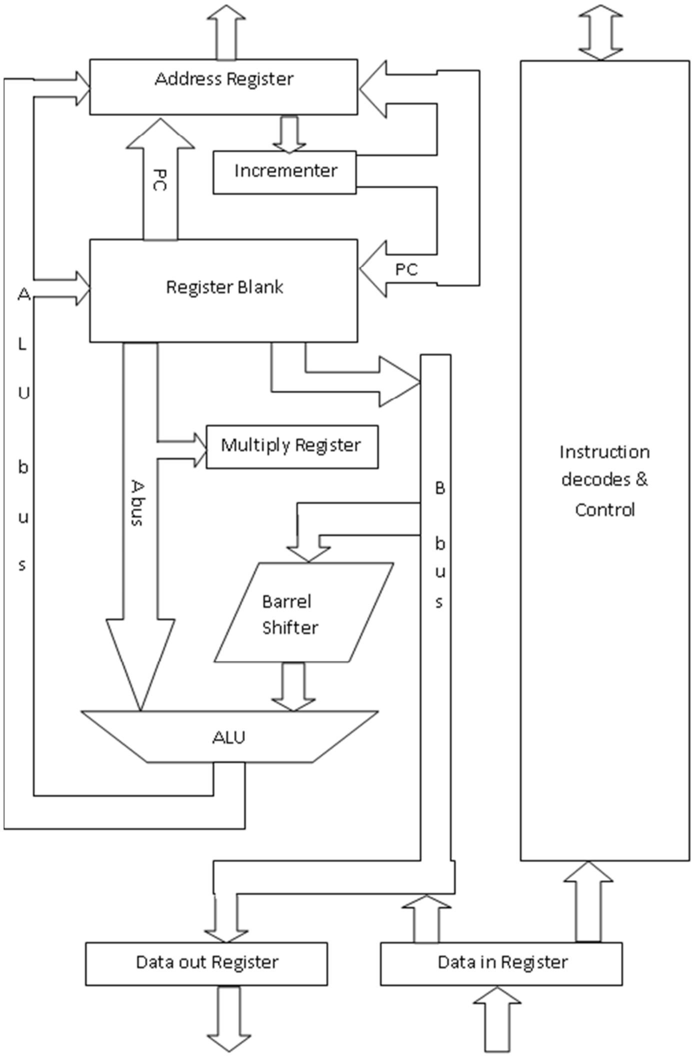


Figure 7.1 : ARM Architecture

7.3 FEATURES OF WINDOWS MOBILE

In this section, let’s discuss features of Windows Mobile.

User Interface

Windows Phone shows a new user interface which is developed in MDL. The home screen of windows mobile phone depicts the “Live Tiles” i.e. the thumbnail or image links to applications functions & features; third party applications such as Web pages, contacts etc are also linked up with these tiles on the home screen. Tiles can be added, rearranged or removed by the end users. Tiles get updated automatically and change its status dynamically. For example- the text message tile would show the list of unread messages. User can also resize live tiles to either a small, medium, or large appearance. There is a feature in windows mobile phone called ‘Hubs’, which is used to integrate the local as well as online content with the help of windows phone’s integration technique. It is used to connect windows phone with commonly known social networking websites such as Facebook, WhatsApp Twitter etc. For example- the people hub is used to fetch contacts from various sources like Facebook , Gmail and windows live. Similarly picture hub is used to show photographs captured using device camera and the user photo album from Facebook. Users can also perform different types of activities on social networking websites using Hubs.

The user interface of Windows Phone is shown in Figure 7.2.

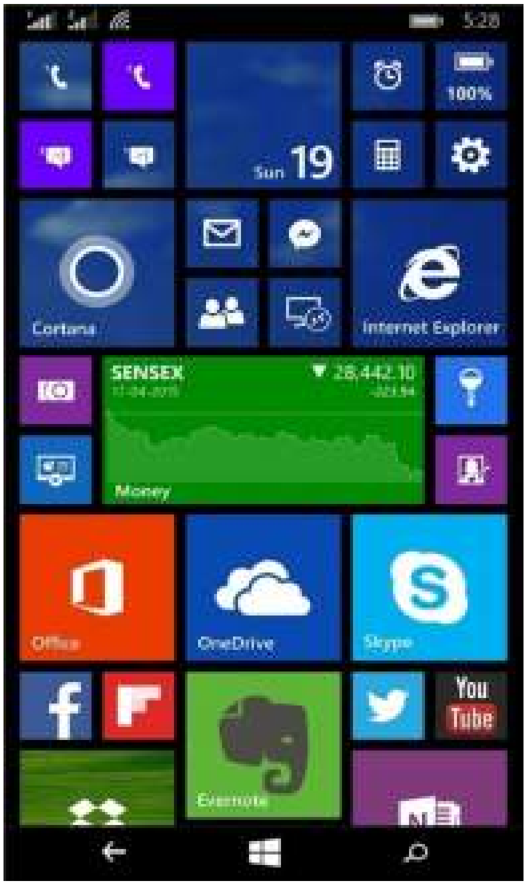


Figure 7.2: User Interface of Windows Phone

Multitasking

Multitasking refers to an idea of executing multiple tasks or activities concurrently or at the same time. In Windows Mobile Phone, multitasking feature can be accessed by long pressing the "back" arrow. A card-based task switcher is used in windows mobile phone 7, whereas later versions make use of true background multitasking feature.

Contacts

End users can save their contacts in Windows Phone with their Mobile Phone, Email, and Contact Image etc. It also synced with other sites like Facebook, Windows Live Contacts, Twitter, LinkedIn, Google, and Outlook etc. This is also called as “People Hub”. User can manage their contacts by divide them into different groups.

Text Input

Windows phone has on-screen virtual keyboard which also contains the facility to insert emoticons. It has in built spell checker and word prediction which can speed up the typing of user. While user types something, it shows the list of similar words from which user can select the particular word and it will insert automatically. It has the swipe typing feature also which avail the user to type any word without lifting the finger.

Web Browser

Internet Explorer is the default web browser in Windows Phone (Figure 7.3). It enables the user to maintain the list of frequently accessed web pages on the Start screen. It can load upto 6 tabs simultaneously. It also supports multi-touch gestures, zoom in and zoom out of animations etc. User can save the pictures which are available on web page and can share web pages via email. It supports searching the word or phrase in a web page.

Figure 7.3: Microsoft Internet Explorer

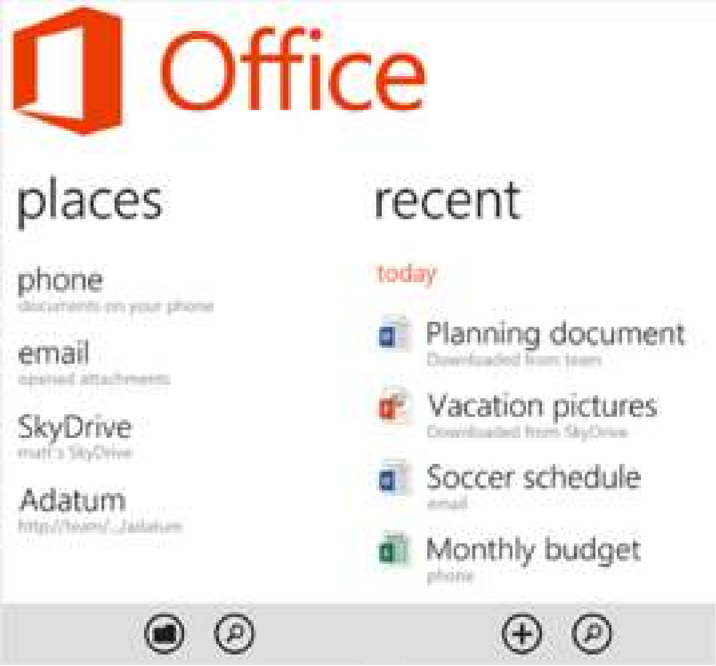
Media Player

Media player is used for playing multimedia files like videos, movies and music. It displays standard media control icons such as play (▶), pause (⏸), fastforward, backforward, and stop (■) buttons. Every Operating System has one built-in media player. Like, windows come with windows media player, Android OS comes with Google Play Music by default.

Office Suite

Microsoft office is the default office suit in Windows Phone which supports synchronization between the desktop version of Microsoft Office and Windows

Phone (Figure 7.4).User can create or edit the documents, reports, presentation etc with the help of Word, Excel and Power point respectively on the go.


Office

places

phone
documents on your phone

email
opened attachments

SkyDrive
matt's SkyDrive

Adatum
http://team/.../adatum

recent

today

Planning document
Downloaded from team

Vacation pictures
Downloaded from SkyDrive

Soccer schedule
email

Monthly budget
phone

Figure 7.4: Microsoft Office Suite in Windows Phone

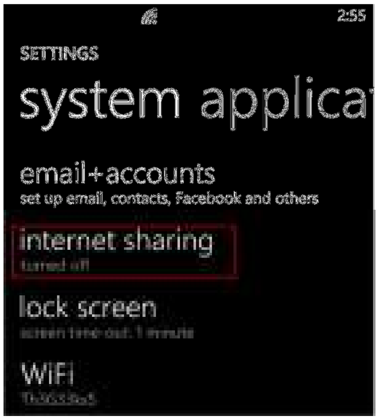
Shared Internet Connection

It is Windows administration that empowers one internet enabled mobile phone to impart its internet connection with other mobile phones. Internet connection can be shared using Windows Phone.

The following is the process to share Internet connection in Windows Phone:

Keeping in mind the end goal to utilize your Windows phone as a versatile Wi-Fi hotspot and offer your internet connection, you'll require an active internet data plan in your phone.

Figure 7.5 shows the screen on tapping *Settings*.


2:55

SETTINGS

system applica

email+accounts
set up email, contacts, Facebook and others

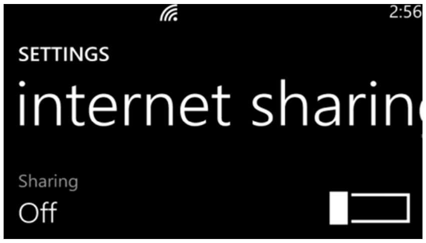
internet sharing
turned off

lock screen
screen time-out: 1 minute

WiFi
[illegible]

Figure 7.5: After tapping *Settings*

Then, tap Internet sharing (Figure 7.6).



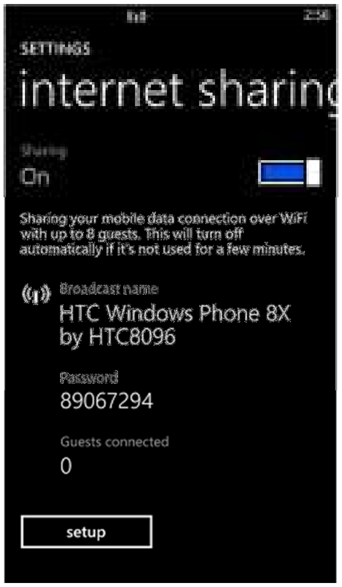
SETTINGS

internet sharin

Sharing
Off

Figure 7.6: After tapping Internet sharing

Move the slider from Off to On. This will automatically turn on your device's Wi-Fi (Figure 7.7).



SETTINGS

internet sharin

Sharing
On

Sharing your mobile data connection over WiFi with up to 8 guests. This will turn off automatically if it's not used for a few minutes.

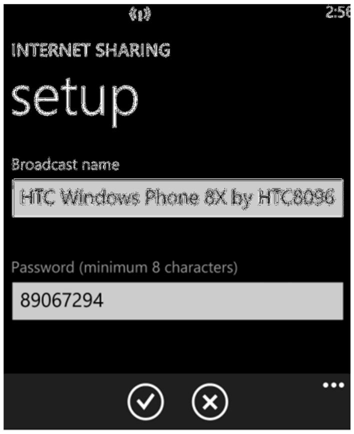
Broadcast name
HTC Windows Phone 8X by HTC8096

Password
89067294

Guests connected
0

Figure 7.7 : Wi-Fi switched ON

Now Tap setup to configure Internet Sharing Setup.



INTERNET SHARING

setup

Broadcast name
HTC Windows Phone 8X by HTC8096

Password (minimum 8 characters)
89067294

Figure 7.8 : Save and Exit

You can modify the default broadcast name and specify the password. Tap the right checkmark for saving your changes, and tap X to get exit (Figure 7.8).

7.3.1 Virtual Private Networking

In Windows Phone, Virtual Private Network can be established over PPTP protocol. The User Interface of Windows Phone continues to change with different versions. The display screen of the phone was later known as the "Home Screen", and shows the current date, time, relevant information, and emails etc.

The following are the steps to configure VPN profile of Windows Phone 8.1:

i) Tap Settings. Choose VPN. Move slider to turn on the VPN status. Choose+add (Figure 7.9).

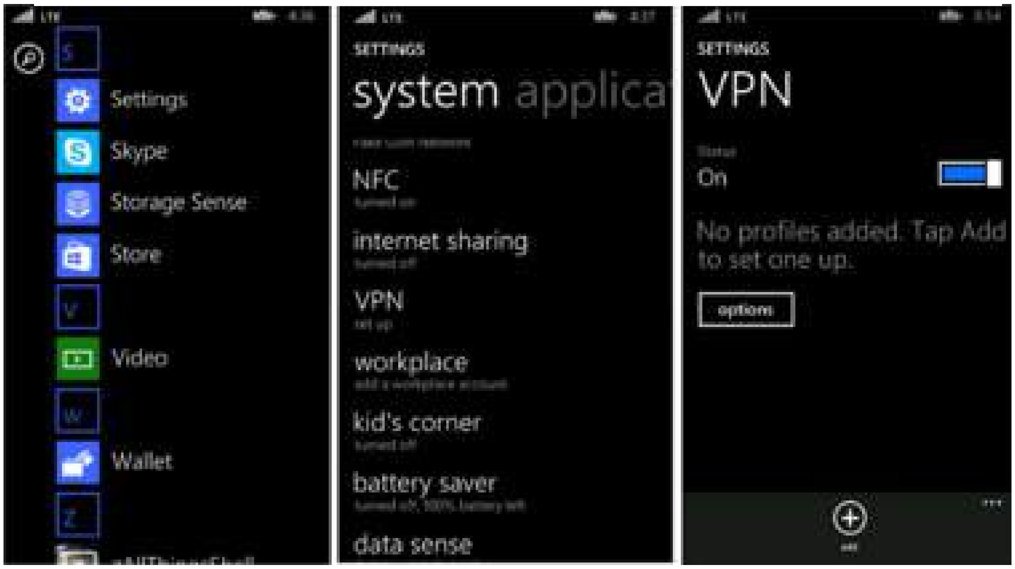
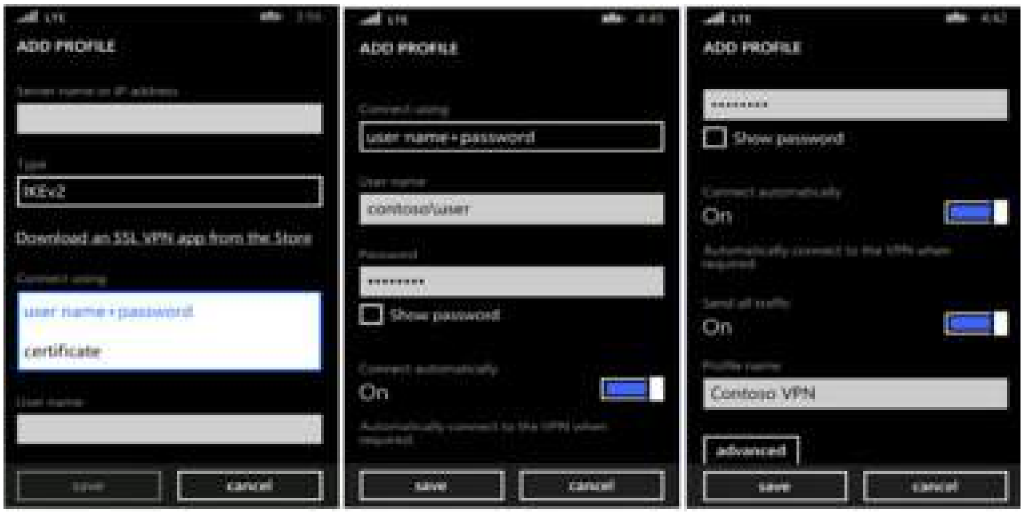


Figure 7.9: Turning ON VPN from *Settings*

ii) Now, type the server name or IP address. Choose IKEv2. For SSL-VPN, click the link shown there and download a third-party plug-in.

iii) For setting up a password for your VPN, choose user name+password. Provide the required details. To provide more details, choose Advanced. Now Tap Save (Figure 7.10).



ADD PROFILE

IKEv2

Download an SSL VPN app from the Store

user name+password

certificate

ADD PROFILE

user name+password

contoso\user

□ Show password

On

ADD PROFILE

□ Show password

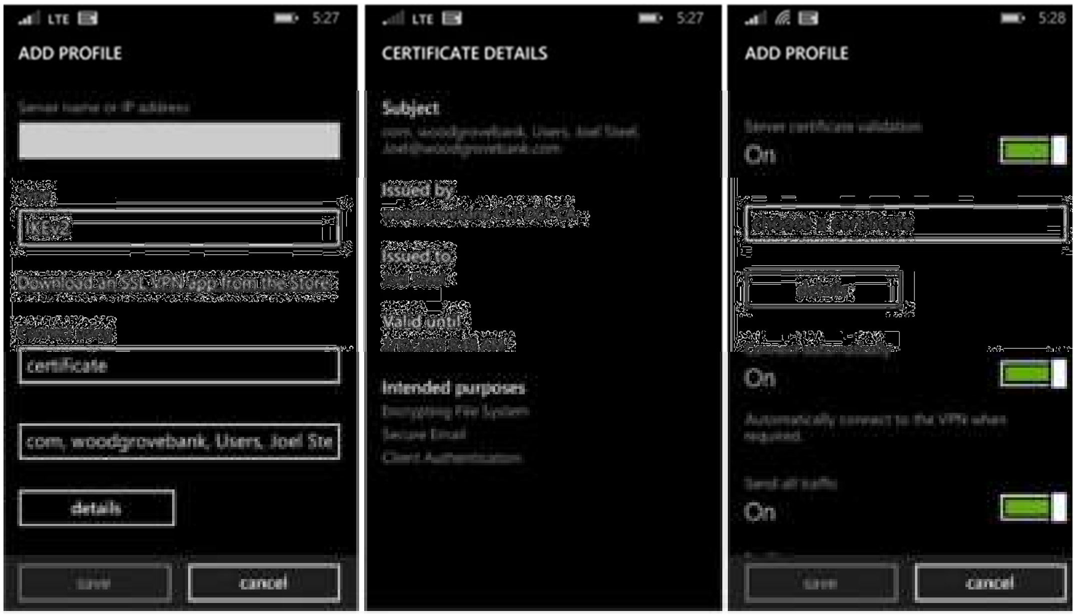
On

On

Contoso VPN

Figure 7.10: Setting User Name and Password

iv) For adding certificate security in your VPN, choose certificate. Select the desired certificate. Toggle the required settings. Provide the required details. To see the chosen certificate, select details (Figure 7.11).



ADD PROFILE

IKEv2

Download an SSL VPN app from the Store

certificate

com, woodgrovebank, Users, Joel Ste

CERTIFICATE DETAILS

Subject

Issued by

Issued to

Valid until

Intended purposes

ADD PROFILE

On

On

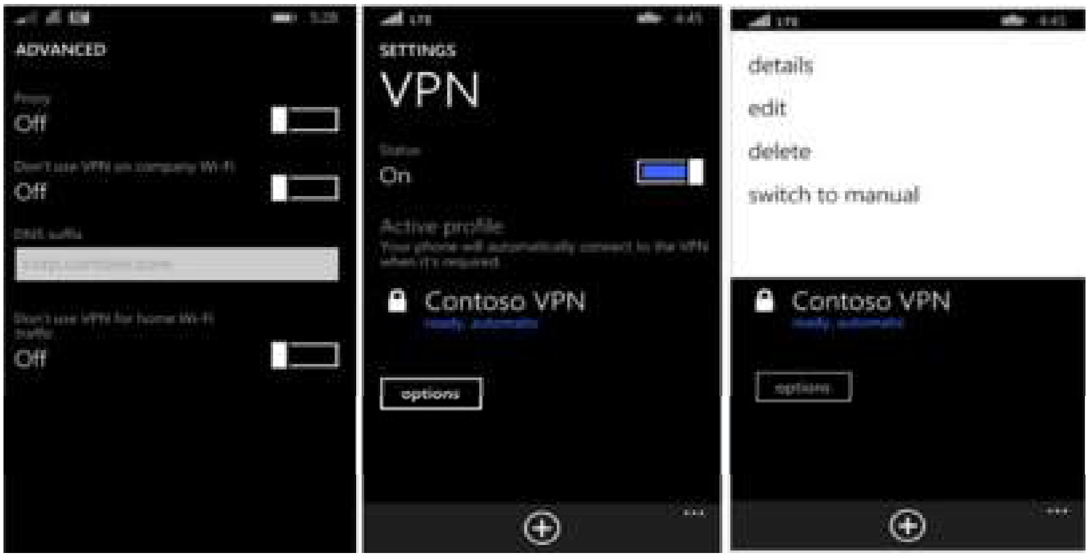
On

Figure 7.11 : Adding Certificate Security to VPN

v) While configuring certificate-based VPN, the certificate that is chosen under Connect is basically required for setting up the VPN connection.

vi) If you select server certificate validation, then option to select another certificate will be provided to you that will be used for VPN authentication process once the connection gets established.

vii) To provide more details, Choose Advanced, Choose Save. Now VPN profile will be get created automatically. To make further changes or see details on the profile, tap and hold the profile. To toggle to a manual profile, choose Switch to Manual (Figure 7.12).



ADVANCED

Off

Off

Off

SETTINGS

VPN

On

Active profile

Contoso VPN

Contoso VPN

Figure 7.12 : VPN Ready

7.4 RELEASES

Microsoft released different versions of Windows Phone. In this section, we shall discuss about released versions.

7.4.1 Windows Phone 7

On October 21, 2010, Microsoft released the first mobile client operating system as Windows Phone 7. Afterwards, few more updates have been done as shown in table 7.1.

Table 7.1: Versions of Windows Phone 7

Version	Changes
7.0.7004	It was the first version of windows phone 7 OS.
7.0.7008	Update process was improved for future update.
7.0.7390	New features were added viz. Copy and Paste, CDMA support, Fast application start-up etc.
7.0.7392	Issues of Fraudulent certificates were resolved.
7.0.7403	This version was required for updating to *MANGO.*

The comprehensive interface and Metro UI of the windows mobile operating system was highly appreciated for its design and fresh clean look. Now, let’s have a look on Market Share of Microsoft Windows Phone.

Windows Phone 7 (2010-2012)

Gartner (It is an American research and advisory firm providing information technology related insight for IT and other business leaders located across the world.) gives the report about devices running Microsoft OS sold to customers worldwide.

- In first quarter of 2011, 1.6 million devices were sold out.
- In second quarter of 2011, 1.7 million devices were sold out for a market share of 1.6%.
- In third quarter of 2011, Microsoft market share was slightly dropped to 1.5%.
- In fourth quarter of 2011, Microsoft market share was increased by 1.9% till first quarter of 2012.
- In second quarter of 2012, Microsoft market share got dropped back to 1.3%.

Windows Phone 7.5

A major update was announced by Steve Ballmer to Windows Phone 7 at the 2011 Mobile World Congress and this version was known by the name “Mango**”.** This version overcomes various shortcomings of the previous versions like adding up the newer version of web browser i.e. internet explorer 9 which provides the similar web standards and graphics as the desktop versions. In 2012,a minor update was released which was known by the name **“**Tango**”** that fixed the bugs of the previous versions, and also minimized the hardware requirements to let devices to run windows phone using256 MB of

RAM and800 MHz CPUs. Some of the comprehensive features were also disabled on these phones.

Windows Phone 7.8

Due to several changes in the kernel requirements and the hardware specifications, previously released Windows Phone hardware was incapable of being upgraded to Windows Phone 8. Although, it was announced that Windows Phone 7 devices would receive a free upgrade to Windows Phone 7.8.

The new features which were updated in Windows Phone 7.8:

- The start screen was updated in which users can resize live tiles
- New themes were added.
- An update to the lock screen was also introduced.
- New logo was introduced for applications such as Games, Store, and Office etc.
- Issues of volume control were resolved.

7.4.2 Windows Phone 8

Windows phone 8 was released in different phases. The phases were known as GDR (General Distribution Release) (Table 7.2).

Table 7.2 : Releases of Windows Phone 8

Release	Description
GDR 1	In General Distribution Release 1, **Portico** was released out as the minor update on December 2012. In this update, some improvements and bug fixes were done. Messaging techniques were enhanced. Bluetooth connectivity becomes more efficient and some new settings for Wi-Fi connections were also added.
GDR 2	Microsoft released a minor update as General Distribution Release 2 in July 2013.As we learned in the previous section that a partnership was signed between Nokia and Microsoft, with effect of this Nokia released its own update along with GDR 2. Nokia updated the firmware of the users, namely Lumia Amber, which was available for Lumia phones only. In GDR 2, some bugs were fixed and there were some improvements in camera.
GDR 3	General Distribution Release 3 was released in October 2013. Following features were updated in this release. • Changes were made in kernel, file system, drivers, network stack and graphic support. • Internet Explorer 10 was rolled out. • NFC (Near Field Communication) support was added. • Nokia Map technology was added. • Support of new protocols was added that allow people to sync Google contacts and calendar information with another devices.

	• 1080 Pixels screen resolution support. • Accessibility improvements for visually impaired. • Better Storage management. • SkyDrive was integrated which can sync user data such as music.

Windows Phone 8 (2012-2015)

Gartner gives report after the release of Windows Phone 8about devices running Microsoft OS sold to customers worldwide. In Q4 2012, Microsoft market share was increased to 3% which is 124% increase over the same time period in 2011.In Q3 2013, Windows Phone holds a worldwide market share of 3.6%, which is 123% from the same period in 2012.In Q2 2014, Windows Phone market share was dropped back to 2.5%.

Windows Phone 8.1

Windows phone 8.1 was also released in different phases. These phases were known as GDR (General Distribution Release) (Table 7.3).

Table 7.3 : Releases of Windows Phone 8.1

Release	Description
GDR 1	In this update, new language and region support was added for Cortana (Voice Assistant). An option of organising apps into folders was also enabled. Multiple messages in SMS could be sent simultaneously. GDR 1 also includes new VPN (Virtual Private Network) and Bluetooth features for enterprise users. Several changes were made to the Internet Explorer.
GDR 2	GDR 2 was released in February 2015 with increased security features for OEM’s (Original Equipment Manufacturer). Other changes made were: • Extra languages added. • Anti-Theft mode introduced. • Gesture keyboard was introduced. • New feature for unlocking the phone by Double-tapping the screen was introduced. • Dual SIM feature was introduced. • Support for Qualcomm snapdragon 200/400/400 LTE.

7.4.3 Windows 10 Mobile

Windows 10 Mobile is a mobile operating system developed by Microsoft. It is an iteration of the Windows Phone product line and a successor to Windows Phone 8.1 and is an edition of Windows 10, Microsoft's operating system for personal computers, as part of Microsoft's plans to unify Windows' application platform across multiple device classes.

Windows 10 Mobile aims to provide greater consistency with its counterpart for personal computers, including more extensive synchronization of content, a new universal application platform that allows one app to run on multiple Windows 10 devices such as PCs, mobile devices and Xbox, as well as the capability, on supported hardware, to connect devices to an external display and use a "PC-like" interface with mouse and keyboard input support. Microsoft has built tools for developers to easily port some iOS apps with minimal modifications. Windows Phone 8.1 smart phones are eligible for upgrade to Windows 10 Mobile, pursuant to manufacturer and carrier support. Some features may vary depending on hardware compatibility.

Windows 10 Mobile is designed for use on smart phones and phablets running on ARM processor architectures. Windows 10 Mobile entered public beta for selected Lumia brand smart phones on February 12, 2015. The first Lumia smart phones powered by Windows 10 Mobile were released on November 20, 2015 while eligible Windows Phone devices began receiving updates to Windows 10 Mobile on March 17, 2016, pursuant to manufacturer and carrier support.

In October 2017, it was revealed that Microsoft had discontinued active development of Windows 10 Mobile due to its low market share and the lack of third-party development for the platform, and that the operating system will only receive patches and maintenance releases going forward.

7.5 SUMMARY

In this era of technology, there are various brands and companies which are continuously emerging in this competitive world in terms of new techniques, Smartphone, PCs, laptops. Windows phone is one of them, it is being evolved from past 20 years and still it is upgrading itself. It released its first edition in 1996 as Windows CE and after that number of releases and versions took place and ultimately, Windows 10 Mobile is the latest version of Windows Phone. Numerous application developers are taking interest in Windows Phone. As of 2013, 21% of mobile developers are using Windows Phone platform, and many more stating that they are interested in adopting this platform. But, there are some another platforms also that are in high competition with Windows Phone. Windows Phone has some shortcomings in terms of applications, which it should overcome to compete with other platforms like Android and iOS. Five years ago, Windows mobile begun off life as a promising contrasting option to Android and iOS. Microsoft situated its scope of Windows phone 7 handsets as the genuine third portable biological system, yet it's a great opportunity to let it be known has fizzled. In the event that an absence of gadgets from phone creators and even Microsoft itself wasn't sufficient proof, the last nail in the pine box hit today. Microsoft just sold 4.5 million Lumia gadgets in the current quarter, contrasted with 10.5 million in the meantime a year ago. That is a monstrous 57 percent drop. Indeed, even a 57 percent expansion wouldn't be sufficient to spare Windows phone at this moment. Microsoft and Nokia have sold an aggregate of 110 million Windows phones contrasted with 4.5 billion iOS and Android phones in a similar period. IDC as of late detailed that 400 million phones were sold in the current quarter, which means only 1.1 percent of them were Lumia Windows’s phones. Microsoft does not have any convincing Lumia handsets, and the Lumia 950 and Lumia 950 XL were both baffling lead gadgets with incomplete Windows 10 Versatile programming. With Lumia deals on the decay and Microsoft's arrangement to not create a lot of handsets, it's reasonable we're seeing the finish of Windows phone. Gossipy titbits recommend Microsoft is building up a Surface mobile; however it needs to make it to the market first. Windows phone has for some time been in decay and its application circumstance is just deteriorating. With an absence of equipment, absence of offers, and fewer than 2 percent piece of the overall industry, it's an ideal opportunity to call it: Windows phone is dead. Genuine Windows on phones may turn into a thing with Continuum in the long run, however Windows mobile as we probably am aware it is finished.

Check Your Progress 1

1) List the features of Windows Mobile.

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

2) Define ARM architecture. Explain how this architecture proved as a boon for Windows Phone?

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

3) Describe the release of version 8 of Windows Phone.

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

4) What are hubs? Explain the importance of hubs.

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

5) What are the various changes made during the transition of Windows Mobile to Windows Phone?

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

……………………………………………………………………………….

7.6 FURTHER READINGS

- Migrating to Windows Phone by Jesse Liberty and Jeff Blankenburg Publisher: Paul ManningISBN-13(pbk): 978-1-4302-3816-4, ISBN-13(electronic): 978-1-4302-3817-1
- Dive in Windows Phone 8 Development by Jennifer Hawkins, Publisher: CreateSpace Independent Publishing Platform(2016), ISBN-1540658104, ISBN-9781540658104.
- https://en.wikipedia.org/wiki/Windows_Mobile
- https://en.wikipedia.org/wiki/Windows_Phone
- https://en.wikipedia.org/wiki/Windows_10_Mobile

ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Computer and Information Sciences

Commonwealth of Learning

INTRODUCTION TO MOBILE ARCHITECTURE

Mobile Hardware

3

School of Computer and Information Sciences
Indira Gandhi National Open University

Commonwealth *of* Learning

Introduction to Mobile Architecture

Block

3

MOBILE HARDWARE

INTRODUCTION TO MOBILE ARCHITECTURE

This course has been developed as part of the collaborative OER Course Development for ICT Skills initiative of the Commonwealth of Learning (COL). COL is an intergovernmental organisation created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

Indira Gandhi National Open University (IGNOU), established by an Act of Parliament of India in 1985, has continuously striven to build an inclusive knowledge society through inclusive education. It has tried to increase the Gross Enrollment Ratio (GER) by offering high-quality teaching through the Open and Distance Learning (ODL) mode.

Course Team

Author:

Mr. Sumit Srivastava, MCA
Technical Document Writer,
Indian Patent Office,
New Delhi,
India

Content Editor: Dr.P.Venkata Suresh, SOCIS, IGNOU, New Delhi, India

Content Design Team

Dr. P.V.Suresh,
Associate Professor & Director,
SOCIS, IGNOU, New Delhi

Mr. Amit Kumar Saxena,
Chief Technology Officer,
Grasp IO Inc, Bengaluru

Dr. S. Arumuga Perumal,
Associate Professor & Head of the Department, S.T. Hindu College,
Nagercoil, Tamil Nadu

Prof.Parma Nand Astya,
Dean, School of Computing Science & Engineering, Galgotias University,
Greater Noida

Mr. Shashi Bhushan Sharma,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. Akshay Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Dr. Naveen Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. M. P. Mishra,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Dr. Sudhansh Sharma,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Language Editors: Mr. P. Lakshmi Kantham , Faculty in English(Retd.), VSR & NVR College, Tenali, India

CRC Preparation: Tessa Media & Computers, New Delhi

Published by:

Indira Gandhi National Open University, Maidan Garhi, New Delhi-110068, India.

The course has been developed with the support of the Commonwealth of Learning, Canada.

Acknowledgements

Indira Gandhi National Open Universitywishes to thank those below for their contribution to development of this Course Material:

Staff of School of Computer and Information Sciences, Administration, Finance & Accounts Division and Electronic Media Production Centre, IGNOU, New Delhi, India and all External Experts. Special thanks to Dr. Sanjaya Mishra, Education Specialist(eLearning), Commonwealth of Learning, Canada for his support and guidance through out.

BLOCK INTRODUCTION

This is the third block of the Course. This block introduces the learner to hardware in mobile devices.

The capabilities of a mobile device significantly increase with addition of new hardware. Indeed, the hardware of a smart phone is significantly different from earlier versions which do not fall under the category of smart phones. For example, addition of touch screens, high resolution cameras, front camera, and bothie etc. were result of additional hardware and software.

The units in this block will focus on hardware in mobile devices, namely, processors, memory, sensors and I/O.

This block consists of four units and is organized as follows:

Unit 8 introduces mobile processors. The unit covers most categories of processors in mobile devices. It starts with Qualcomm's Snapdragon and explains different mobile processors including ARM and x86 processors.

Unit 9 deals with memory in mobile devices. Indeed, knowing about memory in a PC should not be mistaken as knowing about memory in a mobile device. The unit covers MROM, PROM, EPROM, EEPROM apart from volatile and non-volatile memories. Also, explained are Flash memories.

Unit 10 introduces sensors in mobile devices. It particularly explains about gyroscope, accelerometers, compass and proximity sensor.

Unit 11 discusses about I/O devices in mobiles. It covers display, camera, speakers and microphone which are part of almost every mobile device and are becoming increasing sophisticated with high resolution cameras, LEDs, stereo speakers etc.

UNIT 8 MOBILE PROCESSORS

Structure

8.0 INTRODUCTION

The word “processor” is used interchangeably with the term “CPU”. But, CPU is not the only processor in a computer. There are a lot of things in the computer that have the responsibility to process. In a personal computer, the processor is often called as microprocessor. This simply means that the processor’s elements are embedded in a single integrated circuitry(IC) chip. The four basic functions of a processor are-fetch, decode, execute, and write back.

Let us know about the major elements of a processor:

- **ALU**- It stands for Arithmetic Logic Unit. It is responsible for carrying out arithmetic and logic operations.
- **FPU**- It stands for Floating Point Unit, also known as numeric coprocessor. This can manipulate large numbers more quickly than the basic microprocessor.
- **Register**- Register is used for store data and instructions. It sends the operands to the ALU and stores the result of operations.
- **L1 and L2 Cache memory**- It is a kind of memory which stores the data that is frequently accessed. It saves time compared to having to get data from Random Access Memory (RAM).

Mobile processors are usually found in handheld devices like mobile computers and cell phones. A processor can execute anything whatever you want your Smartphone to do. Thesetypes of processors are generally accommodated in smaller chip package. These small chips usually use lower voltages as compared to the Desktop's embedded CPU chips. And, it also has more “sleep mode” capability. Let us now know something about the functioning of “sleep mode”. Mobile processors can be easily throttled down to different levels of power. And, when not in use, these processors can be turned off entirely.

When the load on the processor is low, the clock frequency may be slowed down, so that the power and battery life of the processor could be saved for further use.

8.1 OBJECTIVES

After going through this unit, you should be able to know about

- various mobile processors
- ARM processors
- x86 processor

8.2 MOBILE PROCESSORS

The following sections describe various mobile processors.

8.2.1 Qualcomm Snapdragon

A US based organization Qualcomm Technologies is occupied with creating distinctive innovations, for example, planning of semiconductors for cell phones, tracing gadgets, Virtual Reality, satellite phones, remote charging, and communications and so on. Qualcomm is presently celebrated for its Snapdragon image which is fundamentally responsible for releasing Long Term Evolution (LTE) modems and mobile processors. The majority of the mobile and tablet makers, utilizes Snapdragon processors in their items except Apple iPhone. These processors are referred to for its execution as it proficiently handles multitasking and can likewise effectively oversee high end processing which is very critical for gaming. An awesome favourable position of these processors is that they deliver less warmth when contrasted with different processors. In any case, they are costlier than some other processor. Snapdragon processor was first developed by Qualcomm and uses its own architecture, but, now Snapdragon processor uses ARM architecture which is becoming popular in smart phones. ARM processors have high multimedia capability and can easily decode HD videos of high resolutions. Snapdragon processors along with HD videos also cater to the GPS navigation system, 3D Video recording, and high performance graphics.

Table 8.1 lists the processor frequencies in the corresponding mobile devices which have Qualcomm Snapdragon processor.

Table 8.1: Mobile devices with Qualcomm Snapdragon processors

Mobile Devices	Processor
HTC	600-800 MHz
Samsung Galaxy Ace Plus, Nokia Lumia 610	800-1000 MHz
HTC Desire S, HTC Incredible S, Sony Xperia ARC	1 GHz
Blackberry Bold 9900, Blackberry Torch 9860	1.2 GHz
HTC Titan, Nokia Lumia 800, Samsung Galaxy S Plus	1.5 GHz
Samsung Galaxy S 2	1.2 GHz Dual Core

8.2.2 Samsung Exynos

It is a series of ARM-based system-on-chips which is manufactured by Samsung Electronics. Previously , Samsung developed the series of S3C, S5L, and S5P line of systems on chip (SoC). Its unique method helps in power savings. Samsung's Exynos processors have significantly 30-40% less power consumption than the competition. Some of the processors in the arrangement are Exynos 7 Dual, Exynos 7420, Exynos 7 Octa 7580, and Exynos 7 Octa 7870. Exynos 8 Octa 8890 is the most recent from Exynos. Exynos 8 Octa 8890 is utilized as a part of Samsung Galaxy S6 and S6 edge. Samsung haven't made any declarations about any new model after Exynos 8 Octa 8890. Snapdragon processors are currently utilized by Samsung smart phones and tablets.

Table 8.2 lists the processor frequencies in the corresponding mobile devices which have Samsung Exynos processor.

Table 8.2: Mobile Devices with Samsung Exynos processors

Mobile Devices	Processor
Samsung Galaxy S line, Samsung GT-S8500 Wave, Samsung Wave II S8530, Nexus S, Meizu M9, Samsung Galaxy Tab, Samsung Droid Charge, Exhibit 4G, Samsung Infuse	1 GHz
Samsung Galaxy S II, Samsung Galaxy Note, Samsung Galaxy Tab 7.7, Samsung Galaxy Tab 7.0 Plus, Hardkernel ODROID-A, Meizu MX, Cotton Candy by FXI Tech	1.2 GHz Dual Core

8.2.3 NVIDIA Tegra

- This is an US based innovation organization which manufactures processing units for gaming, graphics and cell phones. It additionally creates chips for smart phones, tablets and cell phones under the name of brand Tegra. Tegra processors utilized as a part of smart phone and tablets which are mainly Tegra 4/ 4i / Tegra K1.

- Tegra 4 is utilized by HP Slate book, Xiaomi Mi3, Asus Transformer PAD,XOLO Play Tegra Note and Toshiba Excite.
- Tegra 4i is used by Blackphone.
- Tegra K1 is utilized by Google Nexus 9, Acer Chromebook 13and 14, Navi Big Tab, and NVIDIA Shield Tablet.

When it comes to mobile gaming, NVIDIA Tegra is considered best. The processing of Graphic is best in contrast with any other mobile processors. Be that as it may, there are couple of disadvantages of Tegra, it reduces the battery life with heat generation. Tegra processors are bit costlier than different processors.

NVIDIA processors were known for graphics for the desktop. They entered into smart phones and mobile devices very late and their first product was Zune music player in 2008. It is the first to bring a quad core processor in market with the HTC One X. The main feature of NVIDIA Tegra is its graphic processing capability. Table 8.3 lists the processor frequencies of the corresponding mobile devices which have NVIDIA Tegra processor.

Table 8.3: Mobile Devices with NVIDIA Tegra processors

Mobile Devices	**CPU**
Motorola Atrix 4G, Motorola Droid X2, Samsung Galaxy R, LG Optimus 2X, Micromax Superfone A85, LG Optimus Pad, Notion Ink Adam tablet,, Motorola Xoom,Samsung Galaxy Tab 10.1, Lenovo ThinkPad Tablet, Sony Tablet S	1 GHz dual core
Samsung Galaxy S II, Samsung Galaxy Note, Samsung Galaxy Tab 7.7, Samsung Galaxy Tab 7.0 Plus, Hardkernel ODROID-A, Meizu MX, Cotton Candy by FXI Tech	1.4 GHz quad core

8.2.4 More Mobile Processors

The following are some more mobile processors:

- **Apple Mobile Processors-**Apple does not make any microprocessors. Rather, they make contracts with microprocessor making organizations essentially Taiwan Semiconductor Manufacturing Company (TSMC)and Samsung for making custom assembled processors that suits their outline and performance expectations.
- **Intel Atom and core M processors-**Intel is an American multinational organization which manages PC and microchips. Atom is the name allotted to the less power consuming and minimal cost 32/64 bit chips designed for utilizing as a part of smart phones and tablets. **Intel Core M** microprocessors were introduced by Intel which are ultra-low voltage intended for ultra-thin notebooks, smart phones and two in one convertibles. The microprocessor expends less power and along these lines

has long battery life. With the arrival of Intel Core M processors, 40% lift in CPU and graphics performance are also offered. This version is more productive than previous versions. All manufacturers supported Intel Atom processors. Manufacturers like Microsoft, HP, Dell, Asus, Lenovo, uses the latest Atom Processor. The one and only smart phone which uses Atom processor is Asus Zenfone series.

- **MediaTek-**Taiwanese semiconductor organization MediaTek, is providing chips to cell phones, HDTVs and other electronic devices.MediaTek microprocessors are based on 64 bit ARM design.The most recent MediaTek processor is capable to manage up to 3 GHz speed and comes in variety of cores, for example, Dual Core (Two core), Quad Core (Four core), Hexa Core (Six core) and Deca Core (Ten core).The most recent processor from MediaTek are Helio X20 and X25 which are utilized as a part of smart phones and tablets. Mostly Chinese manufacturers deal with MediaTek processors. MediaTek processors are used by Xiaomi, Meizu, LeEco Le, Yu etc on Smart phones. There are some more manufacturers like Acer, Asus, Lenovo, Amazon Fire HD, and QMobile who use this processor on their tablets. There are numerous advantages of MediaTek processor. MediaTekPowerVR GPU is superior to different processors. It creates less heat when contrasted with different processors. These processors are less expensive and are offered with great execution.

- **HiSilicon-**It is a Chinese company and specializes in manufacturing semiconductors. Huaweiis the owner of this company. Chips produced by this company are based on the ARM architecture which we will study in detail in the next section. There are various processors released by HiSilicon are K3V1, K3V2, K3V2E, Kirin 620, Kirin 650, Kirin 910, Kirin 920, Kirin 930, Kirin 950 and destined to be released Kirin 960.

8.3 ARM PROCESSORS

ARM stands for Advanced RISC Machine, and was previously known as Acorn RISC Machine. It is combination of RISC (Reduced Instruction Set Computing) architectures for computer processors. The processors based on RISC architecture require fewer transistors as compared to personal computers having complex instruction set computing x86 processors. Processors based on RISC reduces cost, heat and power use. These are the reasons because of which RISC architecture is supported in light, portable, battery-powered devices including tablets, smart phones, and laptops. The basis of every ARM processor is formed by ARM architecture. When, various components are included ARM engineering so that the developing interest for new usefulness, incorporated security highlights, elite and the necessities of new and developing markets are satisfied. The ARM design is like a Reduced Instruction Set Computer (RISC) engineering, as it joins these common RISC engineering highlights:

- A uniform register file load/store engineering, where information preparing works just on register contents, not straightforwardly on memory contents.

- Basic addressing modes, with all load/store addresses decided from register contents and direction fields as it were.

Improvements to an essential RISC engineering empower ARM processors to accomplish a decent adjust of superior, little code measure, low power utilization and little silicon region. An ARM processor is developed by Advanced RISC Machines (ARM). It is based on the RISC architecture. ARM makes RISC multi core processors of 32-bit and 64-bit. RISC processors are fundamentally intended to perform a smaller number of types of computer instructions with the goal that they can keep running at a higher speed, performing millions of instructions in a second (MIPS).RISC processors evacuate repetitive guidelines with the goal that it can without much of a stretch upgrade its execution on account of CISC (Complex Instruction Set Computing) gadgets. Gadgets, for example, smart phones, tablets, and interactive media players utilize ARM processors.ARM requires fewer transistors because it is based on the reduced instruction set approach and therefore ARM processors use the Integrated Circuitry (IC) chip which is relatively small when compared to the transistors. ARM is compatible with miniaturized devices because of its salient features like smaller size, reduced complexity, and lower power consumption.

8.3.1 Features of ARM Processor

The following are features of a ARM processor:

- The architecture of ARM processor is kind of Load/Store.
- It is designed with a view to enhance power saving.
- Single cycle execution is implemented in this processor.
- It supports hardware virtualization.
- In need of high performance, it can execute 64-bit and 32-bit states.

Developers think that it’s simple to code for ARM processors due to its exceptional features like more proficiency and multi core processing. At the point when the developers deal with x86 processors, they don't have an indistinguishable throughput as of ARM processors, in any case, once in a while ARM processors surpass the execution of x86 processors for applications that exist on both architectures.

8.3.2 ARM Architecture

The ARM design was initially created by ACORN Computers in 1980s. 32-bit Reduced Instruction Set Computer (RISC) stack, store design is utilized as a part of ARM machines. The manufacturers rely on ARM machines because of its relative simplicity for low power applications like mobile, embedded and microcontroller applications and small microprocessors. This architecture uses registers to manipulate memory as it is not possible to directly manipulate the memory. The main focus of the instruction set of this architecture is on reducing the number of cycles per instruction featuring mostly single cycle executions. Figure 8.1 depicts ARM architecture.

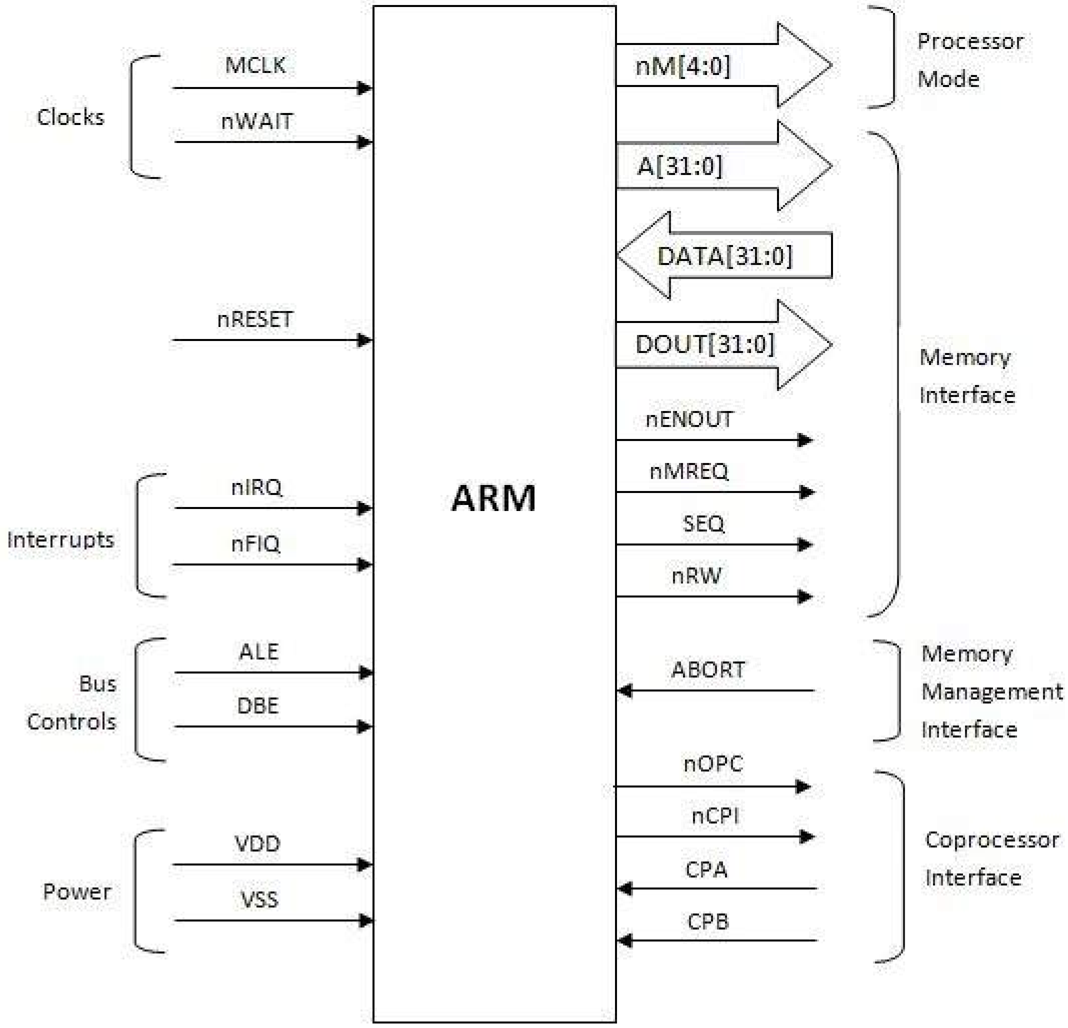


Figure 8.1: ARM Processor Architecture

8.4 x86 PROCESSORS

x86 is the mix of backward-compatible instruction set architectures in view of the Intel 8086 CPU and Intel 8088. Intel 8086 was first presented in 1978 , it was 16-bit based microchip and was the expansion of Intel's 8-bit based 8080 chip. In Intel 8086, more memory can be tended to with the assistance of memory divisionx86 architecture has been executed in processors of AMD, Intel, Cyrix, VIA and numerous different organizations. Till now the laptops and computers sold in the emerging markets are majorly based on x86 architecture. Intel 8086 was primarily developed for single user computers, small multi user or embedded systems but gradually it grew in features and processing power. Today, x86 is also used in workstations, servers, mid range computers or super computer clusters. The x86 architecture depends on CISC (Complex Instruction Set Computing) plan with primary concentrate on backward compatibility. The instruction set of this design is the augmented form of the 8-bit 8008 and 8080 architectures. Byte addressing is empowered in this design and words are put away in memory with little little-endian byte arrange. For all substantial word sizes, memory access to unaligned addressed is permitted. The biggest size for memory locations is 16, 32 or 64 bits.

8.4.1 Basic Design of x86 Processor

The following is basic information related to x86 processor:

- A place where all the calculations and logic operations are being done, is known as CPU (Central Processing Unit). There are numerous things that are stored in CPU like storage location, registers, high frequency clock, a control unit, and arithmetic logic unit.
- The synchronization of the internal operations of the CPU and other components in the system is a responsibility of the clock.
- Control Unit (CU) is responsible for arranging the sequence of steps involved in executing system instructions.
- The arithmetic operations such as addition and subtraction and logical operations such as AND, OR, and NOT are done by ALU (Arithmetic Logical Unit).
- The CPU and the rest of the computer is connected to each other via pins that are attached to the CPU socket in the computer's motherboard. Most of the pins on the CPU are connected to Data Bus, control bus, and the address bus. The instructions and data are temporarily stored in the memory storage unit while a computer program is running. The CPU sends a request for data to the storage unit, then it transfers all data stored from Random Access Memory (RAM) to the CPU, and finally transfers data from the CPU to memory. In CPU all the data processing takes place, so before execution of any command, programs that exist in memory must be copied into the CPU.

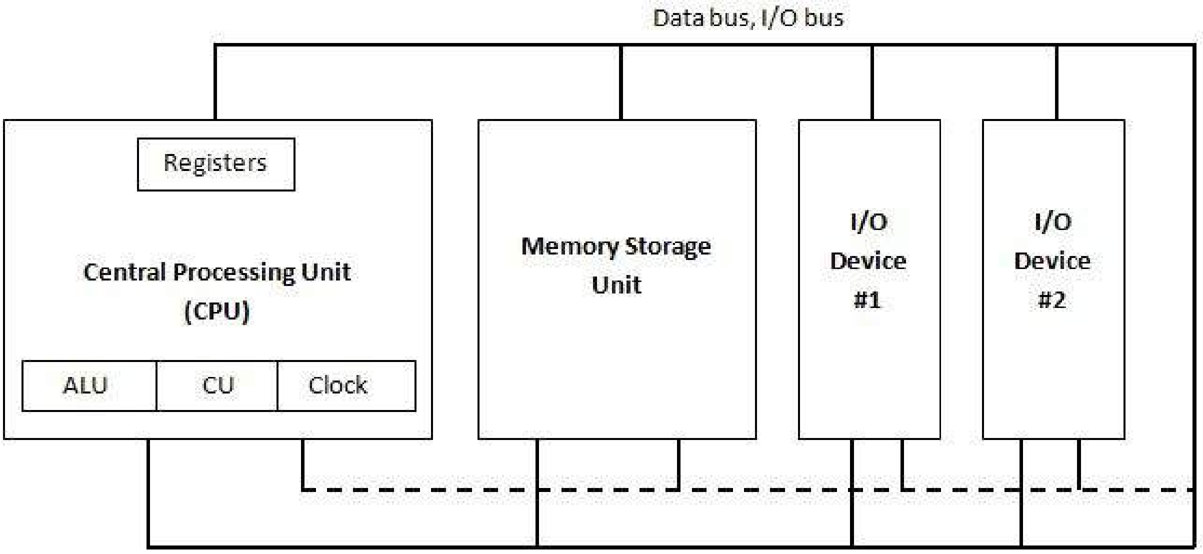


Figure 8.2: Block Diagram of Microcomputer

- A Bus is a combination of parallel wires that is responsible for transferring data from one part of the computer system to another. There are four types of buses in a computer system i.e. DATA, Input & Output (IO), Control, and Address. The **Data Bus** is responsible for transferring instructions and data between the CPU and memory. The **IO Bus** is responsible for transferring data between the CPU and the devices that have input/output command. The **Control Bus** is used to synchronise actions of any device that is connected to the system Bus. The addresses of instructions and data are stored in the **Address Bus** which transfers data between the CPU and memory.

Figure 8.2 shows block diagram of a Microcomputer.

8.4.2 Instruction Execution Cycle

Instruction execution cycle is a sequence of individual operations that are separated from the execution of a single machine instruction. Before any execution, a program is loaded into memory. There is an instruction pointer that contains the address of the next instruction to be executed.

Fetch The Control Unit fetches the next execution instruction from the instruction set and increment the Instruction Pointer (IP) which is also known as Program Counter (PC).

Decode Control Unit decode the function of the instruction to determine what the instruction will do. The input operands of the instruction are passed to the ALU, and signals are sent to the ALU indicating the operation to be performed.

Execute At last, the ALU uses the registers as operands and executes the instructions and then forward the output to registers/memory.

8.4.3 Differences between ARM and x86 Processors

The following are differences between ARM and x86 processor:

The first and foremost difference between ARM and x86 processors is that ARM processors are based on RISC (Reduced Instruction Set Computer) architecture, whereas x86 processors are based on CISC architecture. In ARM processors, instructions are executed in one clock cycle whereas x86 processors' instructions are mostly complex and take multiple CPU cycles to execute each instruction. Explicit load and store model is followed in ARM processors whereas in x86 processors, the load and store register is in built. Therefore, CISC processors have more hardware logic to decode and execute the complex instructions.

Check Your Progress 1

1) ----------- developed Snapdragon processor.

2) RISC stands for ----------.

3) ------------,------------ and Execute form part of Instruction Execution Cycle.

8.5 SUMMARY

A processor executes what you need your smart phone to do. Early cell phones were basically cousins of conventional landline phones. Smart phones, in any case, are convenient PCs that happen to have phone capacities implicit. Underneath that astounding touch screen show is an undeniable PC, in charge of advising your applications how to work, your GPS how to get you home and you who to approach on the phone. The processor is the brains of the operation.

It's a component found in the fundamental processor that peruses and executes instructions. Gadgets started with a single-core processor, however builds made all the more effective gadgets by incorporating more cores in one gadget. That prompted dual-core gadgets. Before long there were quad-core processors (as in four cores), and now there are hexa (six) and octa-core (eight) smart phones and tablets.

The more cores, the quicker they can divvy up the work you're requesting that the phone to do. That implies various cores make your experience smart; Apps load rapidly. You can capture high-quality photographs or HD video and after that peruse through your phone gallery immediately. Animations and recordings play easily without stammering. Games don't get stalled. There are special cases, yet for the most part, the more cores the better the execution.

No. An octa-core processor is speedier than a quad-core processor just when it's running an application that exploits its capacities or when you're multitasking. For instance, one core might be grinding away on your web browsing while another is standby. A call comes in, and the second core gets the chance to work. Both your web browsing and your phone call proceed easily, however not really twice as quick.

There are numerous—the processor is just a single segment of your brilliant gadget (consider it like all the parts of your auto's motor). There's likewise a graphics handling unit, RAM for short-term memory, antennas for Wi-Fi and GPS. These cooperate as a unit and all should be considered when you purchase a smart phone or tablet.

When looking for a smart phone or tablet, the processor specs are helpful to get a feeling of what the gadget can do. In case you're searching for speed, pull out all the stops with a multi-core processor.

In this era of technology, most of the population is engaged in the usage of smart phones. The sales of smart phones is rapidly increasing. There are numerous companies who are engaged in manufacturing and selling smart phones like Microsoft, Apple, Samsung, Asus, etc. These companies release upgraded versions of their products after time to time. Mostly these upgrades are in processors of mobiles, tablets, Laptops, etc. Processors in smart phones are manufactured by different companies and under the name of different brands. Some of them are:

- Qualcomm-Snapdragon
- Samsung Exynos
- Texas Instruments
- NVIDIA-Tegra

8.6 SOLUTIONS/ANSWERS

1) Qualcomm

2) Fetch, Decode

3) Reduced Instruction Set Computing

8.7 FURTHER READINGS

- Smart Phone and Next Generation Mobile Computing by Pei Zheng and Lionel Ni;Morgan Kaufmann; ISBN-13: 978-0-12-088560-2, ISBN- 10: 0-12-088560-3
- Handbook of Research in Mobile Business : Technical, Methodical and Social Perspectives by Bhuvan Unhelkar;Information Science Reference; ISBN-10: 1605661562, ISBN-13: 978-1605661568
- https://en.wikipedia.org/wiki/Mobile_processor
- https://www.qualcomm.com/products/mobile-processors
- https://en.wikipedia.org/wiki/Exynos
- http://www.samsung.com/semiconductor/minisite/exynos/
- https://en.wikipedia.org/wiki/Apple-designed_processors
- http://www.NVIDIA.com/object/tegra-4-processor.html
- https://en.wikipedia.org/wiki/HiSilicon
- https://en.wikipedia.org/wiki/MediaTek
- https://www.intel.in/content/www/in/en/processors/processor-numbers.html
- https://www.intel.com/content/www/us/en/products/processors/atom.html
- https://www.intel.com/content/www/us/en/products/processors/core/m3-processors.html

UNIT 9 MEMORY

Structure

9.0 INTRODUCTION

Memory resembles a human cerebrum. We can store information and instructions in it. When we discuss memory as far as PC, a storage room is accessible in a PC where information to be processed and instructions required for processing are put away. The memory is partitioned into little parts called cells. Every cell has a special address; the address can store the value from 0 to memory measure minus 1.

For example, if PC has 64K words, then the memory has 64*1024=65536 memory areas; then the address may lie in the range of 0 and 65535.

There are three types of memory available:

- Primary Memory
- Secondary Memory
- Cache Memory

Primary Memory

It is also known as main memory. It contains the current data and instructions on which computer is working. In primary memory, data is lost whenever power supply breaks down. All the information i.e. data and instructions required to be processed resides in this memory. This memory is made up of semiconductor. There are two types of memories which come under the category of Primary Memory i.e. RAM and ROM.

RAM stands for random access memory, is one of the basic segments of the smart phone alongside the processing cores and dedicated graphics. Without RAM in any kind of electronic gadget like this your smart phone would neglect to perform basic operations in light of the fact that getting to documents would be surprisingly slow. This kind of memory is a center man between the file-system which is put away on the ROM, and the processing cores, serving any kind of data as fast as could reasonably be expected. Basic files that are required by the processor are temporarily saved in the RAM, expecting to be processed. These files could be things, for example, OS components, application information and game designs; or usually anything that requires to be fetched at quicker than other storage can provide.RAM utilized part of smart phones is in fact DRAM, where D stands for Dynamic. The structure of DRAM is with the end goal that every capacitor on the RAM board stores a bit, and the capacitors leak charge and require consistent "refreshing"; thus the "dynamic" way of the RAM. It likewise implies that the content of the DRAM module can be changed rapidly and effectively to store diverse files. The benefit of the RAM not being static is that the capacity can change to adapt to whatever assignments the system is attempting to perform. On the off chance that a whole operating system was, say, 2 GB on disk, it wouldn't make sense or be effective for the RAM to archive the whole thing, particularly when smart phones with lower space of RAM (like 512 MB) can't bear.RAM is diverse to the blaze style ROM storage on the gadget in that at whatever point power is separated from the RAM module, the contents are lost. This is known as volatile memory, and it mostly helps the access time of the RAM to be so quick. It likewise clarifies loading screens: data from the slower ROM must be passed to the speedier RAM, and the restricting variable much of the time is the perused speed of the ROM. At the point when the system is fueled off, the content of the RAM is lost thus at the following boot, the RAM should be filled yet again from the content on the slower storage. Figure 9.1 depicts RAM.

Figure 9.1: RAM

Secondary Memory

It is also known as non-volatile or External Memory. It permanently stores data and instructions. It is slower than Primary Memory. CPU access these memories via input output devices as it cannot access these memories directly. Firstly, contents of secondary memory are transferred to main memory, after that the CPU can access it. For example: CD-ROM, DVD, Disk etc. These memories are magnetic and optical. It is also known as backup memory. Secondary memory is non volatile in nature. Figure 9.2 depicts CD-ROM.

Figure 9.2 : CD-ROM

Cache Memory

It is made up of semiconductor device. It is used to increase the speed of CPU. This memory acts like a buffer between main memory and the CPU. Most frequently used data and program by CPU resides in this memory. Firstly, the information and program is exchanged from disk to cache memory, from where CPU can get to it.

Merits of Cache memory

- It stores data for temporary use.
- It is faster than primary memory.
- It generally keeps those instructions that can be executed inside a brief timeframe.
- As compared to main memory, access time of cache memory is less.

Demerits of Cache memory

- It has restricted limit.
- It is extremely costly.

9.1 OBJECTIVES

After going through this unit, you should be able to know about

- ROM (Read Only Memory);
- Flash memory; and
- Other types of memories in a mobile device.

9.2 MEMORY IN A MOBILE PHONE

In this era of technology, Smartphone is a device which is used by every second person of the population. There are a huge number of brands available of Smartphone. Every Smartphone have some memory associated with it. Basically, Smartphone uses two major types of memories i.e. volatile memory and non-volatile memory. In every Mobile phone, there are three memories, namely, phone memory, memory on SD card and RAM. SD card memory is optional. The sizes of memories can be found by tapping on *settings* and other options that follow.

Now, let's learn more about volatile and non-volatile memories.

9.2.1 Volatile Memory

Volatile memory will not hold its content after the power supply is off. It loses the data it holds after the device is powered off. Volatile memory is like RAM(Random Access Memory). In RAM, each memory location has its own unique address that can be easily read or written independently. Advantage of using this memory is that the access time is very less. This memory acts like the run-time memory of software applications(device's operating system).RAM is also used as a storage drive in which part of it is reserved/allocated, this is known as RAM disk. RAM disk is available in smartphones as D-drive. Because, this is a volatile memory, data is lost when the device is switched off, therefore only small and temporary data should be stored in this.

9.2.2 Non-Volatile Memory

Unlike Volatile memory, non-volatile memory doesn't lose its data even if the power/device is switched off. Non-volatile memory always retains its state.

The following are basically two types of non-volatile memories-

- **ROM :** It stands for Read-only Memory. Usually, it is based on flash memory.
- **Flash RAM :** It is a kind of ROM, however it stores the data related to the Operating System and other applications. It stores that data which is not to be erased.

ROM can be seen as Z-drive on smart phones. This drive can only be view/read but it is not permissible to write. In new devices, this drive could not be accessed by unprivileged third party applications. Another type of non-volatile memory is Flash-RAM. It is based on the flash memory technology and it has write permission also along with read and view. This memory is also known as "user memory" or "phone memory". This device stores system software i.e. operating system like C-drive in computer. While initializing, the C drive needs that files/data which is used by the operating system for various purposes. This is the memory where various by default things are stored such as contacts, messages or photos.

9.3 MEMORY CARD

As depicts by its name, it is another kind of memory. Nowadays, almost every smart phone comes with the slot given for memory card to increase its memory space. Memory cards are available in different memory size like 8GB, 16GB, and 32GB etc.

Different types of memory card are:

- **MMC:** It stands for Multi Media Card. MMC is used in portable devices from which it can easily be removed. For example, it is used in digital camera for storing image files. A user can copy the pictures of digital camera to his/her computer with the help of MMC reader. MMCs are available up to 512GB in size. Figure 9.3 depicts MMC card.

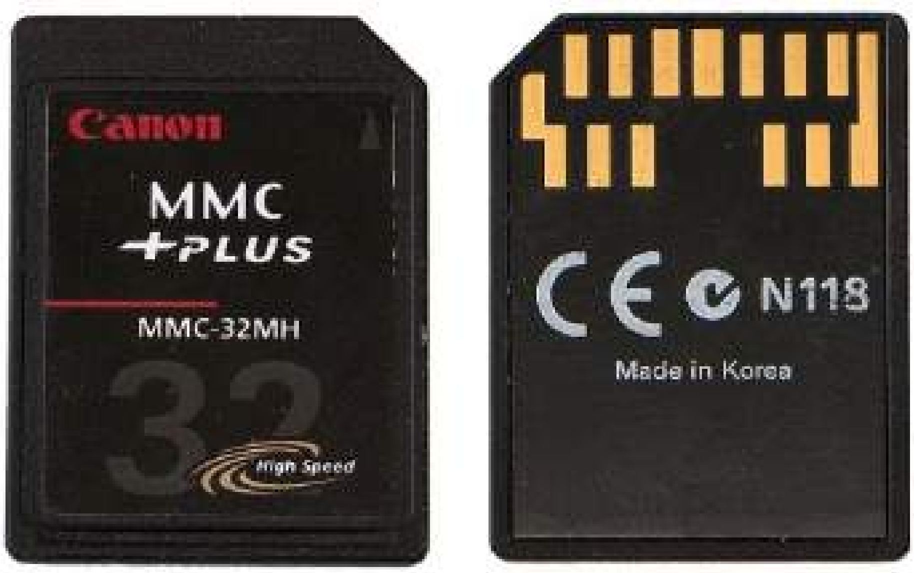


Figure 9.3 : MMC

- **SD Card:** It stands for Secure Digital Card. SD cards are slightly thicker and have some extra connectors as compared to MMCs. They also support extra security features and can also be used for peripherals.
- **RSMMC:** It stands for Reduced Size Multi Media Card. It is almost half of the MMC in size. These are also known as high voltage RSMMC cards because they can be operated at a voltage range of approximately 3.3 volt. Figure 9.4 depicts RSMMC.

Figure (D): RSMMC

- **Micro SD:** It is smaller in size as compared to SD cards and consumes different power than SD cards.

In every smart phone, by tapping on settings and other appropriate options, storage on the device can be known.

9.4 ROM

ROM is non-volatile memory which is used in computers, mobiles and other electronic devices. The data which is stored in ROM, cannot be modified easily, therefore it is mainly used for storing firmware or application software's. The instructions that are required to start a computer are stored in ROM.

The following are different types of ROM:

9.4.1 MROM

It stands for Mask ROM. These sorts of ROMs were first hard wired. They contain preprogrammed set of information or instructions to be processed.

9.4.2 PROM

It stands for Programmable Read Only Memory. It can be modified only once. Contents can be written on it only one time using PROM program and then it cannot be modified or erasable. There are small fuses in this ROM which are burnt open during programming. Figure 9.4 depicts PROM

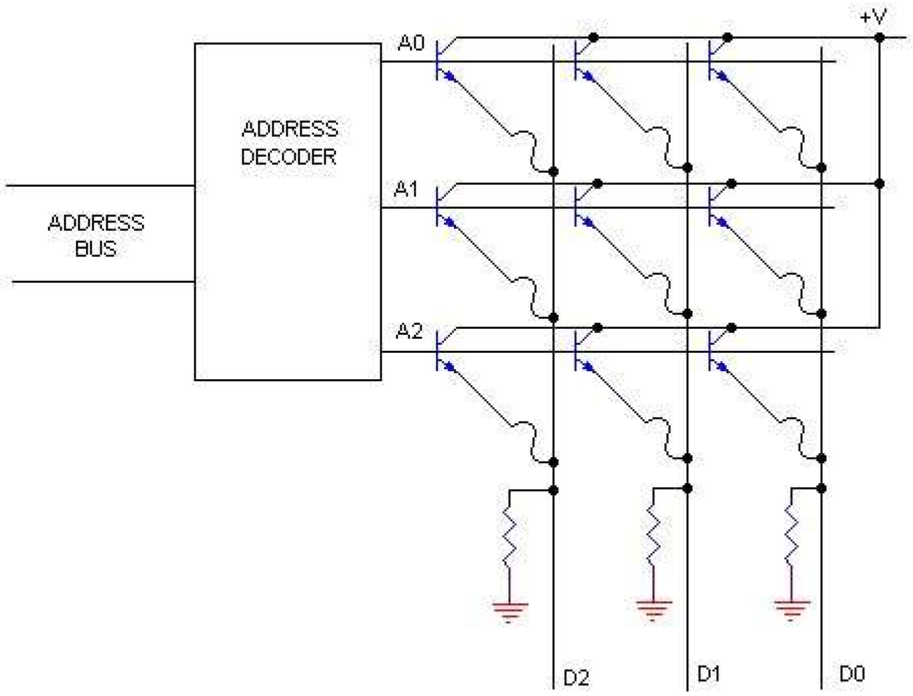


Figure 9.4 : PROM

9.4.3 EPROM

It stands for Erasable and Programmable Read Only Memory. The content of EPROM can be wiped off by presenting it to ultra-violet light for 40 minutes. During programming, in insulated gate region an electric charge is trapped which can be retained for more than 10 years because it has no leakage path. This charge can be erased by passing the ultra violet light through quartz crystal window. Figure 9.5 depicts EPROM.

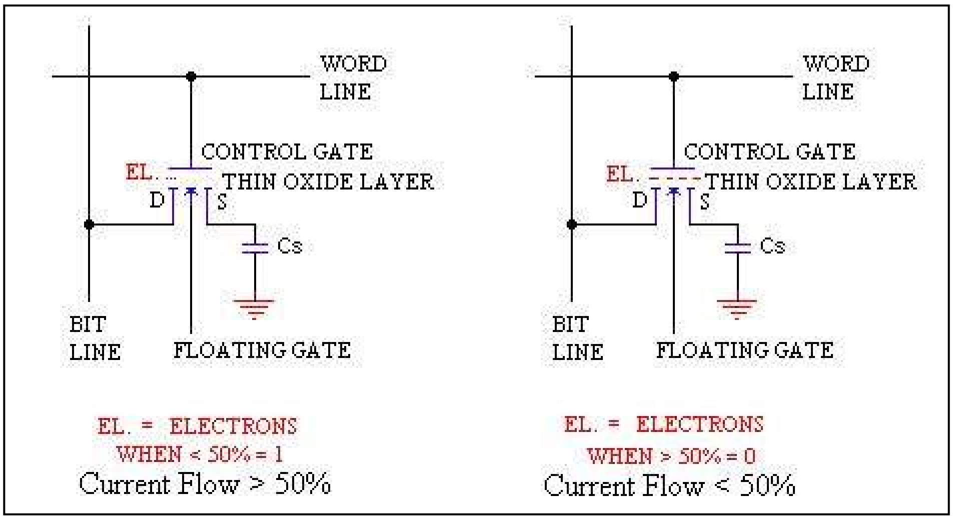


Figure 9.5 : EPROM

9.4.4 EEPROM

It stands for Electrically Erasable and Programmable Read Only Memory. On this ROM, erasing and programming can be done electrically. User can erase and re-program this memory about 10 thousand times. Approximately the whole process of erasing and programming take about 4-10 milliseconds. We can easily erase and program a selected location in EEPROM. It doesn't wipe off the whole chip at once, notwithstanding it delete one byte at any given moment. Subsequently, re-writing computer programs is adaptable in EEPROMs.

The following are advantages of ROM:

- It is non-volatile in nature.
- It cannot be changed accidently.
- They are easy to test
- They are cheaper and more reliable than RAMs.

9.5 FLASH MEMORY

Flash Memory is another kind of storage medium that can easily be erased and programmed. This memory is non-volatile in nature. It does not lose the data even when there is breakdown in power supply. It was first developed by Toshiba in 1980, but it was released in the market in 1984 and it was named as EEPROM (Electrically Erasable Programmable Read-only Memory). There are basically two types of flash memories which are based on the concept of NAND and NOR logic gates. The characteristics of both the memories reflect the internal characteristics of respective gates. It is used in devices like USB flash drives, digital cameras and video games. In NAND-type flash memories, data can be written and read in blocks, and the size of these blocks are much smaller than the size of entire device. Whereas, NOR-type flash memories can independently read or write single machine word (byte). Examples of both types of flash memories are video games, digital audio players, digital cameras, mobile phones, scientific instrumentation, industrial robotics etc. Along with its non-volatile nature, it also offers fast access time; however, it is not as fast as RAM or ROM. Also, there is a limitation on write cycles in flash memory.

User can read data n number of times from flash memory but it will stop working once the limitation of ‘write’ operation is exceeded. Approximately 1, 00,000 to 10, 00,000 write operations can be done on flash memory.

Flash memory stores data and instructions in an array like structure of memory cells, these cells are made up of floating-gate transistors. There are various types of devices that are made up of these cells, for example, single-level cell (SLC) devices are made up of single cell and each cell stores one bit of information, whereas, multi-level cell (MLC) devices are made up of multiple cells and they can store more than one bit per cell.

In flash memory, which is made up of one transistor, each memory cell shows the characteristics of a standard MOSFET (metal-oxide-semiconductor field-effect transistor). On the top of the memory cell, there is a control gate and below the control gate, there is a floating gate which is insulated by an oxide layer. The floating gate is placed between the Control gate and the MOSFET channel.

Low-level access of flash memory is quite different from other types of storage devices such as DRAM, ROM, and EEPROM. Flash memory supports alteration of bits i.e. 0 to 1 and 1 to 0 and it also supports random access through external buses. For reading and programming, NOR memory has an external address bus, and NAND memory do it page-wise.

9.5.1 NOR Memories

NOR flash and RAM follow the same process of reading. For proper execution of the process of reading, the data and address bus should be mapped correctly. User can directly execute the programs stored in NOR flash without copying the data into the RAM first. Just like reading, NOR flash can be programmed in a random-accessed manner. In programming, bits change from 1 to 0, and the bits that are already zero remain unchanged.

9.5.2 NAND Memories

NAND flash architecture was first introduced by Toshiba in 1989. NAND flash is generally used as block devices, such as hard disks. Each block contains number of pages. The size of each page is about 512, 2048, or 4096 bytes. With every page, few bytes are attached for the storage of error correcting code (ECC) checksum. NAND devices use ECC for compensating for the bits that may spoil/fail during normal device operation. Table 9.1 gives differences between NOR and NAND flash memories.

Table 9.1 : Differences between NOR and NAND flash memories

NOR Flash	NAND Flash
It allows Random-access for reading	It allows page access only
In this, cells are connected in parallel to the bit lines, so that cells can be read and programmed individually	In this, cells are connected in series, similar to NAND gate
It consumes more space as compared to NAND flash	It consumes less space as compared to NOR flash
Cost is high	Cost is low
The size of each cell is comparatively larger than NAND flash.	The size of each cell is smaller than the corresponding cells of NOR flash memory

Check Your Progress

1) __________________________ memory loses data if the device is powered off.

2) MMC stands for ______________________________.

3) In EEPROM, _______________________ and _________________ can be done electronically.

4) NOR flash memory allows ____________________________ access for reading.

9.6 SUMMARY

While considering the memory particulars of any smart phone, it is vital to comprehend the distinction amongst internal and external (or expandable) memory. Internal memory is the pre-installed memory space provided along-with the phone, generally 16, 32 or 64GB, where the working framework, pre-introduced applications and other system applications are installed. The space of internal memory can't be expanded or diminished by the user, so if your phone has just 16GB of internal capacity and not expandable further, this is all the storage room you will ever have numerous gadgets which include a Micro SD card space are sold with a card as of now embedded. Be that as it may, not all phones will have this additional storage space included, and not all phones even have the feature to include external memory. iPhone, for instance, has never given users the capacity to include more storage space by utilizing a SD card, neither have both of the LG Nexus gadgets. In the event that storage, for music, pictures, or other user files, is essential to you, the capacity to include another 32 or even 64GB sensibly economically ought to be an imperative thought.

Cloud Space

To conquer the issue of lessened internal storage space, a few top of the line smart phones are sold with free cloud storage accounts. This offered space could be 10, 20 or even 50GB. While this is a decent additional, remember that not all information and files can be stored to cloud storage (applications for instance). You will likewise be not able get to files put away in the cloud on the off chance that you don't have a Wi-Fi or active mobile data pack.

Increasing Internal Storage Space

There are two or three conceivable approaches to make some additional space in your internal storage, contingent upon the mobile phone you have.

Impair Bloatware

Not all cell phones will give you a chance to do this, but rather on the off chance that you have an Android phone running form 4.2 or later, the procedure is genuinely simple. Although disabling 100MB pre-introduced application won't free up a relating measure of memory, it ought to positively make some additional space.

Backup and Clean Photographs

This is a decent practice to get into regardless of the possibility that storage space on your phone is not an issue. Utilize the sync program that is important to your handset to reinforcement your photographs to your PC routinely. You are then ready to erase the photographs off of your phone (or possibly some of them) to free up some space.

For example, Cleanmaster is an application to clear unneeded or undesirable documents from your phone, frequently at the touch of a button. Once more, this measure won't free up immense measures of space, yet it can have some effect.

Un-installing-necessary or un-used Apps

In the event that you are anything like me, you will be liable of downloading the most recent prevailing fashion application, utilizing it always for half a month and afterward disregarding as you move onto the following. Check through your applications list and uninstall those you never again utilize. This is effectively done through the settings menu of a wide range of smart phone, or by utilizing an application like Cleanmaster.

In today's world of electronics, memory technology is an essential element. With the use of semiconductor technology, any device which uses processor embeds memory to store data/information. As the new devices in market come with the microprocessors, correspondingly the need for semiconductor memory is also increasing. Therefore, new and advanced semiconductor memory technologies are being researched. With the increased growth for semiconductor memories, there are a number and types of memories that emerged in the market. Some of them are- RAM, ROM, PROM, EPROM, EEPROM, and Flash memory, DRAM, SRAM, SDRAM and MRAM etc. Each kind of memory has its own advantages and features. Semiconductor memories technology is moving forward with pace. With the increase in the size of the memories, data density is also improving. Various new forms of memories are also introduced in the past few years like MRAM (Magneto resistive Random Access Memory). As the requirements for larger, faster and lower power consuming memories are increasing, the area of this technology will be one of the most dynamic in the electronics industry.

9.7 SOLUTIONS/ANSWERS

1) Volatile
2) Multi Media Card
3) Erasing, Programming
4) Random

9.8 FURTHER READINGS

- How Computers Work: Processor and Main Memory by Roger Young; Create space Independent; ISBN-13: 9781442113985, ISBN-10: 1442113987.
- Media and Memory (Media Topics) by Joanne Garde-Hansen; Edinburgh University Press; ISBN-10: 0748640339, ISBN-13: 978-0748640331.
- https://en.wikipedia.org/wiki/File:15-04-29-MMC-Karte-RalfR-dscf4734-d.jpg
- http://www.electronics.dit.ie/staff/tscarff/memory/prom.gif

- http://www.electronics.dit.ie/staff/tscarff/memory/eprom.gif
- https://en.wikipedia.org/wiki/Memory
- https://en.wikipedia.org/wiki/Computer_memory
- https://en.wikipedia.org/wiki/MultiMediaCard

UNIT 10 SENSORS

Structure

10.0 INTRODUCTION

In our everyday life, we see many articles, for example, touch-sensitive elevator and lights which diminish or light up by touching the base, other than various uses of which the vast majority are never aware. With fast improvement in the field of smaller scale hardware and simple to-utilize micro-controllers, the employments of sensors have extended beyond the traditional fields of temperature, pressure or flow estimation. Moreover, analog sensors like potentiometers and force-sensing resistors are still usually utilized. You may utilize sensors in assembling and apparatus, planes and aviation, autos, medicine, robotics and numerous aspects of our everyday life. A sensor is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a Computer processor. A sensor is always used with other electronics, whether as simple as a light or as complex as a computer.. A sensor is constantly utilized with different electronic gadgets, regardless of whether as basic as a light or as complex as a PC. To the extent specialized meaning of sensor is considered, sensors are refined gadgets that are every now and again used to recognize and react to electrical or optical signs. Let’s consider the case of temperature. In the glass thermometer, mercury extends and gets the fluid to change over the deliberate temperature which can be perused by a viewer on the adjusted glass tube. At the end of the day, a sensor is a gadget that identifies and reacts to some sort of contribution from the physical condition.

Figure 10.1 shows working of touch screen of a mobile device.

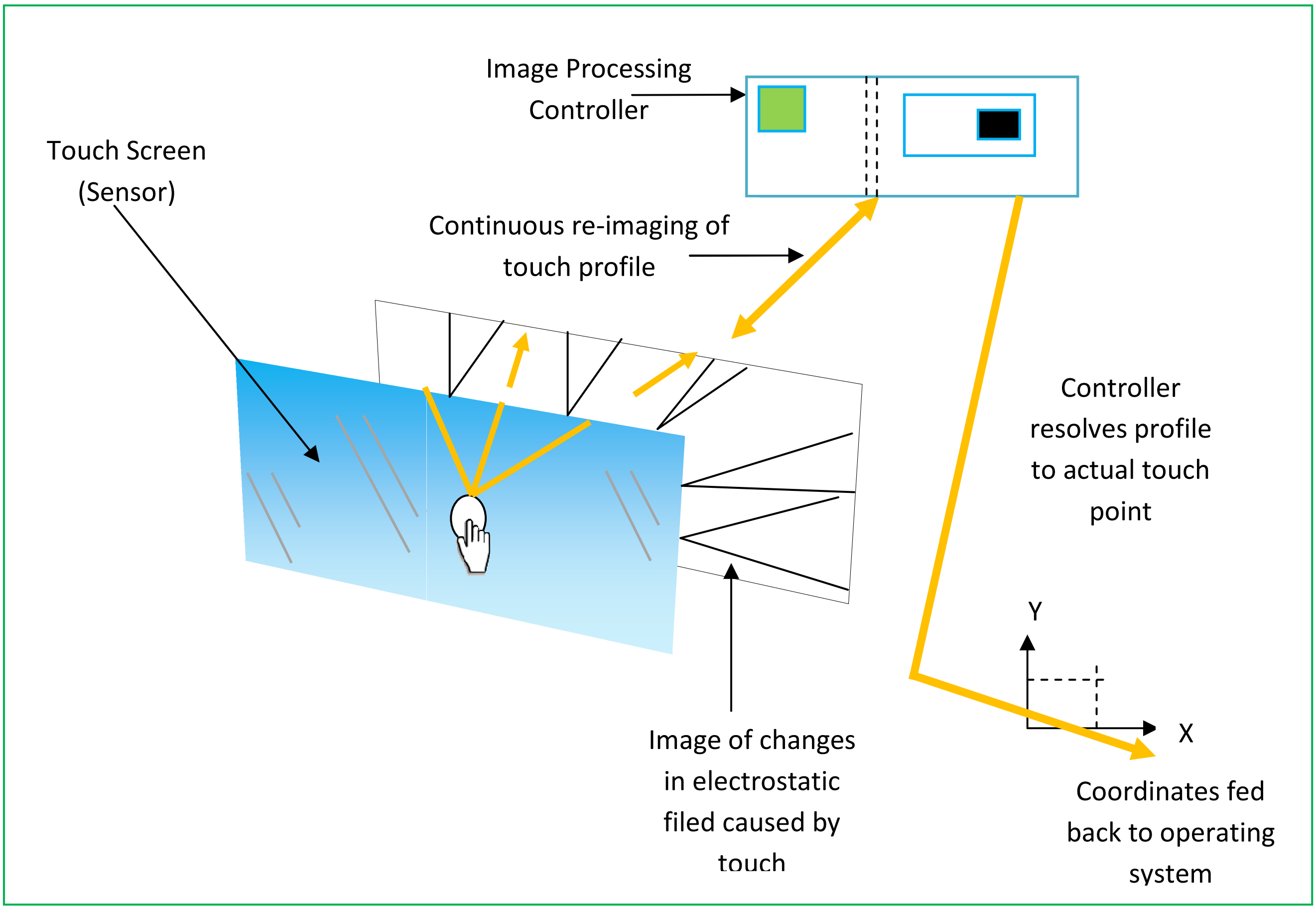


Image Processing Controller
Touch Screen (Sensor)
Continuous re-imaging of touch profile
Controller resolves profile to actual touch point
Y
X
Image of changes in electrostatic filed caused by touch
Coordinates fed back to operating system

Figure 10.1 : Working of a Touch Screen of a Mobile Device

10.1 OBJECTIVES

After studying this unit, you should be able to know about:

- Gyroscope
- Accelerometer
- Compass
- Proximity sensor

10.2 GYROSCOPE

A gyroscope is a device used for measuring or maintaining orientation and angular velocity.

It's a device which is used to determine orientation with the help of Earth's gravity. The design of gyroscope comes up with a freely-rotating disk known as a rotor, mounted onto a spinning axis in the center of a larger and extra balanced wheel. As the axis turn, the rotor remains stationary to imply the central gravitational pull, and thus which direction is down. The degree of strength of the gyroscope can be maintained by measuring its rate of rotation round a particular axis. When estimating the rate of revolution around the roll axis of an aircraft, it places an actual value until the object stabilizes out. Applying the central principles of angular momentum, the gyroscope helps indicate orientation.

Applications of gyroscopes include the inertial route systems where attractive compasses would not act, consistently in the Hubble telescope, or inside the steel structure of a submerged submarine, or would not be adequately right. Because of their accuracy, gyroscopes are likewise utilized as a part of gyrotheodolites to maintain direction in tunnel unearthing. Gyroscope can be utilized to develop gyrocompasses, which supplement or supplant magnetic compasses (in boats, flying machine and shuttle, vehicles by and large), to aid stability (Hubble Space Telescope, bikes, motorcycles, and ships) or be utilized as a major aspect of an inertial navigation system. A gyroscope would be applied in an aircraft to assist in indicating the rate of revolution about the aircraft roll axis. As an aircraft rolls, the gyroscope will measure non-zero values until the platform leaves out, whereupon it would record a null value to designate the direction of “down.” The best example of reading a gyroscope is that of the attitude indicator on typical aircrafts. It is typified by a circular display with the screen divided in half, the top half being blue in color to indicate sky, and the bottom being red to indicate ground. As an aircraft banks for a spell, the orientation of the display will transfer with the bank to account for the actual direction of the Earth. Figure 10.2 shows gyroscope.

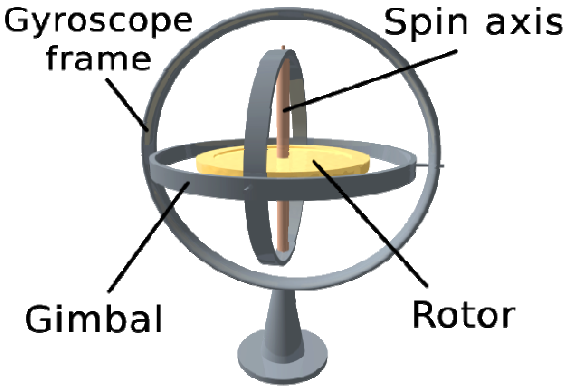


Figure 10.2 : Gyroscope

A gyroscope sensor has the following basic specifications:

- Measurement scale
- Sensing axes
- Non-linearity
- Operating temperature scale
- Shock sustainability
- Transmission Speed
- Noise Measurement
- Bias
- Bias Drift
- Bias Instability

Table 10.1: Describes various specifications of a Gyroscope.

Specification	Description
Measurement scale	This parameter decides the greatest angular speed with which the sensor can quantify, and is typically in degrees per second (˚/Sec).
Sensing axes	Gyroscopes are available that measure angular rotation in 1, 2, or 3 axes. Multi-axis detecting gyros have various single-axes gyros arranged orthogonal to each other. Vibrating plan gyroscopes are usually single-axis gyros or double-axis gyros, and revolving and visual gyroscope systems usually measure rotation in 3 axes.
Non-linearity	Gyroscopes deliver a voltage which is equal to the detected angular rate. Non-linearity is a bar of how near to linear the outputted voltage is in respect to the real angular rate. By not considering the non-linearity of a gyro, estimation can bring about some mistake. Estimation of non-linearity is figured as a rate error from a linear fit over the full-scale extend, or a mistake in parts per million.
Operating temperature scale	The greater part of the gadgets works just in some scope of temperatures. Working temperatures for gyroscopes are very substantial; their working temperatures go from generally - 40˚C to anyplace between 70 and 200˚C and have a tendency to be quite linear with temperature. Numerous gyros are accessible with a locally available temperature sensor, and afterward one doesn't have to stress over temperature related adjustment issues.
Shock sustainability	In frameworks where both linear acceleration and angular rotation rate are measured, it is vital to perceive how much force the gyroscope can withstand before breaking. Luckily gyroscopes are extremely rich, and can hold up a big container (over a short span) without ceasing. This is normally measured in g's (1g = earth's acceleration because of gravity), and at times the fourth dimension with which the most extreme g-force can be given before the unit fails is likewise displayed.
Transmission Speed	The data transfer capacity of a gyroscope regularly measures what number of measurements can be drawn every second. Hence the gyroscope transmission capacity is typically cited in Hz.
Angle Random Walk	Angle random walk (ARW) is a noise determination, in units of deg/hr1/2 that is specifically appropriate to angle computations. ARW depicts the normal deviation or mistake that will happen when you incorporate the signal. The "Random walk" is utilized as a part of statistics to portray a circumstance where

	the output of a system is driven by arbitrary, uncorrelated “steps”.
Bias	At the point when there is no rotation, the bias, or bias error, of a rate gyro is the signal output from the gyro Even the absolute best gyros in the world have mistake sources and bias is one of these wrongdoings. Bias can be appeared as a voltage or a percent of full scale output, yet basically it speaks to a rotational speed (in degrees every second). Once more, in a perfect universe, one could make allowance for a fixed bias error. Unfortunately a bias error has a tendency to differ, both with temperature and over time. The bias error of a gyro is because of various parts: • Calibration blunders • Switch-on to switch-on • bias drift • effects of shock (g level) Single estimations of bias are excessively influenced by noise, which is the reason an important bias measurements is always an averaged series of measurements.
Bias Drift	This alludes particularly to the change of the bias after some time, acquiring every single other segment stay constant. Fundamentally, this is a warm-up impact, prompted by the self heating of the gyroscope and its related mechanical and electrical parts. This force would be required to be more dominating over the initial few moments after change on and to be nearly non-existent after (say) five minutes.
Bias Instability	Bias Instability is a simple paradigm of the "goodness" of a gyroscope. It is indicated as the base point on the Allan Variance curve, typically measured in °/hr. It outlines the best bias security which could be achieved for a given gyroscope expecting that bias averaging executes in the period recognized as the Allan Variance least 9.

10.3 ACCELEROMETER

An accelerometer is a device that measures proper acceleration. It’s used to quantify the best possible acceleration which is not like coordinate acceleration. For example, an accelerometer at rest on the airfoil of the Earth will measure acceleration because of Earth's gravity, straight upwards (by definition) of g ≈ 9.81 m/s2. By complexity, accelerometers in free fall (going down toward the core of the world at a rate of roughly 9.81 m/s2) will measure zero. Accelerometers have different applications in industry and scientific discipline. Exceedingly delicate accelerometers are parts of inertial route navigation systems for aircraft and rockets. Likewise they are utilized to oversee screen vibration in rotating apparatus. Accelerometers are utilized as a part of tablet PCs and digital cameras so that pictures on screens are constantly shown upright. Accelerometers are utilized in drones for flight adjustment. Coordinated accelerometers can be utilized to measure differences in appropriate acceleration, especially gravity, over their partition in space; i.e., gradient of the gravitational field. This gravity gradiometry is valuable since supreme gravity is a weak impact and relies on upon local density of the Earth which is quite variable. Single-and multi-axis models of accelerometer are accessible to recognize magnitude and direction of the proper acceleration, as a vector amount, and can be utilized to detect orientation (since direction of weight changes), organize acceleration, vibration, shock, and falling in a resistive medium (a situation where the best possible acceleration changes, since it begins at zero, then increments). Small scale machined accelerometers are progressively present in convenient electronic gadgets and computer game controllers, to recognize the position of the gadget or accommodate game input.

Types of Accelerometer

There are few unique standards whereupon an analog accelerometer can be constructed. Two exceptionally common types use capacitive sensing and the piezoelectric effect to detect the displacement of the evidence mass relative to the applied acceleration. Table 10.2 explains them in detail.

Table 10.2 : Capacitative and Piezoelectric Accelerometer

Capacitive	Accelerometers that execute capacitive sensing output a voltage subject to the distance between two planar surfaces. Either of these "plates" is charged with an electrical current. Changing the gap between the plates changes the electrical limit of the system, which can be measured as a voltage output. This strategy for detecting is known for its high precision and stability. Capacitive accelerometers are additionally less prone to noise and variation with temperature, normally disseminates less power, and can have larger transfer speeds because of inner feedback circuitry.
Piezoelectric	Piezoelectric detecting of acceleration is normal, as acceleration is straightforwardly corresponding to force. At the point when certain sorts of crystal are compressed, charges of inverse polarity accumulate on inverse sides of the crystal. This is known as the piezoelectric effect. In a piezoelectric accelerometer, charge aggregates on the crystal and is translated and amplified into either an output current or voltage. Piezoelectric accelerometers just react to AC phenomenon, for example, vibration or shock. They have a wide powerful range, however can be costly relying upon their quality. Piezo-film based accelerometers are best used to measure AC phenomenon, for example, vibration or shock, as opposed to DC phenomenon, for example, the acceleration of gravity. They are cheap, and react to other phenomenon, for example, temperature, sound, and pressure.

Table 10.3: Describes accelerometers that are less used in audio applications.

Piezoresistive	Piezoresistive accelerometers (otherwise called Strain gage accelerometers) work by measuring the electrical resistance of a material when mechanical anxiety is connected. They are favored in high shock applications and they can measure acceleration down to 0Hz. Be that as it may, they have a constrained high frequency reaction.
Hall effect	Corridor impact accelerometers work by measuring the voltage varieties brought about by the change in magnetic field around them.
Heat transfer	Warm move accelerometers comprise in a solitary heat source focused in a substrate and suspended across cavity. They incorporate similarly separated thermoresistors on the four side of the heat source. They measure the inward changes in heat because of acceleration. At the point when there is zero acceleration, the heat gradient will be symmetrical. Otherwise, under acceleration, the heat gradient will become asymmetrical because of convection heat exchange

The following are other types of accelerometer:

- Null-balance
- Servo force balance
- Strain gauge
- Resonance
- Optical
- Surface acoustic wave (SAW)

10.3.1 Specifications of an Accelerometer

A typical accelerometer has the following basic specifications:

- Analog/digital
- Number of axes
- Output range (maximum swing)
- Sensitivity (voltage output per g)
- Dynamic range
- Bandwidth

- Amplitude stability
- Mass

Analog and Digital accelerometers The most essential detail of an accelerometer for a given application is it's kind of output. Analog accelerometers output a constant factor variable relying upon the measure of acceleration applied. Earlier digital accelerometers output a variable frequency square wave, a technique known as pulse-width modulation. A pulse width modulated accelerometer takes readings at a fixed rate, regularly 1000 Hz (however this might be user-configurable in light of the IC selected). The value of the acceleration is corresponding to the pulse width (or obligation cycle) of the PWM signal. Newer digital accelerometers will probably output their value utilizing multi-wire digital protocols, for example, I2C or SPI.

For use with ADCs generally utilized for music collaboration systems, analog accelerometers are typically preferred.

Number of axes Accelerometers are accessible that measure in 3 dimensions. The most well-known kind of accelerometer measures across two axes. However, three-axis accelerometers are progressively normal and economical.

Output range To quantify the acceleration of gravity for use as a tilt sensor, an output scope of ±1.5 g is adequate. For use as an effect sensor, a standout amongst the most widely recognized melodic applications, ±5 g or more is desired.

Sensitivity A marker of the measure of progress in output signal for a given change in acceleration. A sensitive accelerometer will be more exact and most likely more precise.

Dynamic range The range between the smallest acceleration is discoverable by the accelerometer to the largest before distorting or clipping the output signal.

Bandwidth The data transmission rate of a sensor is typically measured in Hertz and demonstrates the capacity of the near-unity frequency reaction of the sensor, or how frequently a reliable reading can be taken. People can't make body movement much beyond the scope of 10-12 Hz. Thus, a data transmission of 40-60 Hz is sufficient for tilt or human movement detecting. For vibration estimation or exact reading of impact forces, transmission capacity ought to be in the scope of hundreds of Hertz. It ought to likewise be noticed that for some older microcontrollers, the data transmission rate of an accelerometer may extend beyond the Nyquist frequency of the A/D converters on the MCU, so for higher transmission rate sensing, the advanced signal might be associated. This can be resolved with simple passive low-pass filtering before sampling, or by basically picking a superior microcontroller. It is important that the data transmission rate or bandwidth may change by the way the accelerometer is mounted. A stiffer mounting will support to maintain a higher usable frequency range and the inverse will diminish it.

Mass The mass of the accelerometer ought to be fundamentally smaller than the mass of the system to be observed with the goal that it doesn't change the characteristic of the object being tested.

The following are some of the other details associated with accelerometer:

- Zero g balance (voltage output at 0 g)
- Noise (sensor least resolution)

- Temperature scale
- Bias drift with temperature (impact of temperature on voltage output at 0 g)
- Sensitivity drift with temperature (impact of temperature on voltage output per g)
- Power utilization

10.3.2 Output of an Accelerometer

An accelerometer output value is a scalar relating to the magnitude of the acceleration vector. The most widely recognized acceleration, and one that we are always presented to, is the acceleration that is a consequence of the earth's gravitational pull. This is a typical reference value from which every single other accelerations are measured (known as g, which is ~9.8m/s^2).

Digital output Accelerometers with PWM output can be utilized as a part of two diverse ways. For most exact outcomes, the PWM signal can be input directly to a microcontroller where the obligation cycle is perused in firmware and converted into a scaled acceleration value. (Check with the datasheet to acquire the scaling component and required output impedance.) When a microcontroller with PWM information is not accessible, or when different methods for digitizing the signal are being utilized, a simple RC reconstruction channel can be utilized to get a analog voltage proportional to the acceleration. At rest (half obligation cycle) the output voltage will show no acceleration, higher voltage values (coming about because of a higher obligation cycle) will show positive acceleration, and lower values (<50% obligation cycle) show negative acceleration. These voltages can then be scaled and utilized as one may the output voltage of an analog output accelerometer. One weakness of a digital output is that it takes somewhat more timing resources of the microcontroller to measure the obligation cycle of the PWM signal. Correspondence protocols could utilize I2C or SPI.

Analog output At the point when contrasted with most other modern sensors, analog accelerometers require small conditioning and the correspondence is basic by just utilizing an Analog to Digital Converter (ADC) on the microcontroller. Regularly, an accelerometer output signal will require a offset, amplification, and filtration. For analog voltage output accelerometers, the signal can be a positive or negative voltage, contingent upon the course of the acceleration direction. Additionally, the signal is persistent and relative to the acceleration force. Similarly as with any sensor bound for a analog to digital converter, the value must be scaled or potentially increased to maximize the range of acquisition. Most analog to digital converters utilized as a part of melodic applications acquire signals in the 0-5 V range.

The picture at right depicts an amplification and offset circuit, including the on-board operational amplifier in the ADXL105, limiting the requirement for extra IC components. The gain connected with output is set by the proportion R2/R1. The balance is controlled by biasing the voltage with variable resistor R4. Accelerometers output bias will float as indicated by surrounding temperature. The sensors are adjusted for operation at a particular temperature, commonly room temperature. However, in most brief length indoor applications the balance is generally consistent and stable, and in this way does not require modification. If the sensor is proposed to be utilized as a part of different situations with varying surrounding temperatures, the bias capacity ought to be adequate for analog alignment of the gadget. In the event that the ambient temperature is liable to uncommon changes through the span of a solitary use, the temperature output ought to be summed into the bias circuit. Brilliant sensors may even think about this.

The resolution of the information gained is at last controlled by the analog to digital converter. It is conceivable, in any case that the noise floor is over the base resolution of the converter, diminishing the resolution of your system. Accepting that the noise is similarly circulated over all frequencies, it is conceivable to channel the signal to just incorporate frequencies inside the scope of operation. The channel required relies on both kind of acquisition and the location of the sensor. The transfer speed is essentially impacted by the three distinct methods of operation of the sensor.

10.3.3 Applications of an Accelerometer

The following are some of the applications of an accelerometer:

- The sensor can be executed in a system that identifies speed, position, shock, vibration, or the acceleration of gravity to decide orientation.
- A system comprising of two orthogonal sensors which are equipped for detecting pitch and roll. This is helpful in catching head movements. A third orthogonal sensor can be added to the network to acquire orientation in three dimensional space. This is proper for the detection of pen angles, and so forth. The detecting abilities of this network can be advanced to six degrees of spatial measurement freedom by the expansion of three orthogonal gyroscopes.
- As a shock detector, an accelerometer is searching for changes in acceleration. This jerk is detected as an over damped vibration.

Data transmissions related with different usage of accelerometers as input devices were illustrated in Table 10.4.

Table 10.4 : Data transmission rates with accelerometers

Location	Usage	Frequency	Acceleration
Head	Tilt	0-8 Hz	xx
Hand , Wrist, Finger	Cont.	8-12 Hz	0.04-1.0 g
Hand, Arm, Upper Body	Cont.	0-12 Hz	0.5-9.0 g
Foot, Leg	Cont.	0-12 Hz	0.2-6.6 g

Depending upon the sensitivity and dynamic range required, the cost of an accelerometer can increase to thousands of dollars. In any case, exceptionally accurate reasonable sensors are available.

10.4 COMPASS

A Compass is an instrument used for navigation and orientation that shows direction relative to the geographic cardinal directions (or points). The advanced compass that is typically in light of a sensor called magnetometer gives cell phones a straightforward orientation in connection to the Earth's magnetic field. Accordingly, your smart phone dependably knows which way is North so it can auto rotating your digital maps relying upon your physical orientation. The cell phones have a little magnetometer implicit, which can measure the Earth's magnetic field. This data is consolidated with an accelerator that gets data in regards to the phone's position in space. It can pinpoint the phone's position from solid state sensors inside the phone that can measure their tilt and movement. The data given by these gadgets implies that the compass application can show cardinal directions regardless of which orientation the phone is in, as indicated by the algorithmic software development organization Sensor Platforms. The compass can decide north and south because of the magnet's association with the Earth's magnetic field. The reason for the magnetic field is not unmistakably known, but rather geologists have made speculations with respect to the phenomenon by analyzing the layers of the Earth. The Earth is comprised of an outer crust, trailed by the upper mantle, the inward mantle, the outer core, and after that at long last the inner core at the very center. The inner core is made up for the most part of liquid iron, however the extreme center of the inner core is under so much pressure that the liquid iron is in a solid form. It is trusted that the rotation of the Earth and the tremendous warmth from the core make the iron move in a rotational manner. This rotational manner might be the source of the magnetic field that we see on Earth. The field created is extremely feeble, be that as it may, which is the reason the needle on the compass should be exceptionally lightweight and on a surface with minimal friction.

10.5 PROXIMITY SENSOR

A proximity sensor is a sensor that is able to detect the presence of nearby objects without any physical contact. A proximity sensor regularly transmits an electromagnetic field or a beam of electromagnetic radiation (infrared, for example), and searches for alterations in the field or return signal. The object being sensed is regularly concerned to as the proximity sensor's target. Distinctive proximity sensor targets request different sensors. Like for a plastic target, capacitive or photoelectric sensor may be reasonable and for a metal target, an inductive proximity sensor is constantly required. This sensor can detect maximum distance up to its defined "nominal range". Proximity sensors can have a high reliability and long functional life due to the absence of mechanical parts and absence of physical contact between sensor and the detected object. Proximity sensors are usually utilized on mobile phones to detect and skip incidental touchscreen taps when held to the ear amid a call. They are likewise utilized as a part of machine vibration checking to measure the variation in distance between a shaft and its support bearing. This is basic in large steam turbines, compressors, and engines that utilize sleeve-type bearings. International Electrotechnical Commission (IEC) characterizes the specialized subtle elements of proximity sensors. A proximity sensor changed in accordance with a short range is regularly utilized as a touch switch.

Check Your Progress 1

1) A ______ is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor.

2) A _______ is a device used for measuring or maintaining orientation and angular velocity.

3) An __________ is a device that measures proper acceleration.

4) A _________ is an instrument used for navigation and orientation that shows direction relative to the geographic cardinal directions.

5) A ________ is a sensor that is able to detect the presence of nearby objects without any physical contact

10.6 SUMMARY

Now a day, mobile phones accompanied with variety of sensors that simplify or automate our day to day assignments. This field considers the presence of an accelerometer, a compass, a gyroscope and a barometer.

Accelerometers in smart phones are utilized to recognize the orientation of the phone. The gyroscope, adds an extra dimension to the data provided by the accelerometer by following rotation or twist. Linear acceleration of movement is measured by accelerometer whereas gyroscope measures the angular rotational speed. Both sensors measure rate of progress; they simply measure the rate of progress for various things. Both sensors measure rate of progress; they simply measure the rate of progress for various things. In practice, accelerometer will quantify the directional movement of a gadget however won't have the capacity to determine its lateral orientation or tilt during that movement precisely unless a gyro is there to fill in that data. With an accelerometer you can either get a truly "loud" information output that is responsive, or you can get a "spotless" output that is slow. In any case, when you join the 3-pivot accelerometer with a 3-axis gyro, you get an output that is both perfect and responsive in a similar time.

The digital compass that is generally in view of a sensor called magnetometer gives cell phones a basic orientation in connection to the Earth's magnetic field. Thus, your phone dependably knows which way is North so it can auto rotating your digital maps relying upon your physical orientation.

Lastly, you may see a gadget sporting a barometer in its specification sheet. As opposed to what you may recommend, it has nothing to do with climate. Rather, the barometer is there to help the GPS chip inside the gadget get a quicker lock by immediately conveying altitude information.

10.7 SOLUTIONS/ANSWERS

1) Sensor

2) Gyroscope

3) Accelerometer

4) Compass

5) Proximity sensor

10.8 FURTHER READINGS

- Introduction to Sensors by John Vetelino and Aravind Reghu;CRC press; ISBN-10: 143980852X, ISBN-13: 978-1439808528.
- Sensors and Transducers by Ian Sinclair; Newnes; Third edition; ISBN-10:0080516998, ISBN-13:978-008051699.

- https://en.wikipedia.org/wiki/Sensor
- https://en.wikipedia.org/wiki/Gyroscope
- https://en.wikipedia.org/wiki/Accelerometer
- https://en.wikipedia.org/wiki/Compass
- https://en.wikipedia.org/wiki/Proximity_sensor
- http://www.te.com/usa-en/products/sensors.html
- https://science.howstuffworks.com/gyroscope.htm
- https://www.gsmarena.com/glossary.php3?term=accelerometer
- http://www.machinedesign.com/sensors/proximity-sensors-compared-inductive-capacitive-photoelectric-and-ultrasonic
- http://www.sensorwiki.org/doku.php/sensors/accelerometer

UNIT 11 INPUT-OUTPUT

Structure

11.0 INTRODUCTION

Any information or data which is sent to a computer/mobile device for processing is considered input. Input is most often sent to the device using an input device, the following pictures are examples of input being sent from the computer keyboard to the computer and using touchpad or virtual keyboard to mobile.

An output device is referring to a device or electronic equipment which is utilized for getting information from another gadget, however it can't send information to another gadget, for example, PC screen, projector, and speakers, which can get information (output) from the PC, yet they can't send data (input) to the PC. Similarly, in mobile phones, output device is the display screen of the phone where you see result of the action performed, like watching videos, viewing pictures etc. Hence we can say an input/output device is the device that can send data to a system for processing, and reproduces or displays the results of that processing. The output is depends upon on the interaction, a device can be both, referred to as an input/output or I/O device.

11.1 OBJECTIVES

After going through this unit, you should be able to:

- know about various input/output operations;
- understand different input/output interfaces; and
- understand the technology behind I/O devices.

11.2 DISPLAY

Display is the most important interface between humans and machines, with the ability to present information in terms of text, graphics, animation, and video. Historically, the display industry was stuck in neutral compared to the rest of the high tech sector, which has evolved at warp speed. For more than eight decades, CRT (Cathode Ray Tube) was the king of display that has quickly changed now. Today, display devices, such as LCD, Plasma, DLP (digital light processing), LCoS (Liquid crystal on silicon), and SED (Surface-conduction electron-emitter display) have suddenly emerged. Driven by computers, high definition TV (HDTV), and mobile devices (including cell phones), the display industry is evolving at a much faster rate than ever before. With so many options, choosing the best display solution for you mobile device has become a real challenge. First, let's start by asking which display features and parameters are favorable to the application. Next, let's define some important general display performance attributes, as shown in the following table. Table 11.1 also describes some important mobile display performance attributes.

Table 11.1 : Mobile display performance attributes

Display Attribute	Description
Resolution	Number of pixels in X by Y dimensions For example, 1024 x 768
Screen size	Size of image area, as measured diagonally in inches For example, 16" or 42"
Contrast ratio	Brightest color (white) over darkest color (black) of display luminance
Brightness	Luminance
Response time	Display update transition time
Uniformity	Evenness in brightness and color
Pixel or color depth	Number of bits per pixel
Color gamut	Subset of colors that can be accurately represented
Refresh rate	Number of vertical scans per second (This is different from the Frame rate, which usually refers to the number of frames the host renders to the display device per second.)
Viewing angle	Maximum angel at which the display is properly visible, in horizontal and vertical directions.

As mobile and handheld devices grow exponentially, so do mobile displays. In general mobile displays are smaller than general displays, but size is not the only thing that matters. The previous section defined performance attributes of general displays. But mobile displays have additional requirements and attributes as outlined in table 11.2.

Table 11.2 : Attributes of a mobile display

Mobile Display Attribute	Description
Low power	Critical to save battery power
Thin	Low profile mobile form factor
Lightweight	Easy to carry around
Outdoor reading	Readable even in sunlight
Durable	Robust
Bi-stability	Retains image without refreshing and even without power. This can conserve significant power for still image display.
Pixel density	High ppi (pixel per inch) is needed for short-distance viewing

Presently, we can discover fast change since the days mobile displays required just to show the number of the individual calling. When we included messaging and emails, we require more space to perceive what we'd composed. We added color to give them more intrigue. When we began including cameras to the mobile phones, we needed the display to be sharper, so that we could view the pixelated VGA photographs we'd taken. Once the capacity to keep video reached, we required them to be smooth, with great refreshing rates. In the period of touch screen cell phones, we anticipate them will be on a par with PC display, providing crisp text, dynamic pictures, obscure free video and sufficient brightness to view outdoors, all via a responsive touch screen. Typically, a wide range of choices have emerged, particularly with regards to top of the line smart phones.

We've gathered all the essential data about various mobile screens which are explained in table 11.3.

Table 11.3 : Types of mobile screens

Screen Type	Description
TFT-LCD	TFT-LCD is the most well-known kind of display screen utilized as a part of mobile phones which ranges from smart phones like the HTC Desire C to first class tablets, like the Google Nexus 7. The thin-film transistor liquid crystal display known as TFCT-LCD; and there are varieties of ways for manufacturing LCD screens. Practically speaking, low-budget mobile screens will generally show dull colors, and have narrow viewpoint angles i.e. if you see at them from off-center; it ends up noticeably thick to perceive what's on-screen. LCD screens with High quality will have radiant, perfect colors and with lucidity from pretty much any angle.

AMOLED (Active-matrix organic light-emitting diode)	AMOLED displays don't require a backdrop illumination - every pixel generates its own light - so smart phones utilizing them can possibly be slimmer. It likewise implies that a generally dark screen will utilize next to no power, and genuine blacks while playing videos, instead of the dim some LCD screens deliver. Notwithstanding, AMOLED screens have demonstrated exorbitant and hard to produce in an indistinguishable numbers from LCD, a reality that prompted the HTC Desire powered by AMOLED display; replaced with Super-LCD part of the way through its assembling life. AMOLED utilizes a fluctuated subpixel course of action to LCD, so that pictures that don't seem quite as sharp.
Super-AMOLED	Super AMOLED is extended form of AMOLED screen that really comprise of capacitive touch screen innovation i.e. it doesn't need to be laminated later. It offers expanded brightness and minimal power utilization over prior AMOLED screens.
Super-AMOLED Plus	This is another innovation initially utilized by the Samsung in their Samsung Galaxy S2 Handset. The huge change is in the subpixel development, changing to something substantially nearer to that utilized by LCD, which implied sharper, clearer pictures.
Super AMOLED HD	Super AMOLED HD is essentially a HD (720 x 1280 or higher) display innovation. In an odd move Samsung disposed of the subpixel development from super AMOLED Plus and returned rather to a standard Super AMOLED subpixel arrangement for this display, asserting that it was more dependable. It was first utilized as a part of the Samsung Galaxy Note. Obviously, the Samsung Galaxy S3 likewise has a screen of this type.
Retina display	A showcasing term brought by Apple, a Retina display is a display with a pixel thickness sufficiently high that the human eye can't make out any individual pixels at a typical review separate. As it were it's a show with no pixilation by any means. Retina show fluctuates between gadgets as it is reliant on viewpoint seeing distance. iPhone X, which would be held near the face, has 458 pixels for every inch, while an iPad pro just had 264 pixels for each inch.
Super-LCD	This is a particular sort of TFT-LCD screen, and has been broadcasted as matching AMOLED for picture quality. It gives bring down power

	utilization than most LCD advancements, yet without trading off any photo quality.
Mobile BRAVIA Engine	BRAVIA is a Sony mark and has for some time been utilized for their TVs, it remains for Best Resolution Audio Visual Integrated Architecture which. The mobile BRAVIA Engine is comprised of various picture handling advancements and is intended to enhance pictures and videos by making them more sharper and diminishing noise. It likewise intends to enhance the contrast and make more natural colors. It's but rather a screen sort a suite of post-preparing impacts which can be turned on or off. The mobile BRAVIA Engine is utilized solely in Sony handsets, for example, the Sony Xperia S.
NOVA	LG utilizes this LCD-determined screen innovation in their Optimus Black phone which offers immensely expanded shine while being highly power saver.

Table 11.4 shows resolutions of various types of mobile displays.

Table 11.4 : Resolutions of mobile displays

Mobile Phone Display Resolutions	
QVGA (240 x 320)	QVGA remains for 'Quarter Video Graphics Array'; it is generally utilized for economical phones. Its resolution is quarter the resolution of VGA. The HTC Wildfire and the Samsung Galaxy Y supports this resolution.
WQVGA	WQVGA is characterized as the display that has similar height from QVGA (240) however a more prominent width. The "W" here stands for wide. The Sony Ericsson Aino has a WQVGA resolution of 240x432.
HVGA (320 x 480)	The "H" here stands for "Half" or 'half size' VGA. It alludes to a modest bunch of resolutions, yet the most well-known one is 320 x 480, which is 1/2 the size of VGA. The iPhone 3GS has a resolution of 320 x 480 thus too does the HTC Wildfire S.
nHD (360 x 640)	This resolution is not ordinarily utilized by phones, but rather can be seen on the Nokia N97 or the later Nokia 808 PureView.
WVGA	It remains for "Wider" VGA (Video Graphics Array) and alludes to any display with a height of 480 pixels however a width more prominent than 640 pixels. The most widely recognized WVGA resolution is 480 x 800 and this can be found on heaps of smart phones like Samsung Galaxy S2, Nokia Lumia 900 and so on.

FWVGA (480 x 854)	FWVGA remains for 'Full Wide' VGA and not at all like WVGA it doesn't require to be trimmed to deliver an aspect ratio of 16:9.This resolution can be found on the Sony Ericsson Xperia Arc S.
qHD (540 x 960)	The "q" here stands for One fourth of a full HD outline; these sorts of high-resolution displays are being utilized by smart phones as screens are getting bigger and bigger in size. The HTC Sensation is an ideal case of this kind of resolution in action.
DVGA (640 x 960)	The "D" here stands for Double VGA, this is a high resolution for phones and is utilized by the iPhone 4 and iPhone 4S and is named Retina display by Apple. This stands for 'Extended Graphics Array' and isn't generally utilized as a part of phones; however the forthcoming LG Optimus Vu has a XGA display.
HD (720 x 1280)	HD remains for high definition resolution which is alluded to as 720p. It is most strikingly utilized by the Samsung Galaxy S3 and HTC One X.
WXGA	The "W" here stands for 'Wide Extended Graphics Array', this, as you may have speculated, is an enlarged XGA resolution. Various resolutions fall under the WXGA flag, yet the main smart phone to at present have a resolution this high is the Samsung Galaxy Note, which has a 800 x 1280 display.

Apple utilizes high pixels per inch count for its display, instead of a specific innovation, though Sony Ericsson's Reality display with mobile Bravia screen utilizes for software enhancements to the video to get great outcomes. In both circumstances, it is really the LCD innovation powering things. On the off chance that you might want to watch motion pictures and play games on your phone then AMOLED may be the better decision as it offers the tremendously better contrast ratio compared to LCD. In the event that web browsing and working on a file is more, LCD will be better as it offers somewhat crisper content and make it easier to read text.

11.3 CAMERA

A camera is an optical instrument for recording or capturing images, which may be stored locally, transmitted to another location, or both. A camera phone is a mobile phone which is able to capture photographs and often record video using one or more built-in digital cameras. Sharp Corporation brought the first camera phone J-SH04 in the commercial market. The camera phone J-SH04 had built-in CCD sensor, with the Sha-Mail (Picture-Mail in Japanese) infrastructure developed in collaboration with Kahn's LightSurf venture, and marketed from 2001 by J-Phone in Japan today owned by Softbank. Initials phones were equipped with Internet connectivity, working web browsers and email-programs but the phone menu offered no way of including a photo in an email or uploading it to a web site. Modern smart phones have almost unlimited connectivity and transfer options with photograph attachment features. During 2003 as camera phones were gaining popularity in Europe some phones without cameras had support for MMS and external cameras that could be connected with a small cable or directly to the data port at the base of the phone. The external cameras were comparable in quality to those fitted on regular camera phones at the time, typically offering VGA resolution. In 2013-2014 Sony and other manufacturers announced add-on camera modules for smart phones called lens-style cameras which can be mounted to an Android or iOS phone or tablet and use its display and controls.

Now a days, Camera in a mobile phone is very much common. You can easily find mobile phones which are coming up with in-built camera. It let users to take decent quality photographs and record a video; they are quite simpler than any digital camera. In most of the smart phones; you'll discover a menu to begin a camera application and an on-screen button to empower the shutter. For convenience, some of phones have a specific camera button. You can likewise discover phones which are particularly intended to resemble separate low-end digital compact cameras in appearance and to some degree in features and image quality, and are marked as both mobile phones and cameras. The fundamental advantage of camera phone is cost and portability; to be sure for a user who carries a mobile phone anyway, the expansion is irrelevant. smart phones with camera may run mobile applications to include abilities, for example, geotagging and picture stitching. Also the touch screen in smart phones can be used to direct their camera to focus on a specific object in the field of view; in any case, the touch screen, being a universally useful control, does not have specific camera's buttons and dial. Most of the camera phones use CMOS (complementary metal-oxide-semiconductor) image sensors, due to largely reduced power consumption compared to CCD (Charge Coupled Device) type cameras, which are also used, but in few camera phones. Some of the camera phones even use more expensive Back Side Illuminated CMOS which uses energy lesser than CMOS, although more expensive than CMOS and CCD. In smart phones, cameras are used as input devices in various research projects and commercial applications. The use of QR Codes attached to physical objects is a commercially successful example of camera in smart phones. Camera senses the QR Code and provides an according link to related digital content, usually a URL. Another approach is using camera images to recognize objects. The recognition of physical objects such as advertisement posters to provide information about the object can be done via content-based image analysis. Hybrid approaches use a combination of unobtrusive visual markers and image analysis. Some of the smart phones can provide an augmented reality overlay for 2D objects and to recognize multiple objects on the phone using a stripped down object recognition algorithm as well as using GPS(Global Positioning System) and compass. A few can translate text from a foreign language. Geotagging (also written as GeoTagging) is the process of adding geographical identification metadata to various media such as a geotagged photograph or video, websites, SMS messages, QR Codes[1] or RSS feeds and is a form of geospatial metadata. This data usually consists of latitude and longitude coordinates, though they can also include altitude, bearing, distance, accuracy data, and place names, and perhaps a time stamp.

Auto-geotagging can show the location where a picture is taken. Front camera (of lesser performance as compared to rear camera) of smart phones can be used for purposes like self-portraiture (selfie) and videoconferencing.

In Android phones, camera applications can be started by touching the Camera app icon as shown in figure 11.1.

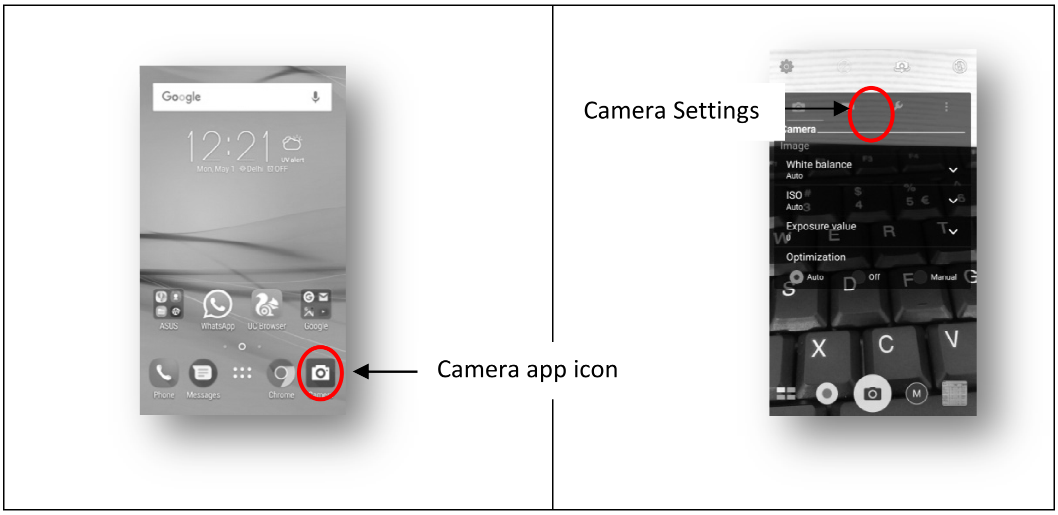


Figure 11.1: Camera app icon and settings

Over the years, camera phone technology has progressed a lot, the design of lens has evolved from a Cooke triplet or simple double Gauss to many molded plastic aspheric lens elements formed with varying dispersion and refractive indexes. The current generation of phone cameras also applies distortion (optics), vignetting, and various optical aberration corrections to the image before it is compressed into a .jpeg format. Most camera phones have a digital zoom feature. A few have optical zoom. An external camera can be added, coupled wirelessly to the phone by Wi-Fi. Images are usually saved in the JPEG file format, except for some high-end camera phones which have also RAW feature and the Android 5.0 Lollipop has facility of it. Microsoft Windows Phones can be configured to use as a camera even if the phone is asleep. An external flash can be employed, to improve performance. By default, phones stores videos and pictures in a folder called /DCIM. Storage can be changed to external memory.

11.4 SPEAKERS

Computer speakers, or multimedia speakers, are speakers sold for use with computers, although usually capable of other audio uses. Most of the speakers have built-in amplifier and simultaneously require a power supply, which may be by a primary power supply, USB port or batteries.

Wireless Bluetooth speakers need no physical connections at all as they are battery-powered. A wireless speaker receives audio signals using radio frequency (RF) waves rather than over audio cables. We can categorize speakers into following two types:

- Active Speakers
- Passive Speakers

11.4.1 Active Speakers

The crossover components in an active speaker which splits the frequency band of the sound signal into smaller parts which are then sent to individual speaker drivers configured to handle those frequencies. These speakers are, themselves "powered." This is why active and powered are often used synonymously.

Active speakers are suitable for home entertainment and home theater solutions. In larger home theater settings, the crossover components and amps can be outside of the speaker. Active speaker system refers to active crossover and two or three separate amplifiers for the separate drivers. According to Bart LoPiccolo, National sales manager for Genelec, Inc., Genelec's speakers are active because, "they have active electronic crossovers before the amps, they have dedicated amps for each driver, each channel has protection, and there are room response controls per driver." Active speaker proponents like LoPiccolo believe that an active speaker system has certain advantages over a passive system due to the fact that the crossover components, amps, drivers, and speaker enclosures are all manufactured with one another in mind.

11.4.2Passive Speakers

In comparison to the active speaker, passive speakers have following features:

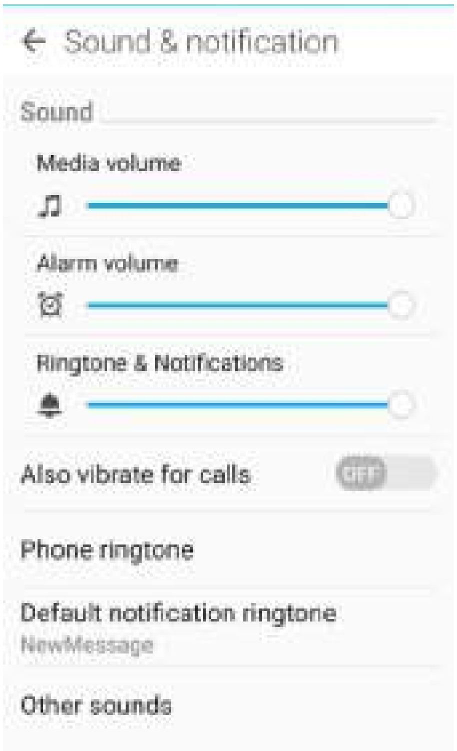
- Here crossover components splits the audio signal and send each band directly to the loudspeaker drivers
- A separate individual amp is used for driving the signals
- Here components are typically consists of capacitors, resistors and inductors which splits the signal and send each frequency part to the drivers
- a separate power amplifier sends the full audio signal to the speaker

Passive Speaker user has the freedom to upgrade the amplifier, or swap it out at any time, where in an active speaker scenario the amps are part of the speaker package i.e. what you hear is what you get, generally speaking. But, the producers of amplifiers don't know what speaker system will be matched with their product, which is why amplifiers are over-built to accommodate a wide range of speakers, a step that results in extra expensive and more power consuming amplifiers.

A powered speaker is technically one that has its own in-built amplifier, and therefore plugs into a nearby power outlet. However, a powered speaker is not essentially an active one, as the crossover components within a powered speaker may be passive. Powered speaker systems have the edge of being a bit more compact, streamlined and portable—most of the speaker systems are designed for desktop home computers, laptops, single-room application, wireless multi-room application (like Sonos), and easy all-in-one iPod-docking-and-speaker-solution fall into the "powered" speaker category. In reality, a powered speaker can still have the "amp-passive crossover-driver" chain feature of any passive speaker scenario. According to Logan Pabor, Distribution Manager at Audioengine, "Powered speakers can cut costs, and reduce clutter," The built-in amplifier characteristic of a powered speaker, enables wireless speaker scenarios, and eliminates the requirement for extra, expensive components.

It will be a big debate, if we try to find out one technology as superior over another in the active vs. passive speaker vs. powered or unpowered speaker. Every system has some pros and cons, and ways of perfecting (as nearly as possible) one system to "match" the performance of another. The mobile market for audio products is expanding, creating freedom for sound designers, content providers, and record companies. Due to an insatiable driven consumer demand, audio technologies for mobile devices are being developed and deployed at a spectacular rate. The feature to personify cell phones with audio (particularly in the form of ringtones) is a great motivational factor for consumers to spend money, which generates extra revenue options for carriers eager to recoup their massive infrastructure investments. However, the plethora of file formats and operating systems makes audio content creation and delivery a difficult and inefficient proposition. Content providers have to maintain multiple catalogs in different formats and resolutions. Game audio designers and programmers must either re-invent the wheel for each mobile title, or forgo complex interactive soundtracks altogether. Although Java-based systems attempt to address some of these problems, audio is still severely limited by limited bandwidth constraints and cross-platform incompatibilities. The ultimate quality of any sound system depends upon the speakers.

In android phones, sound/audio can be configured though settings menu as shown in figure 11.2.



← Sound & notification

Sound

Media volume

Alarm volume

Ringtone & Notifications

Also vibrate for calls OFF

Phone ringtone

Default notification ringtone
NewMessage

Other sounds

Figure 11.2 : Audio settings in an Android phone

11.5 MIC

A microphone, colloquially nicknamed mic or mike is a transducer that converts sound into an electrical signal. Numerous applications utilizes microphones, for example, phones, amplifiers, open address systems for show corridors and open occasions, movie generation, live and recorded sound designing, sound recording, two-way radios, TV broadcasting, and in PCs for recording voice, speech recognition, VoIP, and for non-hearing reasons, for example, ultrasonic sensors or knock sensors. There are a few unique sorts of microphones which are utilized to employ distinctive ways to deal with change over the air pressure deviations of a sound wave to an electrical signal. The dynamic microphones are most regular as they uses a coil of wire suspended in an attractive field; the vibrating diaphragm is utilized by the condenser microphone as a capacitor plate, and the piezoelectric microphone, that uses a precious stone of piezoelectric material. Sound is an amazing thing. All of the different noises we hear are caused by minute pressure differences in the air around us. Amazing fact about it is that the air disseminates those pressure changes so well -- and so accurately -- over relatively long distances.

The First microphone was a metal diaphragm attached to a needle, and this needle scratched a pattern onto a piece of metal foil. The pressure in the air; is noticeable all around that happened when somebody talked toward the diaphragm moved the diaphragm, which moved the needle, which then recorded on the foil. At the point when the needle was later keep running back over the foil, the vibrations scratched on the foil would then move the diaphragm and re-make the sound. The fact that this purely mechanical system works shows how much energy the vibrations in the air can have. All the latest microphones are trying to achieve the equivalent thing as the original, but do it electronically rather than mechanically. A microphone wants to take varying pressure waves in the air and convert them into varying electrical signals. Table 11.5 describes various types of mics, including one of the first invented by Alexander Graham Bell.

Table 11.5: Types of Microphones

Microphone	Description
Liquid Microphones	It is imagined by Alexander Graham Bell and Thomas Watson, were among the principal working microphones to be designed, and they were an antecedent to what might later turn into the condenser microphones. The underlying fluid microphones designed utilizing a metal glass loaded with sulfuric acid and water. A diaphragm was set over the container with a needle on the getting side of the diaphragm. Sound waves would make the needle move in the water. A little electrical current raced to the needle, which was tweaked by sound vibrations. The fluid microphone was not at all a particular working gadget; however it shapes a huge analysis.
Carbon Microphones	It uses carbon dust technology which is used in the first telephones and is still used in some telephones today. The carbon dust has a thin metal or plastic diaphragm on one side. When sound waves hit the diaphragm, they shrink the carbon dust, which transforms its resistance. By running a current through the carbon, the changing resistance changes the amount of current that flows.
Fiber-Optic Microphone	Fiber-Optic Microphone uses super-thin strands of glass to disseminate message instead of traditional metal wires. Unlike conventional microphones, which are often big and transmit an electrical signal, fiber optic microphones can be extremely small, and they can be used in electrically sensitive environments. Also they can be produced with no metal, which makes them very

	fruitful in magnetic resonance imaging (MRI) applications and other situations where radio frequency interference is a problem.
Dynamic microphone	It takes edge of electromagnet effects. When a magnet moves past a wire (or coil of wire), the magnet activates current to discharge in the wire. Diaphragm in a dynamic microphone moves either a magnet or a coil when sound waves hit the diaphragm, and the movement generates a small current.
Electret microphones	They are the most widely used microphones in the consumer market as they're cheap and relatively simple. They are used in cell phones, computers and hands-free headsets. It is a type of condenser microphone where outside charge is replaced with an electret material which is known as permanent state of electric polarization.
Ribbon Microphone	It is a thin ribbon usually aluminum, duraluminum or nanofilm which is suspended in a magnetic field. Sound waves move the ribbon that changes the current flowing through it. They are bidirectional as they pick up sounds from both sides of the mic. The first ribbon microphone was RCA PB-31 which was produced in 1931, and brought the change in the audio and broadcasting industries as it set a new standard when it came to clarity.
Laser Microphone	It uses a laser beam to identify sound vibrations in a remote object. From window, the laser beam is directed into the room which reflects off the object and returns to a receiver (e.g. a solar panel) which converts the beam to an audio signal. The beam may also be bounced off the window itself. According to recent research, a new type of laser microphone which operates by streaming smoke across a laser beam which is intended at photocell, and then converted to an audio signal.
Condenser Microphone	Condenser microphone uses one plate of the capacitor which moves in response to sound waves. The movement changes the capacitance of the capacitor, and these changes are amplified to create a measurable signal. Usually, condenser microphones required a small power battery to deliver a voltage across the capacitor.
Cardioid Microphone	Cardioid microphone is used to record sound which is located in front of and on the sides of the microphone but not behind it. A polar plot of the gain for cardioid is heart-shaped (hence the name), with the highest sensitivity located directly in front of the mic, and slightly less on the sides. Cardioid microphones are suitable for recording live performances without capturing too much crowd noise. There are many

	handheld microphones used to amplify vocals are cardioid microphones.
Crystal Microphones	It uses a thin strip of piezoelectric material which is attached to a diaphragm. When the crystal is deflected by the diaphragm, the two sides of the crystal acquire opposite charges, which are proportional to the amount of deformation and disappear when the stress on the crystal disappears.

Check Your Progress

1) AMOLED stands for ____________ .

2) ________________________ is the process of adding geographical identification metadata to various media such as a geotagged photograph or video, websites, SMS messages, QR Codes or RSS feeds and is a form of geospatial metadata.

3) A _________ is a mobile phone which is able to capture photographs and often record video using one or more built-in digital cameras.

4) Most of the speakers have ___________.

5) A microphone, colloquially nicknamed mic or mike is a transducer that converts _____ into _______.

11.6 SUMMARY

An input/output (I/O) gadget is characterized as the equipment gadget which has the ability to take inputted, outputted or other prepared information. Additionally it can procure separate media information as info sent to smart phone or send mobile information to storage media as storage output. Input gadgets give input to a mobile, while output gadgets give a route to a mobile to output information for communication with users or other mobile. An I/O gadget is a gadget with both functionalities. Thus, I/O gadget information is bi-directional; such gadgets are typically arranged under storage or communications. Some of cases of I/O storage gadgets are MMC/SD Cards; USB streak drives and so forth. Thus, a portion of the cases of communication I/O gadgets are network connectors, Bluetooth connectors/dongles and modems.

11.7 SOLUTIONS/ANSWERS

1) Active Matrix Organic Light Emitting Diode
2) Geotagging
3) Camera phone
4) Built in amplifier
5) Sound, electric signal

11.8 FURTHER READINGS

- http://www.bestartech.com/base_mount.html
- http://www.voicecouncil.com/vcs-brief-guide-mic-pick-up-patterns/
- https://www.howstuffworks.com/
- https://en.wikipedia.org/wiki/Geotagging
- https://en.wikipedia.org/wiki/Camera
- https://en.wikipedia.org/wiki/Camera_phone
- https://en.wikipedia.org/wiki/Computer_speakers
- https://en.wikipedia.org/wiki/Microphone
- https://en.wikipedia.org/wiki/Display
- https://en.wikipedia.org/wiki/Display_device

ignou
THE PEOPLE'S
UNIVERSITY
Indira Gandhi National Open University
School of Computer and Information Sciences

C·O·L
Commonwealth of Learning

INTRODUCTION TO MOBILE ARCHITECTURE

Software Development Tools

4

School of Computer and Information Sciences
Indira Gandhi National Open University

Introduction to Mobile Architecture

Commonwealth *of* Learning

Block

4

SOFTWARE DEVELOPMENT TOOLS

INTRODUCTION TO MOBILE ARCHITECTURE

This course has been developed as part of the collaborative OER Course Development for ICT Skills initiative of the Commonwealth of Learning (COL). COL is an intergovernmental organisation created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

Indira Gandhi National Open University (IGNOU), established by an Act of Parliament of India in 1985, has continuously striven to build an inclusive knowledge society through inclusive education. It has tried to increase the Gross Enrollment Ratio (GER) by offering high-quality teaching through the Open and Distance Learning (ODL) mode.

For the avoidance of doubt, by applying this license the Commonwealth of Learning does not waive any privileges or immunities from claims that it may be entitled to assert, nor does the Commonwealth of Learning submit itself to the jurisdiction, courts, legal processes or laws of any jurisdiction. The ideas and opinions expressed in this publication are those of the author/s; they are not necessarily those of *Commonwealth of Learning* and do not commit the organisation.

Course Team

Author:

Mr. Shailesh Kumar Shiva Kumar, Research Scholar (SOCIS, IGNOU) and Senior Technology Architect (Infosys Limited) , Bangalore, India

Content Editor: Dr.P.Venkata Suresh, SOCIS, IGNOU, New Delhi, India

Content Design Team

Dr. P.V.Suresh,
Associate Professor & Director,
SOCIS, IGNOU, New Delhi

Mr. Amit Kumar Saxena,
Chief Technology Officer,
Grasp IO Inc, Bengaluru

Dr. S. Arumuga Perumal,
Associate Professor & Head of the
Department, S.T. Hindu College,
Nagercoil, Tamil Nadu

Prof.Parma Nand Astya,
Dean, School of Computing Science &
Engineering, Galgotias University,
Greater Noida

Mr. Shashi Bhushan Sharma,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. Akshay Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Dr. Naveen Kumar,
Associate Professor, SOCIS,
IGNOU, New Delhi

Mr. M. P. Mishra,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Dr. Sudhansh Sharma,
Assistant Professor, SOCIS,
IGNOU, New Delhi

Language Editors: Mr. P. Lakshmi Kantham , Faculty in English(Retd.), VSR & NVR College, Tenali, India

CRC Preparation: Tessa Media & Computers, New Delhi

Published by:

Indira Gandhi National Open University, Maidan Garhi, New Delhi-110068, India.

The course has been developed with the support of the Commonwealth of Learning, Canada.

Acknowledgements

Indira Gandhi National Open University wishes to thank those below for their contribution to development of this Course Material:

Staff of School of Computer and Information Sciences, Administration, Finance & Accounts Division and Electronic Media Production Centre, IGNOU, New Delhi, India and all External Experts. Special thanks to Dr.Sanjaya Mishra, Education Specialist(eLearning), Commonwealth of Learning, Canada for his support and guidance through out.

BLOCK 4 INTRODUCTION

This is the fourth and final block of the Course. This block introduces the learner to tools that can be used to develop software for mobile apps.

The process of software development has been rapidly changing since years. Initially, programmers used to write code. Then, we had 4GLs which can be used to run queries which mean that underlying code was already available. Also, we had tools which can be used to generate code if we drag and drop the elements that are part of our application.

With the advent of smart phones, several tools were developed for professionals to write apps. There are different types of tools. Some tools are developed by the developers of the concerned operating system. Then, there are tools that are developed by third party vendors that can be used to develop mobile apps for a particular mobile operating system.

In this block, our focus will be on the tools that can be used to develop software for mobile operating systems, namely, iOS, Android and Windows. Some tools were native to the operating system. Some are platform independent.

This block consists of four units and is organized as follows:

Unit 12 deals native development tools. It covers Android studio and Eclipse IDE for Android, Xcode and Swift for iOS, and C# and XAML for Windows.

Unit 13 deals with cross platform mobile app development tools. It covers Xamarin and PhoneGap.

Unit 14 deals with app publishing tools and Developer programs. It covers Google’s Play Store, Apple’s App Store and Window’s Store.

Unit 15 deals with monetization. It deals with monetization of Android apps, iOS apps and Windows based mobile apps.

UNIT 12 NATIVE DEVELOPMENT TOOLS

Structure

12.0 INTRODUCTION

Developing mobile apps needs a good supporting eco system. Development tools and Integrated Development Environment (IDE) form an integral part of the support eco system. Development tools and IDEs provide all necessary services, editors, resources to quickly and efficiently develop the mobile app. They support the full lifecycle of the mobile app starting from development, design, testing, debugging and deployment.

In this unit, we will discuss various development tools for key mobile platforms. The learner needs to note that the system requirements etc. may change from time to time with different releases and hence needs to follow the instructions given on the portals while downloading and installing various software mentioned in this unit. Also, in the case of programming , latest syntax will apply.

12.1 OBJECTIVES

After going through this unit, you should be able to understand

- key concepts of Android Studio and Eclipse IDE for Android;
- migration from Eclipse IDE to Android Studio;
- concepts of Xcode IDE and Swift programming language for iOS apps development; and
- C# and Extensible Application Markup Language (XAML) for Windows mobile.

12.2 DEVELOPMENT TOOLS FOR ANDROID

In this section we will look at Android Studio and Eclipse IDE which can be used for development of Android based mobile apps.

12.2.1 Android Studio

Google provides Android studio for developing Android based mobile apps. Android studio uses Gradle build system. The following are the main features of the Android Studio:

- Build , debug and release variants are available.
- Code templates are provided.
- Drag and drop themes are available.
- Lint tool for performance optimization purposes.
- User interface design to preview layout for multiple screens.
- Provides data binding with model entities and UI elements.
- Android studio provides source control integration with many systems such as GIT, Apache Subversion (SVN), and Concurrent Versions System (CVS).
- Provides built-in cloud messaging.

Android monitor can be used along with Android Studio for performance monitoring and testing.

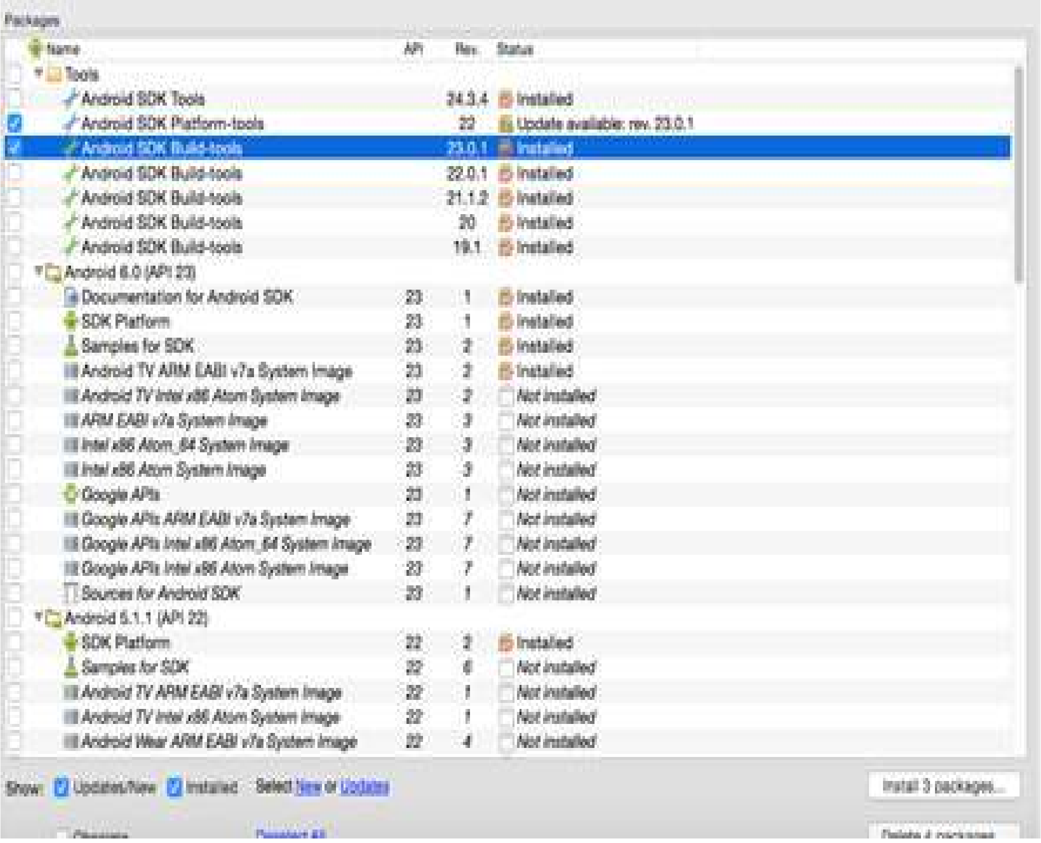
The following are various steps in setting up and running Android Studio:

a) Setting up Android Studio

Android Studio setup needs 4GB of RAM, 500GB disk space with JDK 7 or above. Detailed installation steps are given below:

i) Install Android studio from https://developer.android.com/sdk/index.html

ii) Update the SDK bundle using the Android SDK manager (Figure 12.1).



Packages

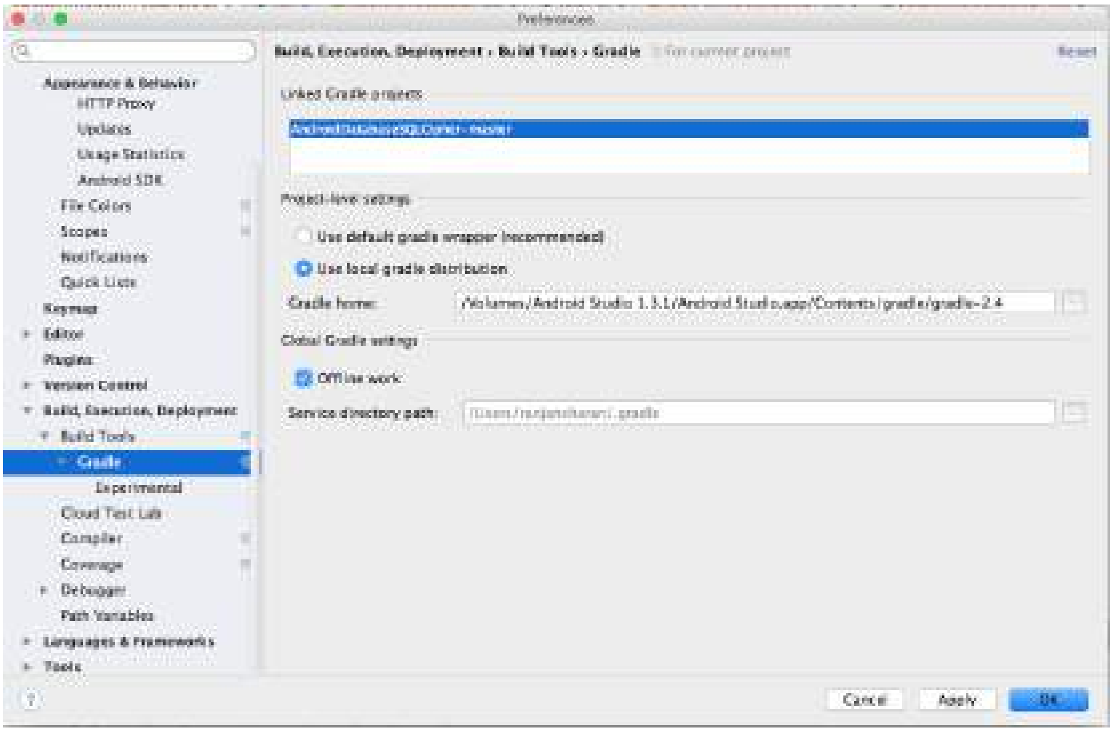
	Name	API	Rev.	Status
□	Tools			
□	Android SDK Tools		24.3.4	Installed
☑	Android SDK Platform-tools		22	Update available: rev. 23.0.1
☑	Android SDK Build-tools		23.0.1	Installed
□	Android SDK Build-tools		22.0.1	Installed
□	Android SDK Build-tools		21.1.2	Installed
□	Android SDK Build-tools		20	Installed
□	Android SDK Build-tools		19.1	Installed
□	Android 6.0 (API 23)			
□	Documentation for Android SDK	23	1	Installed
□	SDK Platform	23	1	Installed
□	Samples for SDK	23	2	Installed
□	Android TV ARM EABI v7a System Image	23	2	Installed
□	Android TV Intel x86 Atom System Image	23	2	Not installed
□	ARM EABI v7a System Image	23	3	Not installed
□	Intel x86 Atom_64 System Image	23	3	Not installed
□	Intel x86 Atom System Image	23	3	Not installed
□	Google APIs	23	1	Not installed
□	Google APIs ARM EABI v7a System Image	23	7	Not installed
□	Google APIs Intel x86 Atom_64 System Image	23	7	Not installed
□	Google APIs Intel x86 Atom System Image	23	7	Not installed
□	Sources for Android SDK	23	1	Not installed
□	Android 5.1.1 (API 22)			
□	SDK Platform	22	2	Installed
□	Samples for SDK	22	6	Not installed
□	Android TV ARM EABI v7a System Image	22	1	Not installed
□	Android TV Intel x86 Atom System Image	22	1	Not installed
□	Android Wear ARM EABI v7a System Image	22	4	Not installed

Show: ☑ Updates/New ☑ Installed Select New or Updates

□ Obsolete

Figure 12.1 : Android SDK manager

iii) Download Gradle from https://gradle.org/gradle-download/ and configure Gradle in Android Studio at Android Studio → Preferences → Build, Execution, Deployment → Build tools → Gradle → select Use local gradle distribution → Gradle home → <Path to gradle file> (Figure 12.2).



Build, Execution, Deployment › Build Tools › Gradle

Linked Gradle projects

Project-level settings

○ Use default gradle wrapper (recommended)

● Use local gradle distribution

Gradle home: /Volumes/Android Studio 1.3.1/Android Studio.app/Contents/gradle/gradle-2.4

Global Gradle settings

☑ Offline work

Service directory path: /Users/[illegible]/.gradle

Figure 12.2 : Gradle in Android Studio

b) Running the Hello World project in Android Studio.

Let us look at steps for building the first app using Android studio. Before we start we need to check if JDK 7 is installed and if JDK 7 and Android Studio are configured. The following are the steps and screenshots for creating the first app mentioned above:

i) In Android Studio, go to File → New Project and create the application (Figure 12.3).

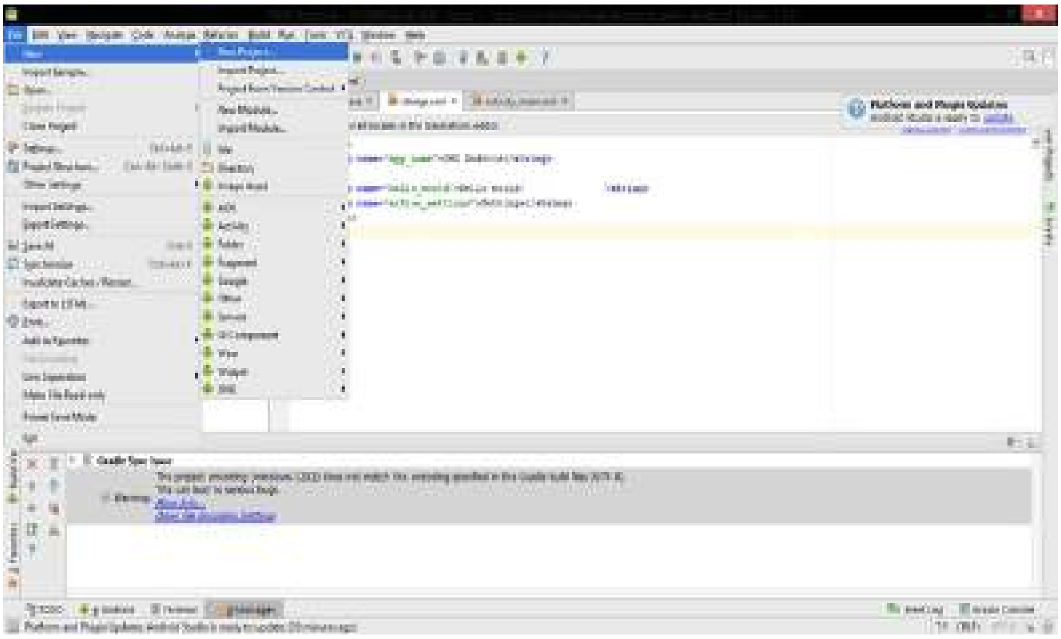
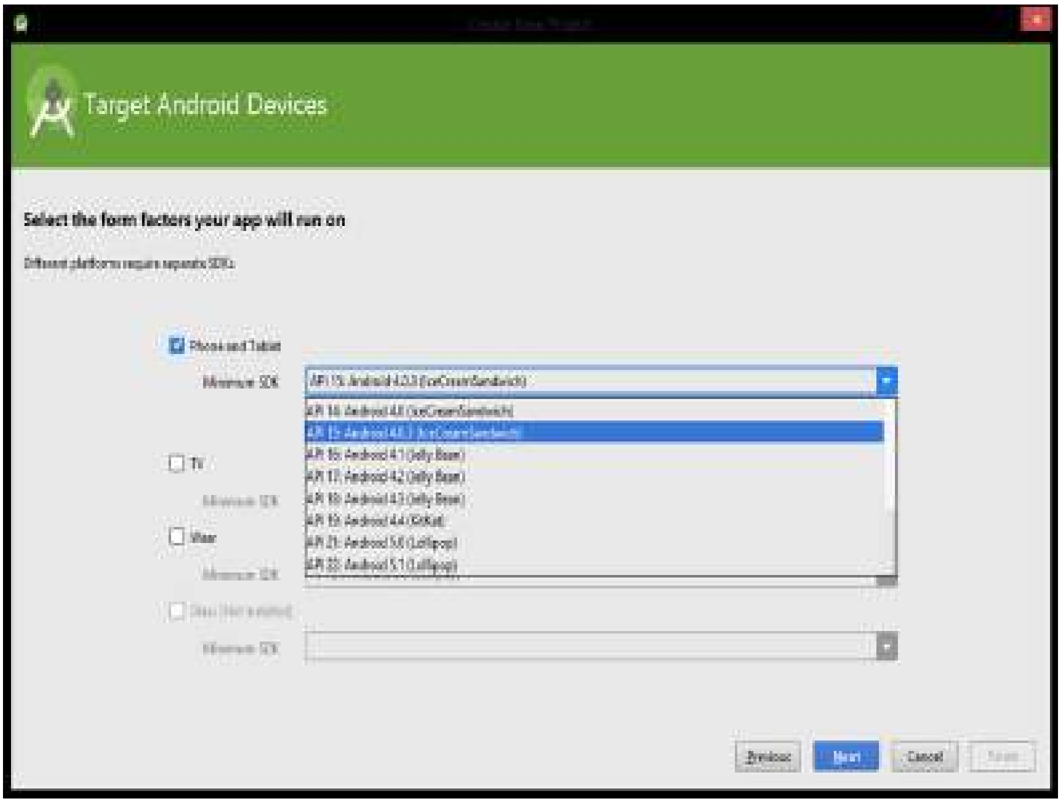


Figure 12.3 : Creating a new project in Android Studio

ii) Select the device and environment (Figure 12.4).



Target Android Devices

Select the form factors your app will run on

Different platforms require separate SDKs

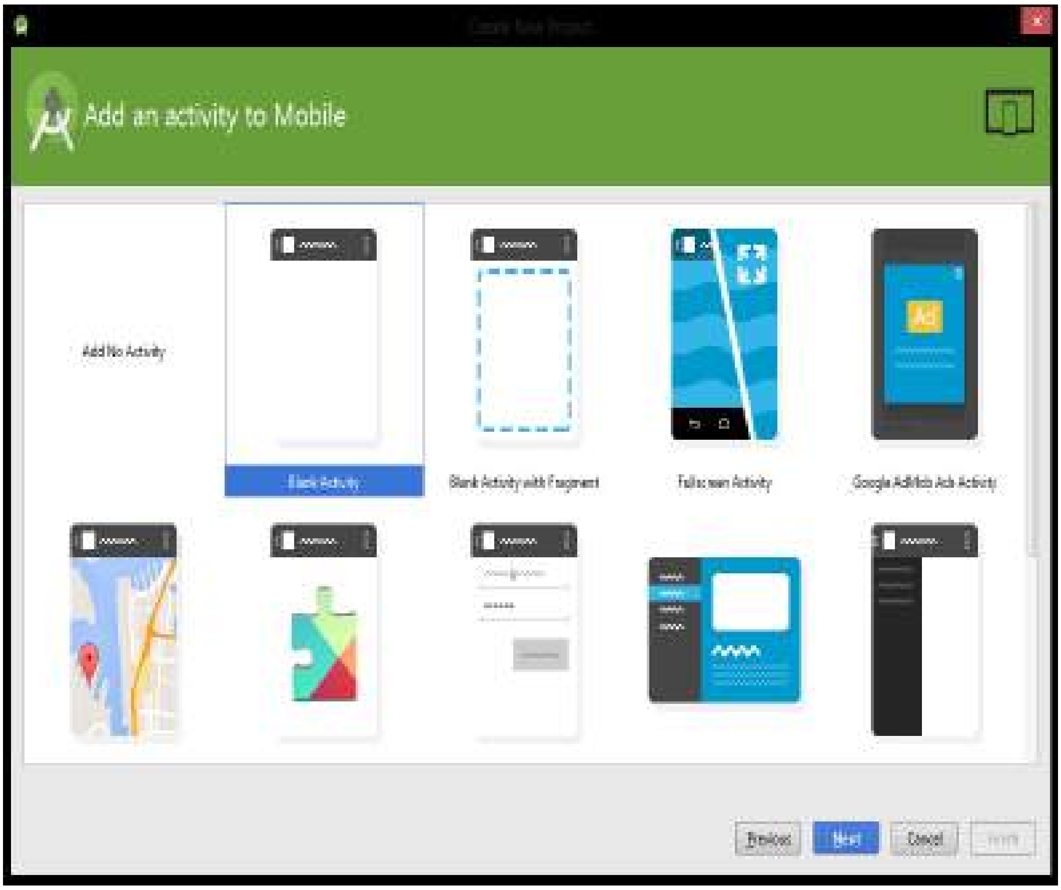
☑ Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

API 14: Android 4.0 (IceCreamSandwich)
API 15: Android 4.0.3 (IceCreamSandwich)
API 16: Android 4.1 (Jelly Bean)
API 17: Android 4.2 (Jelly Bean)
API 18: Android 4.3 (Jelly Bean)
API 19: Android 4.4 (KitKat)
API 21: Android 5.0 (Lollipop)
API 22: Android 5.1 (Lollipop)
□ TV
Minimum SDK
□ Wear
Minimum SDK
□ Glass (Not Installed)
Minimum SDK
Previous
Next
Cancel
Finish

Figure 12.4 : Device selection in Android Studio

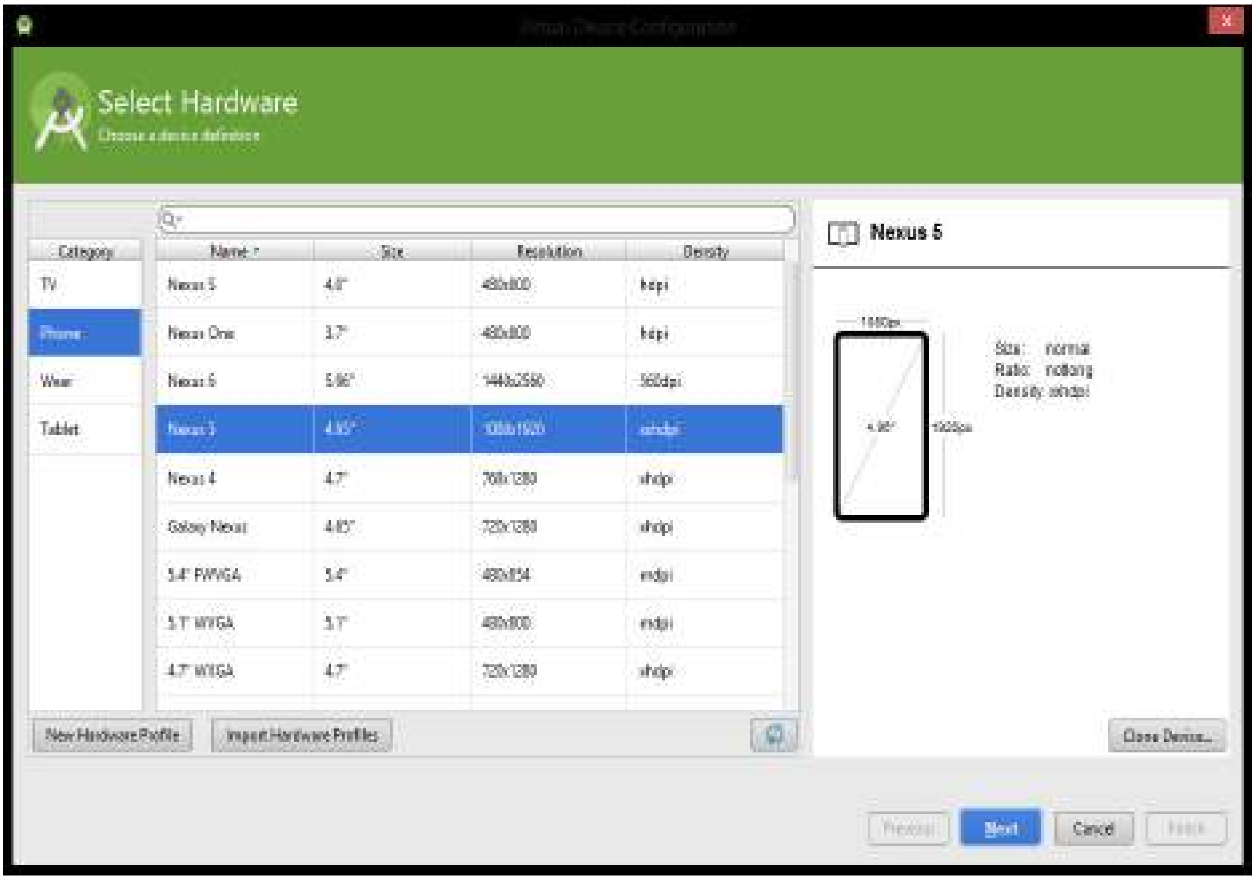
iii) Select the activity and rename it as required (Figure 12.5).



Add an activity to Mobile

Figure 12.5 : Adding activities in selected device

iv) Build the project and select the hardware device in Android Virtual Device (AVD) manager (Figure 12.6).



Select Hardware

Choose a device definition

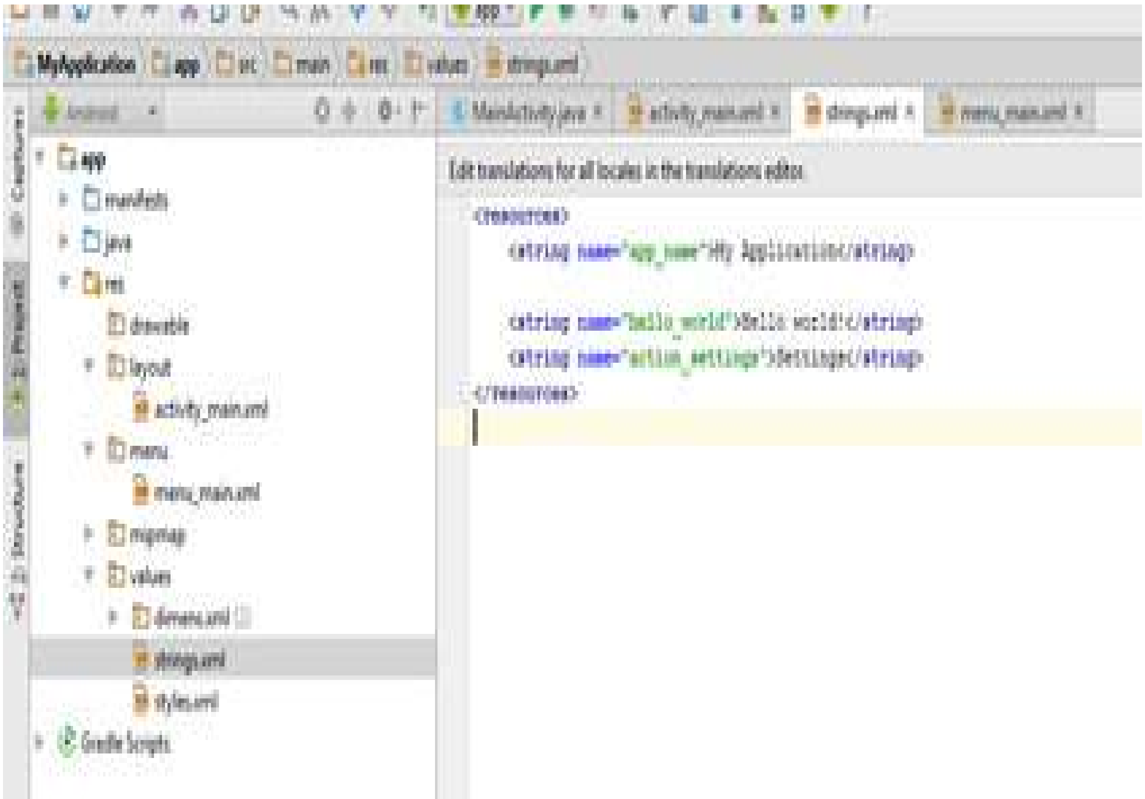
Category	Name	Size	Resolution	Density
TV	Nexus S	4.0"	480x800	hdpi
Phone	Nexus One	3.7"	480x800	hdpi
Wear	Nexus 6	5.96"	1440x2560	560dpi
Tablet	Nexus 5	4.95"	1080x1920	xxhdpi
	Nexus 4	4.7"	768x1280	xhdpi
	Galaxy Nexus	4.65"	720x1280	xhdpi
	5.4" FWVGA	5.4"	480x854	mdpi
	3.7" WVGA	3.7"	480x800	mdpi
	4.7" WXGA	4.7"	720x1280	xhdpi

Nexus 5

Size: normal
Ratio: notlong
Density: xxhdpi

Figure 12.6 : Hardware selection

v) Edit the strings.xml as needed (Figure 12.7).



Edit translations for all locales in the translations editor.

```
<resources>
    <string name="app_name">My Application</string>

    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>
</resources>
```

Figure 12.7 : Editor in Android studio

vi) Run the application on the emulator (Figure 12.8).

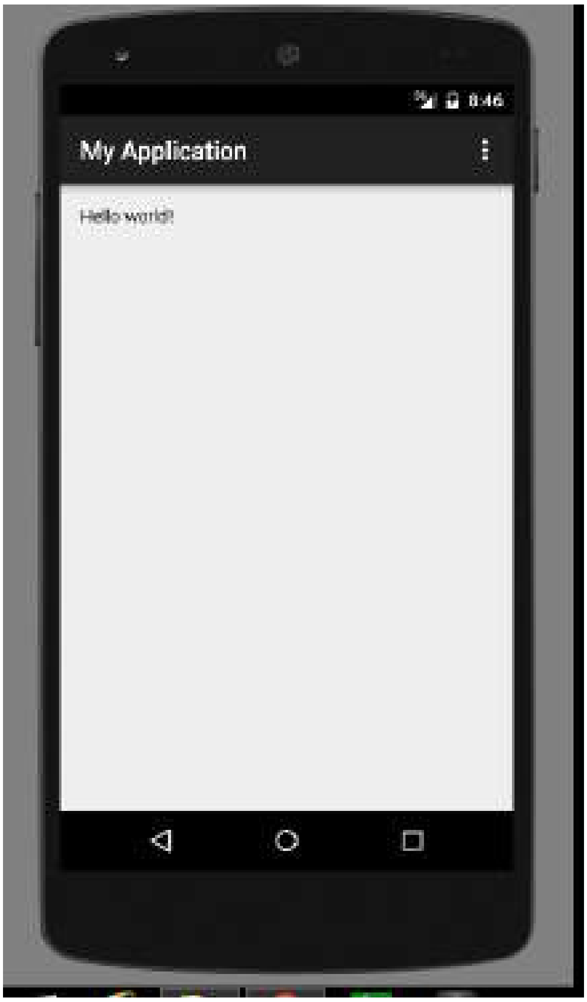


Figure 12.8 : Output in Emulator

12.2.2 Eclipse IDE

Eclipse is the integrated development environment (IDE) for developing Java based applications. We can build, debug, deploy and test the enterprise Java and core Java applications. Eclipse studio can also be used to develop the Android app by using Android SDK, Android development tools plug-in and Android virtual device.

The following is the process of setting up of Eclipse IDE:

a) Download the Eclipse IDE from http://www.eclipse.org/download

b) Select Eclipse IDE for Java EE Developers and select the appropriate OS.

c) Install the appropriate JRE and JDK needed and configure it in Eclipse IDE.

d) For installing the updates and plug-ins, we can use the menu Help → Software Updates. Design Studio can be installed from the Update option.

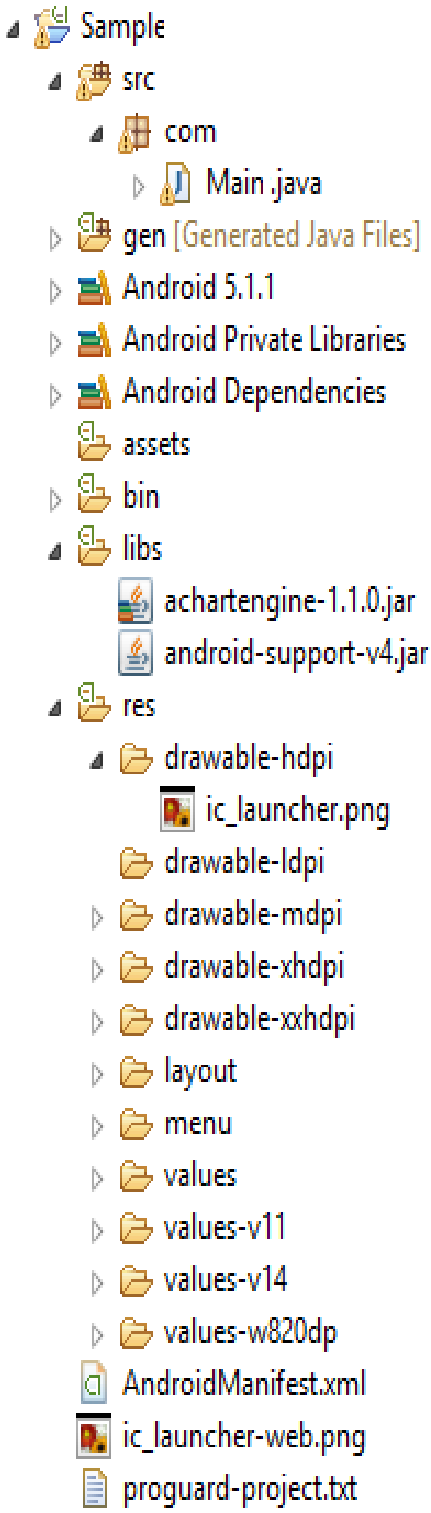
Migration from Eclipse to Android Studio

In this section, we will discuss migration from Eclipse to Android Studio. Table 12.1 provides the mapping of components in both Eclipse IDE and Android Studio:

Table 12.1 : Mapping of components for Eclipse IDE and Android studio

Eclipse IDE	Android Studio
Project	Module
Workspace	Project
Library project	Module

A sample Eclipse project is shown in Eclipse Project Explorer in Figure 12.9.



```
Sample
  src
    com
      Main.java
  gen [Generated Java Files]
  Android 5.1.1
  Android Private Libraries
  Android Dependencies
  assets
  bin
  libs
    achartengine-1.1.0.jar
    android-support-v4.jar
  res
    drawable-hdpi
      ic_launcher.png
    drawable-ldpi
    drawable-mdpi
    drawable-xhdpi
    drawable-xxhdpi
    layout
    menu
    values
    values-v11
    values-v14
    values-w820dp
  AndroidManifest.xml
  ic_launcher-web.png
  proguard-project.txt
```

Figure 12.9: Eclipse Project Explorer

The corresponding Android Studio project is shown in figure 12.10.

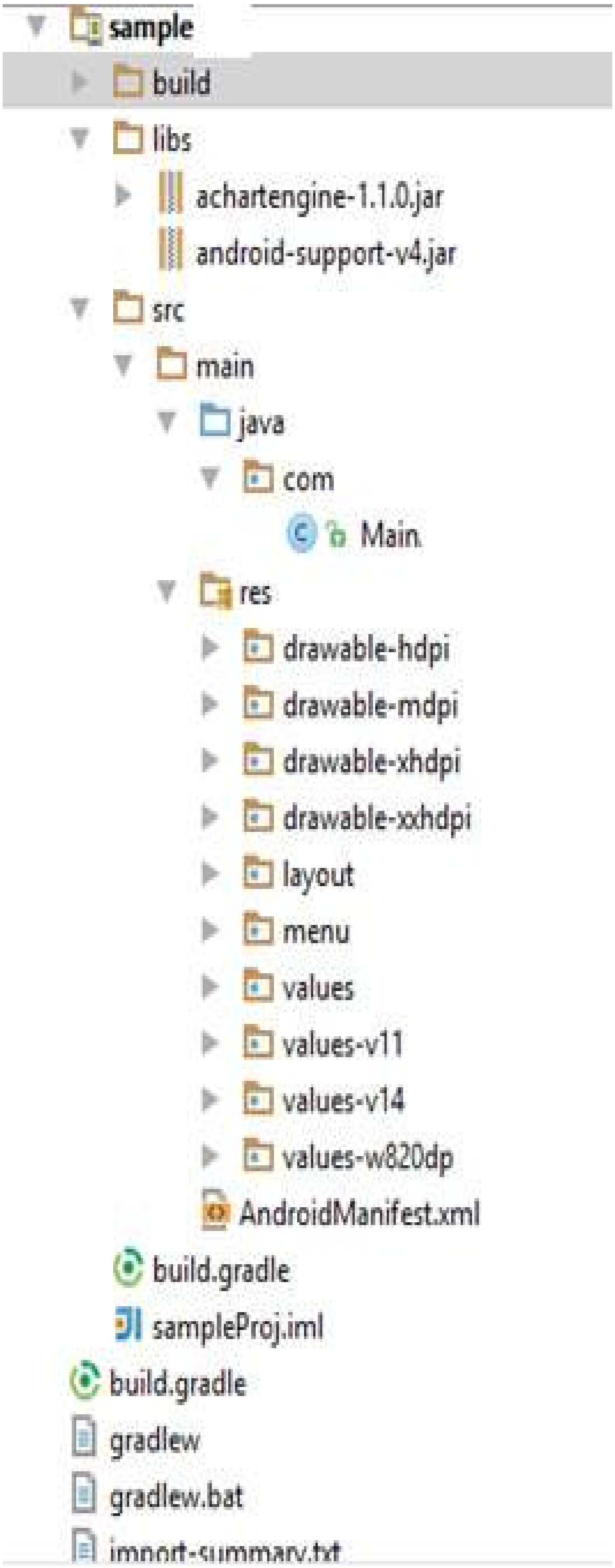
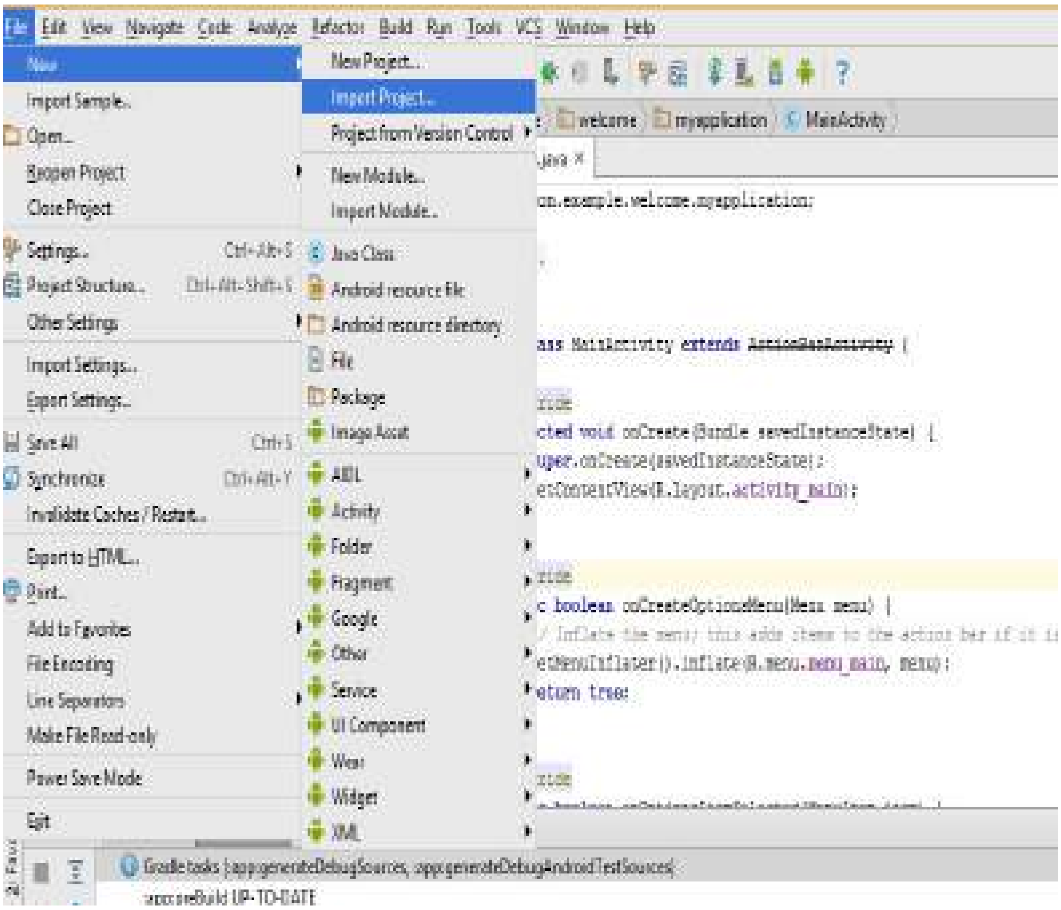


Figure 12.10: Android Studio Project

The following are the steps to migrate from Eclipse project to Android Studio project:



```
om.example.welcome.myapplication;

ass MainActivity extends ActionBarActivity {

ride
cted void onCreate(Bundle savedInstanceState) {
uper.onCreate(savedInstanceState);
etContentView(R.layout.activity_main);

ride
c boolean onCreateOptionsMenu(Menu menu) {
/ Inflate the menu; this adds items to the action bar if it i
etMenuInflater().inflate(R.menu.menu_main, menu);
eturn true;

ride
```

Gradle tasks [:app:generateDebugSources, :app:generateDebugAndroidTestSources]

:app:preBuild UP-TO-DATE

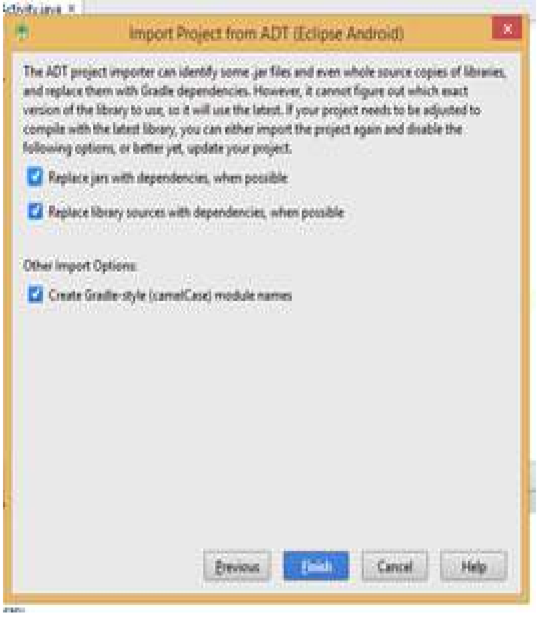
Figure 12.11: Opening a new Project in Android Studio

1) In Android Studio go to File → New → Import (Figure 12.11)

2) Provide Eclipse project path

3) Give target file path

4) Check “Replace jars” and “Replace library sources” (Figure 12.12)



The ADT project importer can identify some jar files and even whole source copies of libraries, and replace them with Gradle dependencies. However, it cannot figure out which exact version of the library to use, so it will use the latest. If your project needs to be adjusted to compile with the latest library, you can either import the project again and disable the following options, or better yet, update your project.

☑ Replace jars with dependencies, when possible

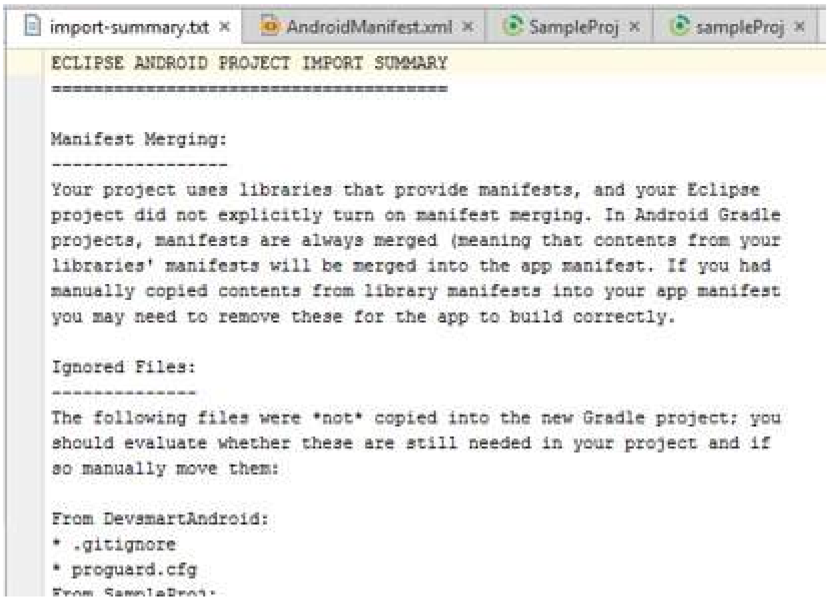
☑ Replace library sources with dependencies, when possible

Other Import Options:

☑ Create Gradle-style (camelCase) module names

Figure 12.12 : Replace jars and Replace Library Sources

5) Figure 12.13 summarizes import from Eclipse to Android Studio :

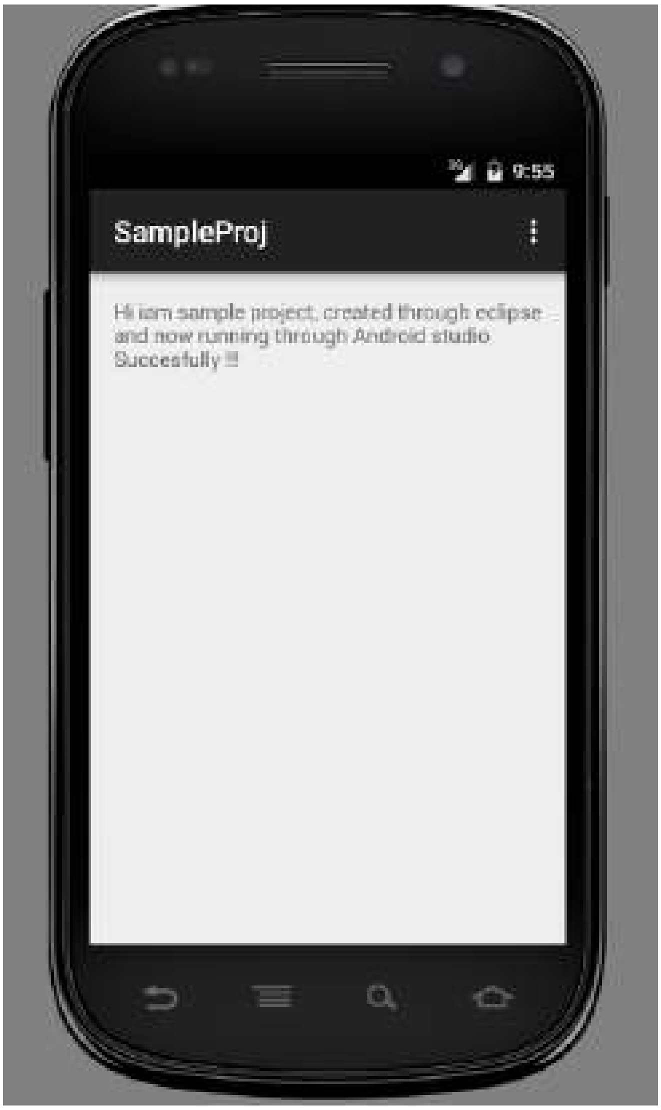


```
ECLIPSE ANDROID PROJECT IMPORT SUMMARY
======================================

Manifest Merging:
-----------------
Your project uses libraries that provide manifests, and your Eclipse
project did not explicitly turn on manifest merging. In Android Gradle
projects, manifests are always merged (meaning that contents from your
libraries' manifests will be merged into the app manifest. If you had
manually copied contents from library manifests into your app manifest
you may need to remove these for the app to build correctly.

Ignored Files:
--------------
The following files were *not* copied into the new Gradle project; you
should evaluate whether these are still needed in your project and if
so manually move them:

From DevsmartAndroid:
* .gitignore
* proguard.cfg
From SampleProj:
```

Figure 12.13: Import summary from Eclipse to Android Studio

6) Once the project is setup , we can add JARs and library project using context menus.

7) Once we run the project, the emulator output is shown in Figure 12.14



Hi iam sample project, created through eclipse and now running through Android studio Succesfully !!!

Figure 12.14 : Output in Emulator

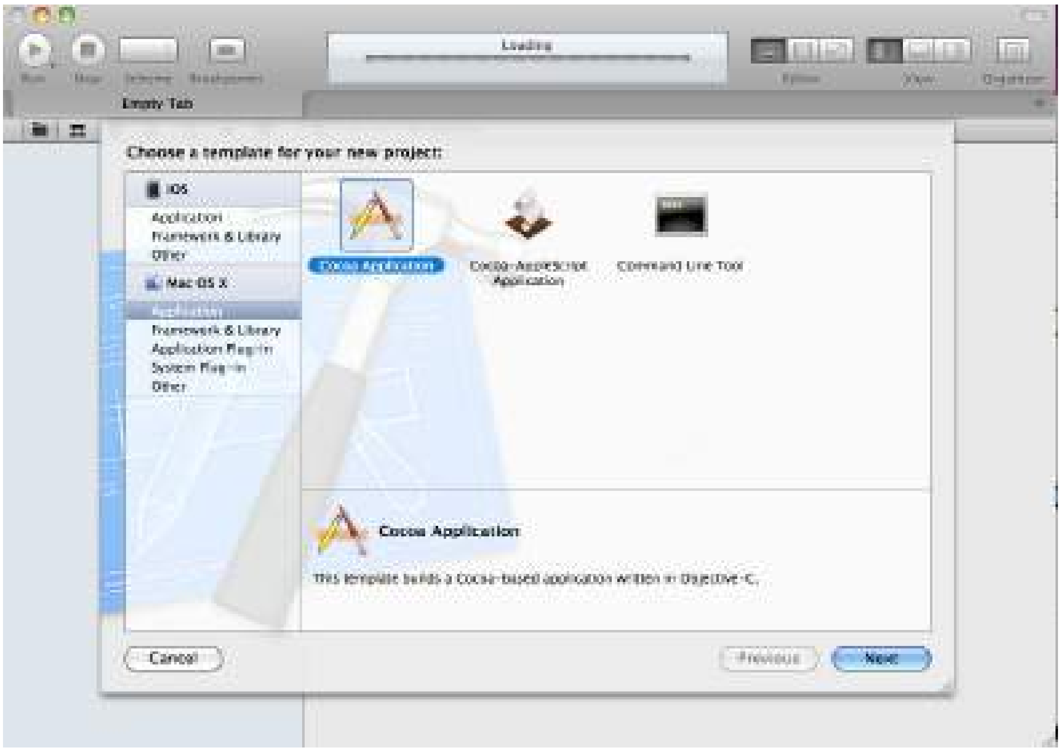
12.3 DEVELOPMENT TOOLS FOR iOS

In this section, we will look at various tools and IDEs needed for development in Apple iOS. Xcode IDE and Swift can be used to develop apps for Apple devices such as iPhone, iPad, Apple Watch, Apple TV, Mac etc.

12.3.1 Xcode

Xcode IDE consists of a set of tools to develop apps for Apple's iOS mobile platform (Figure 12.15). The key features of Xcode are as follows:

- Xcode provides a single interface for all workflow and it is fully integrated with interface builder.
- Xcode provides large number of utility tools for memory leak profiling, allocation management, automation, code signing, code documentation, interface builder etc.
- User interface can be designed by reusing UI objects in the dock.
- It consists of version editor to connect to source control systems such as Git, SVN etc.
- It provides Xcode debugger for code debugging
- Can be used for monitoring and control of the program execution using breakpoints, watch points
- Fix-it can be used for error highlighting and for fixing code errors in real time.
- Xcode tool kit offers UI automation and accessibility testing.
- Xcode also acts as a source management system providing features such as snapshot creation.
- Xcode provides option to save code as a template and provides re-usable code snippet library.



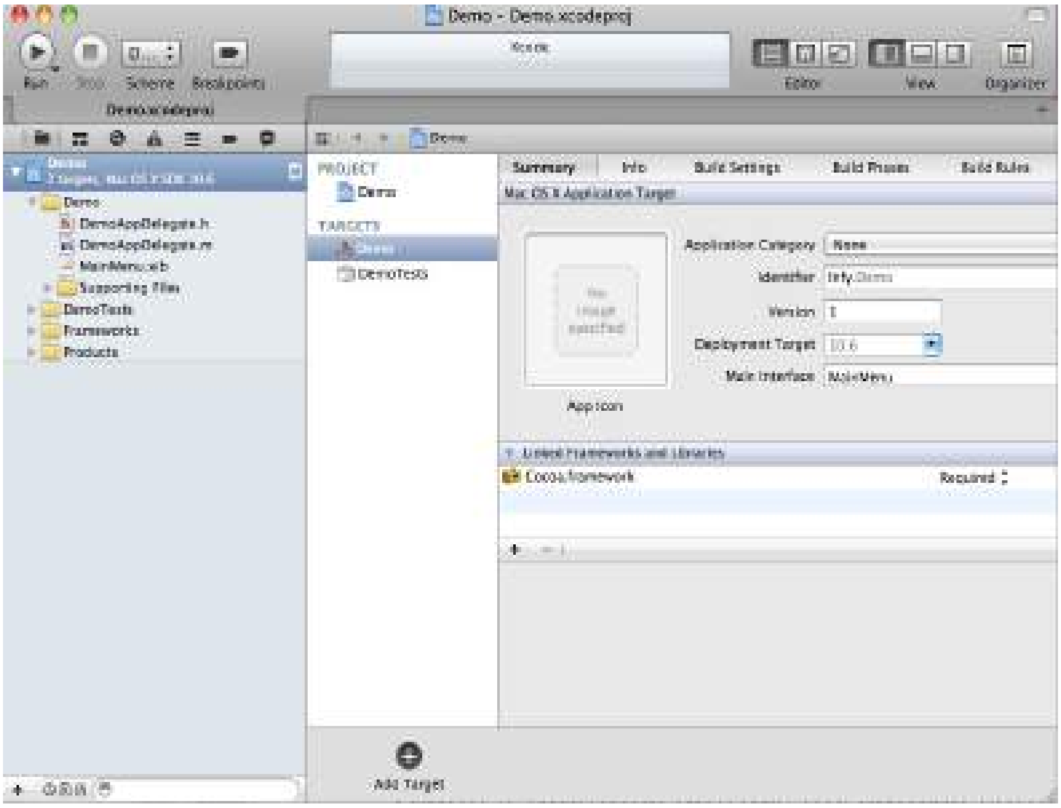
Choose a template for your new project:

Cocoa Application

This template builds a Cocoa-based application written in Objective-C.

Figure 12.15 : Xcode IDE

The project navigator view of Xcode is shown in figure 12.16.



Application Category: None

Identifier: info.Demo

Version: 1

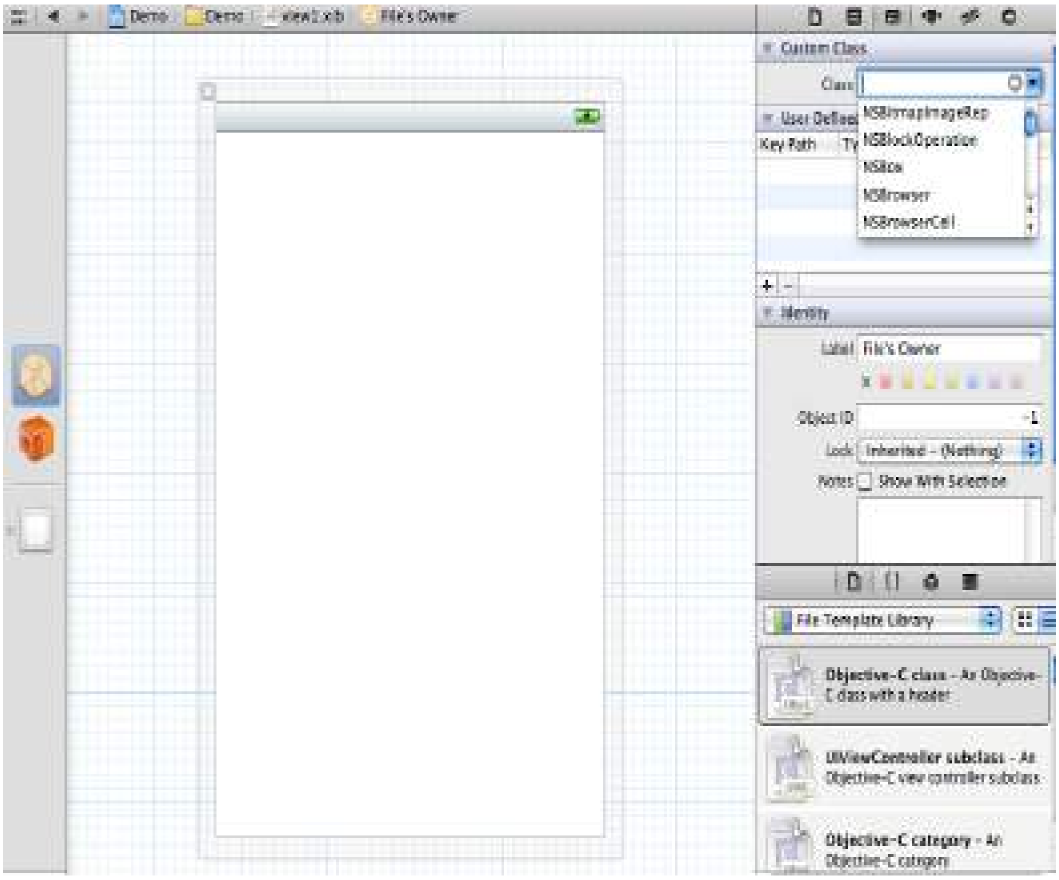
Deployment Target: 10.6

Main Interface: MainMenu

Cocoa.framework Required

Figure 12.16 : Project navigator of Xcode

Interface builder which can be used for UI designing is shown in figure 12.17. Its possible to make direct connections to source code from interface builder.



Label: File's Owner

Object ID: -1

Lock: Inherited - (Nothing)

Notes: □ Show With Selection

Demo
Demo
view1.xib
File's Owner
Custom Class
Class
NSBitmapImageRep
NSBlockOperation
NSBox
NSBrowser
NSBrowserCell
User Defined
Key Path
Type
Identity
File Template Library
Objective-C class - An Objective-C class with a header
UIViewController subclass - An Objective-C view controller subclass
Objective-C category - An Objective-C category

Figure 12.17: Interface Builder

12.3.2 Swift

Swift is the programming language that is supported from iOS 8 onwards and it succeeds objective- C used for earlier Apple development. Swift has a cleaner syntax and is designed to work alongside Objective – C. To use Swift, we need at least Xcode 6 and we we can use Apple's *Playgrounds* feature to try out Swift.

Swift uses Model-view-controller (MVC) programming architecture and uses Apple's Cocoa and CocoaTouch frameworks. Low level virtual machine is used as Swift compiler.

Some of the key advantages of Swift are as follows:

- Enhanced readability: It replaces "@" and "NS" in Objective-C and uses a "." Notation.
- Enhanced maintainability: Swift can be easily maintained as a single file.
- Enhanced safety due to better error handling.
- Enhanced memory management due to automatic reference counting.
- Better performance
- Namespace feature to avoid name collision
- Interoperability with Objective C

The following is a high level description of programming elements of Swift:

Variables

Examples of Swift variables are given below:

```
var myVar:Int8 = 1
var testString = "Hello World"
let someConstant = 12.2
```

We could use "+" operator for string concatenation as well

Arrays

We can define arrays as follows:

```
var noArray = [12,23,2,4,5]
var strArray = ["Ab","Cd","Ef","Gh"]
```

Dictionaries

Swift allows us to save key value pairs in dictionaries. For example,

```
var testDictionary : Dictionary<String,String> = ["1" : "One", "2" : "Two", "3" : "Three"]
```

We can use key as an index to retrieve the value from dictionary

```
var value = testDictionary["1"]
```

In this case, "value" would be "One".

Functions

"Func" can be used to define functions

```
func helloWorld () {
```

```
println("Hello World !")
}
```

Classes

Class keyword can be used to define classes

```
class PersonDetails {

var firstName : String
var lastName : String
var age : Int = 5

init() {
firstName = "John"
lastName = "Smith"
}

func printPersonDetails() {

println("FirstName: \(firstName)")
println("LastName: \(lastName)")
println("Age: \(age)")

}
}
```

☞ **Check Your Progress 1**

1) _______ can be used along with Android Studio for performance monitoring
2) Android Studio uses _________ build system
3) _____ file contains the key text values in Android app
4) A project in Eclipse IDE would become ______ in Android studio
5) ________ can be used for error highlighting and for fixing code errors in Xcode
6) _____ can be used for UI designing in Xcode
7) ___ qualifier is used in Swift to define constants
8) _____ feature is used in Swift to define key value pairs

12.4 DEVELOPMENT TOOLS FOR WINDOWS BASED MOBILES

Apps for Windows based mobiles apps can be developed using two methods: C++/C#/VB with XAML or by using JavaScript, HTML 5 and CSS3.

12.4.1 C#

C# (pronounced as C Sharp) is a .NET based Object Oriented programming language used for Microsoft mobile app development. Microsoft Visual Studio is the main tool used for development of C# applications.

A simple Hello World program in C# is given below.

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace HelloWorld
{
   class Program
   {
      static void Main(string[] args)
      {

      Console.WriteLine("Hello World !");

      }
   }
}
```

Some of the key features of C# are as follows:

- Namespaces: They create easily organizable and reusable compilation units in C#.
- Classes are defined by keyword “class”.
- Variables can be declared using <datatype identifier> syntax. For example *public string pname* declares “pname” as a string type variable.
- Access modifiers include public, private, protected, internal and protected internal.
- Methods form the building block of the C# code.
- Exception handling can be done using try-catch blocks.

12.4.2 XAML

XAML is mainly used for page designs. It provides the flexibility to assign properties for an object. XAML with C# is used when we need dynamic behavior for controls such as focus, hover and when data binding and context setting is required for the app. XAML can also be used to control device services such as GPS, audio, GPRS, Gravity sensor etc.

The developer can set the look and feel properties for the object in XAML. Main properties that can be represented in XAML are as follows:

- **x:Key** uniquely identifies elements that are created and referenced as resources which exist within a **ResourceDictionary**
- Style.TargetType : Gets or sets the type for which this style is intended.
- Setter: This is a class that applies a property value

A resource that can be referenced in XAML can either be static or dynamic. Differences between the static and dynamic resources are given in table 12.2.

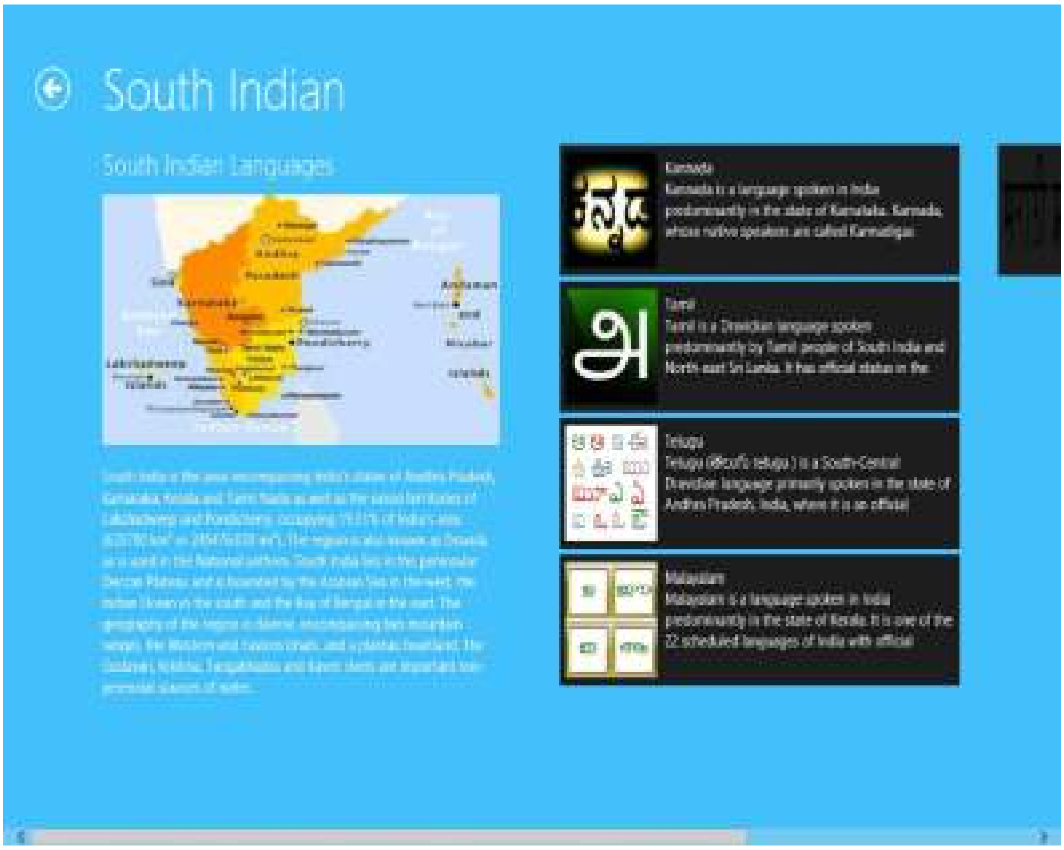
Table 12.2 : Static vs Dynamic Resources

Static Resource	Dynamic Resource
Load time resolution of key wherein the reference is resolved and replaced at load time	Compile time key resolution wherein the reference is resolved at compile time but replaced at run time.
Forward reference not supported	Forward reference supported
Can be used when resources are predefined and not dependent on user input	Can be used when the proper is dependent on the user.

To create an app using C# and XAML we need the following components:

- XAML component which defines the page layout, page controls, visual design, controls nesting and such.
- Associated back end page code that takes care of data binding and handles user responses.

Microsoft Visual Studio helps in rapid application development for the Windows apps by providing various features such as IntelliSense, end-to-end packaging and deployment, and debugging support. Figure 12.18 shows a screenshot of "Desipad" Windows app (https://www.microsoft.com/en-us/store/p/desipad/9wzdncrdccgd) designed using Metro UI developed using MS Visual Studio. The app uses Hub and Grid Windows Store template.



South India is the area encompassing India's states of Andhra Pradesh, Karnataka, Kerala and Tamil Nadu as well as the union territories of Lakshadweep and Pondicherry, occupying 19.31% of India's area (635,780 km² or 245,480 sq mi). The region is also known as Dravida as is used in the National anthem. South India lies in the peninsular Deccan Plateau and is bounded by the Arabian Sea in the west, the Indian Ocean in the south and the Bay of Bengal in the east. The geography of the region is diverse, encompassing two mountain ranges, the Western and Eastern Ghats, and a plateau heartland. The Godavari, Krishna, Tungabhadra and Kaveri rivers are important non-perennial sources of water.

Kannada
Kannada is a language spoken in India predominantly in the state of Karnataka. Kannada, whose native speakers are called Kannadigas

Tamil
Tamil is a Dravidian language spoken predominantly by Tamil people of South India and North-east Sri Lanka. It has official status in the

Telugu
Telugu (తెలుగు telugu) is a South-Central Dravidian language primarily spoken in the state of Andhra Pradesh, India, where it is an official

Malayalam
Malayalam is a language spoken in India predominantly in the state of Kerala. It is one of the 22 scheduled languages of India with official

Figure 12.18 : Desipad Windows app

To create a new Windows app in Visual Studio, we need to create a new project and select the Windows store template with preferred application name.

☞ **Check Your Progress 2**

1) The two options to develop mobile apps in Windows platform are ______ and ______ .

2) ____ represents style name in XAML.

3) _______ resource has load time resolution in XAML .

4) ______ defines the page layout, page controls in Windows app.

12.5 SUMMARY

In this unit, we started discussing various development tools for mobile platforms. We started looking at Android tools such as Android studio, Eclipse and migration from Eclipse to Android studio. For iOS platform, Xcode and Swift programming language were introduced. Finally, for Windows platform, C# and XAML were introduced.

12.6 SOLUTIONS/ANSWERS

Check Your Progress 1

1) Android monitor

2) Gradle

3) Strings.xml

4) module

5) Fix-it

6) Interface builder

7) Let

8) dictionaries

Check Your Progress 2

1) C++/C#/VB with XAML and JavaScript, HTML 5 and CSS3

2) x:Key

3) static resource

4) XAML component

12.7 FURTHER READINGS

References

- https://www.airpair.com/android/android-studio-vs-eclipse
- http://android-developers.blogspot.in/2014/12/android-studio-10.html
- http://en.wikipedia.org/wiki/Eclipse_(software)

Software Development Tools

- http://developer.android.com/tools/studio/index.html
- http://tools.android.com/tech-docs/new-build-system/user-guide
- http://developer.apple.com/
- https://developer.apple.com/swift/blog
- https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language
- http://programmingincsharp.com/
- http://msdn.microsoft.com/en-us/

UNIT 13 CROSS PLATFORM DEVELOPMENT TOOLS

Structure

13.0 INTRODUCTION

Mobile apps are rapidly becoming the primary channels to deliver superior end user experiences. Organizations are choosing native mobile apps to implement their mobile-first strategy. Mobile apps are the de-facto customer engagement platforms which directly impact the organization's revenue.

There are various mobile platforms such as Apple iOS, Google's Android, Windows phone etc. Additionally there are variety of mobile devices differing in form factors, capabilities, resolution and such. As such the development platform for these mobile systems are also different: Apple iOS uses Swift/Objective-C, Android uses Java and Windows uses C#/XAML. The app development standards and patterns are also different: iOS uses MVC pattern whereas Windows app uses Model-View-Viewmodel (MVVM) pattern. Hence in order to develop apps for all mobile platforms we need understanding of diverse languages, standards and patterns. This also impacts the skill set and resources needed for such development projects.

In order to save time, effort and reuse the code across platforms we use cross platform development tools such as PhoneGap, Titanium and Xamarin. These cross platform tools would help us to reuse the code across mobile platforms. Cross platform tools implement the "Write Once Run Anywhere" concept.

13.1 OBJECTIVES

After going through this unit, you should be able to understand:

- various approaches for cross platform mobile app development and its applicable scenarios;
- key concepts of Xamarin for Android and iOS development;
- key concepts of PhoneGap for Android and iOS development; and
- the advantages and disadvantages of cross-platform mobile app development tools.

13.2 VARIOUS APPROACHES FOR CROSS PLATFORM MOBILE APP DEVELOPMENT

Native mobile apps provide superior user experience and leverages native and device-specific features. They provide the best performance and are more suited for game kind of applications. Native apps are distributed through app stores. However, the main drawback of native app development is that in order to develop apps for various platforms, we need developers with varied skill sets. This would lead to escalation in price, effort and impacts time to market. Total cost of ownership (TCO) and the overall license cost would be high for developing apps to various platforms natively.

In order to address this problem, there are various tools and techniques to develop cross platform mobile apps. In this section, we will briefly discuss key cross platform mobile app development techniques.

13.2.1 Web Based Cross Platform Mobile App Development

The following are salient features of Web based cross platform mobile app development:

- Developers will develop web applications that can run in mobile web browsers. This typically follows Model-View-Controller (MVC) architecture;
- Javascript based widgets and UI components would communicate with back end server components (such as servlets, JSP) through light weight services (such as REST based services);
- Backend server components would interact with enterprise interfaces such as database, content management systems, internal applications etc.;
- Due to emergence of HTML 5 standards, it is possible to have limited access to some of the device features through this approach;
- All modern mobile platforms such as Android, iOS, Blackberry, Windows phone support a full-fledged browser that can be leveraged for this approach;
- The main technologies are HTML 5, CSS, JavaScript with RWD and backend technologies are Java, and .Net;
- The key Javascript frameworks that can be used in this approach are jQueryMobile, Sencha Touch, and Dojo Mobile; and

- TCO is minimal as we need to use standard web technologies.

We can use this approach when we would like to reuse the web channels for mobile apps. We can also use this approach when we quickly want to launch the application across various mobile platforms.

If the application needs native device features and device hardware, we cannot use this approach. Visually rich applications like gaming apps are not suited for this kind of approach.

13.2.2 Hybrid Approach for Cross Platform Mobile App Development

The following are salient features of hybrid approach for cross platform mobile app development:

- Developers will develop mobile apps in standard web technologies such as HTML 5 and JavaScript;
- JavaScript APIs expose some of the native device features such as camera, contacts etc. through a bridge;
- App developers need the knowledge of Javascript; and
- Adobe PhoneGap, Apache Cordova are some of the frameworks that provide this approach.

We can use this approach when we would like to use HTML 5 based web applications on mobile and provide a good native device access. The code can be shared across multiple mobile platforms.

If the application needs high quality rich UI experience with high performance, then, this approach should not be followed.

13.2.3 Single Language Approach for Cross Platform Mobile App Development

The following are salient features of single language approach for cross platform mobile app development:

- App is developed in single language;
- We could get native look and feel user experience;
- High performance;
- APIs will be provided to access the native device features; and
- Some tools and frameworks that can be used for this approach are Xamarin, Appcelerator Titanium etc.

We can use this approach when we would like to use reuse the code across various platforms and want native UI and performance.

If the application needs high quality rich UI experience with high performance this approach is not suited.

13.2.4 Mobile Middleware Approach for Cross Platform Development

The following are salient features for Mobile middleware approach for cross platform development:

- A dedicated Mobile Enterprise Application platform (MEAP) acts as mobile middleware;
- The middleware provides adaptors to interact with various mobile platforms;
- The middleware provides various features such as integration, authentication, security, governance, and mobile management features etc.;
- The middleware also interacts with back end systems such as database, ERP, web applications, portals etc. and
- Systems such as IBM Worklight is used for this approach.

We can use this approach when we want to support various types of platforms and have a centralized mobile management platform for all mobility needs.

If the enterprise wants to have reduced total cost of ownership (TCO) or use open source technologies, then, this is not applicable.

13.3 XAMARIN

Xamarin is a cross-platform with an inbuilt IDE and SDK based on .net platform. Xamarin can be used to develop apps for Android, iOS and Windows. Xamarin SDK uses Mono Touch for iOS and Mono Droid for Android for building the mobile apps for respective platforms. Xamarin. Android is the Android variant that is used for accessing native UI features, device features etc.; Similarly Xamarin.iOS is the iOS variant.

Xamarin studio is the IDE that can be used to develop and deploy mobile apps and is available for Windows and Mac platforms. Xamarin can be used as mobile middleware to provide native user interface. Xamarin is also compatible with MS Visual studio. Code debugging can be performed using MS Visual Studio integration of Xamarin and XCode is needed for debugging for iOS.

Using Xamarin we can write the business logic code in C# and access platform specific APIs and user interfaces. The code will be compiled using Xamarin compiler that generates just-in-time compile binaries for Android and ahead-of-time native ARM binaries for iOS. Using this method, it is possible to share and reuse the business logic across various mobile platforms.

The following are some of the features of Xamarin:

- Presentation layer consisting of screens, controls and widgets.
- Application Layer that contains platform specific implementation details.
- Business Layer consisting of business logic.
- Data Layer for handling persistence and data related transactions.
- Service Access layer that provide data access services.

The main advantages of Xamarin are as follows:

- It provides a native look and feel.
- Easily integrates with MS Visual Studio. Enables a C# developer to develop the apps for iOS and Android.
- Good performance.

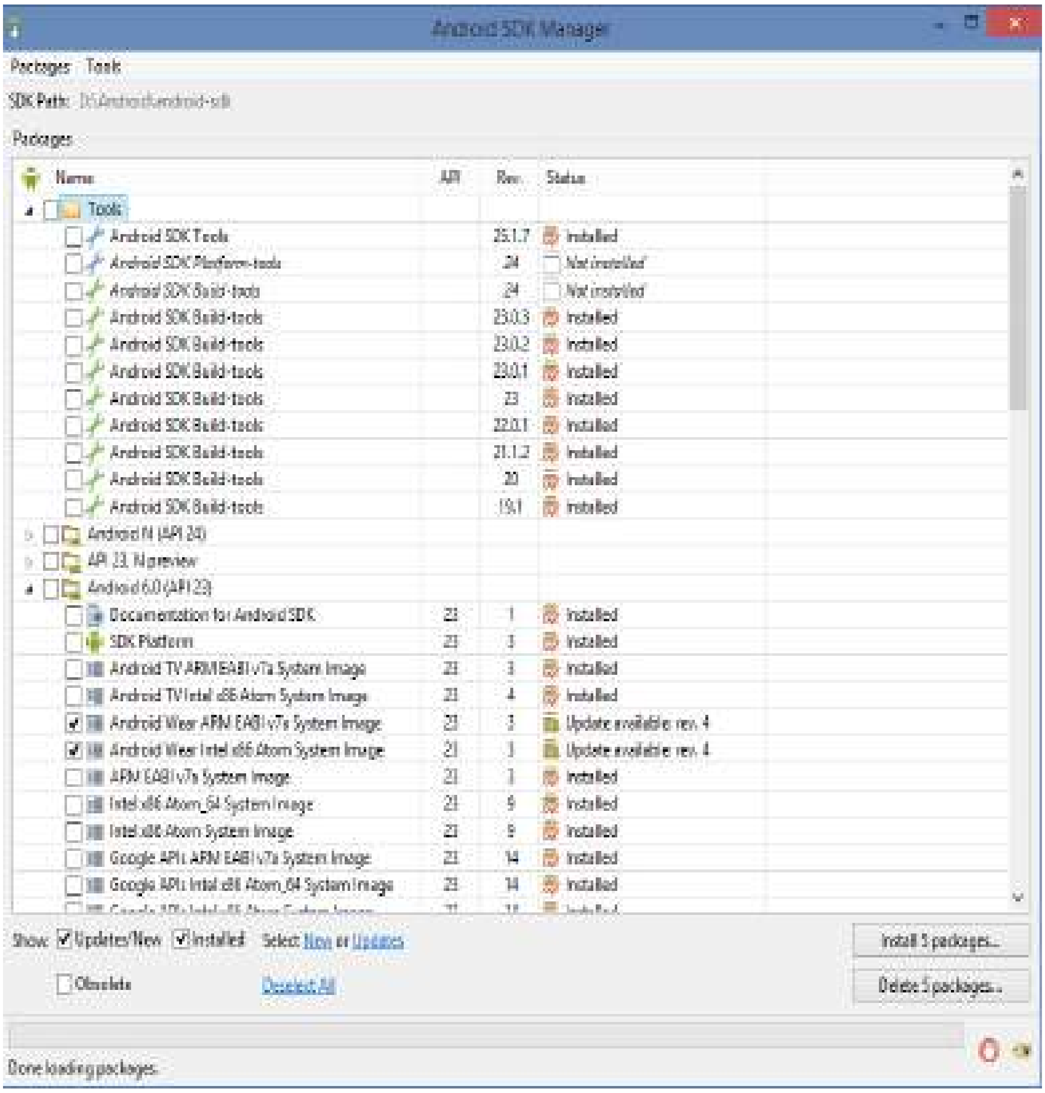
Xamarin.Android and Xamarin.iOS provide framework for invoking platform specific features using C# calls. Using these frameworks, developers can develop full-fledged mobile apps using .net framework and C#.

13.3.1 Xamarin.Android

The following are high level steps to setup Xamarin.Android:

1) Android SDK (https://developer.android.com/studio/index.html) and NDK (https://developer.android.com/ndk/downloads/index.html) are used with MS Visual Studio for developing Android apps.

2) Install Android APIs from SDK manager (Figure 13.1).



SDK Path: D:\Android\android-sdk

Name	API	Rev.	Status
□ Tools			
□ Android SDK Tools		25.1.7	Installed
□ Android SDK Platform-tools		24	Not installed
□ Android SDK Build-tools		24	Not installed
□ Android SDK Build-tools		23.0.3	Installed
□ Android SDK Build-tools		23.0.2	Installed
□ Android SDK Build-tools		23.0.1	Installed
□ Android SDK Build-tools		23	Installed
□ Android SDK Build-tools		22.0.1	Installed
□ Android SDK Build-tools		21.1.2	Installed
□ Android SDK Build-tools		20	Installed
□ Android SDK Build-tools		19.1	Installed
□ Android N (API 24)			
□ API 23, N preview			
□ Android 6.0 (API 23)			
□ Documentation for Android SDK	23	1	Installed
□ SDK Platform	23	3	Installed
□ Android TV ARM EABI v7a System Image	23	3	Installed
□ Android TV Intel x86 Atom System Image	23	4	Installed
☑ Android Wear ARM EABI v7a System Image	23	3	Update available: rev. 4
☑ Android Wear Intel x86 Atom System Image	23	3	Update available: rev. 4
□ ARM EABI v7a System Image	23	3	Installed
□ Intel x86 Atom_64 System Image	23	9	Installed
□ Intel x86 Atom System Image	23	9	Installed
□ Google APIs ARM EABI v7a System Image	23	14	Installed
□ Google APIs Intel x86 Atom_64 System Image	23	14	Installed

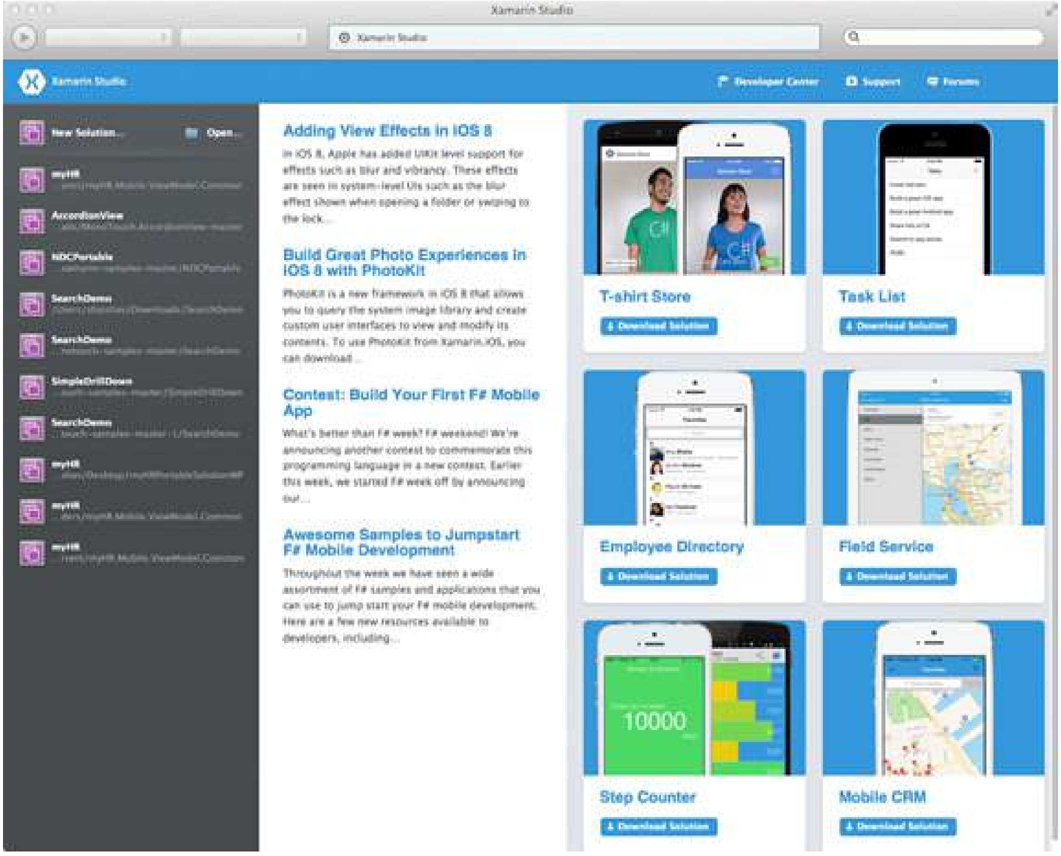
Show: ☑ Updates/New ☑ Installed □ Obsolete

Figure 13.1 : Installation of Android API from SDK manager

13.3.2 Xamarin.iOS

C# and Objective-C can be used for development in this framework. It connects native iOS and C#. Xamarin can be used to develop iOS apps. High level steps to develop a sample app listing the values in Apple iPhone app using Xamarin are given below:

1) Open Xamarin for iOS (Figure 13.2)



Adding View Effects in iOS 8

In iOS 8, Apple has added UIKit level support for effects such as blur and vibrancy. These effects are seen in system-level UIs such as the blur effect shown when opening a folder or swiping to the lock...

Build Great Photo Experiences in iOS 8 with PhotoKit

PhotoKit is a new framework in iOS 8 that allows you to query the system image library and create custom user interfaces to view and modify its contents. To use PhotoKit from Xamarin.iOS, you can download...

Contest: Build Your First F# Mobile App

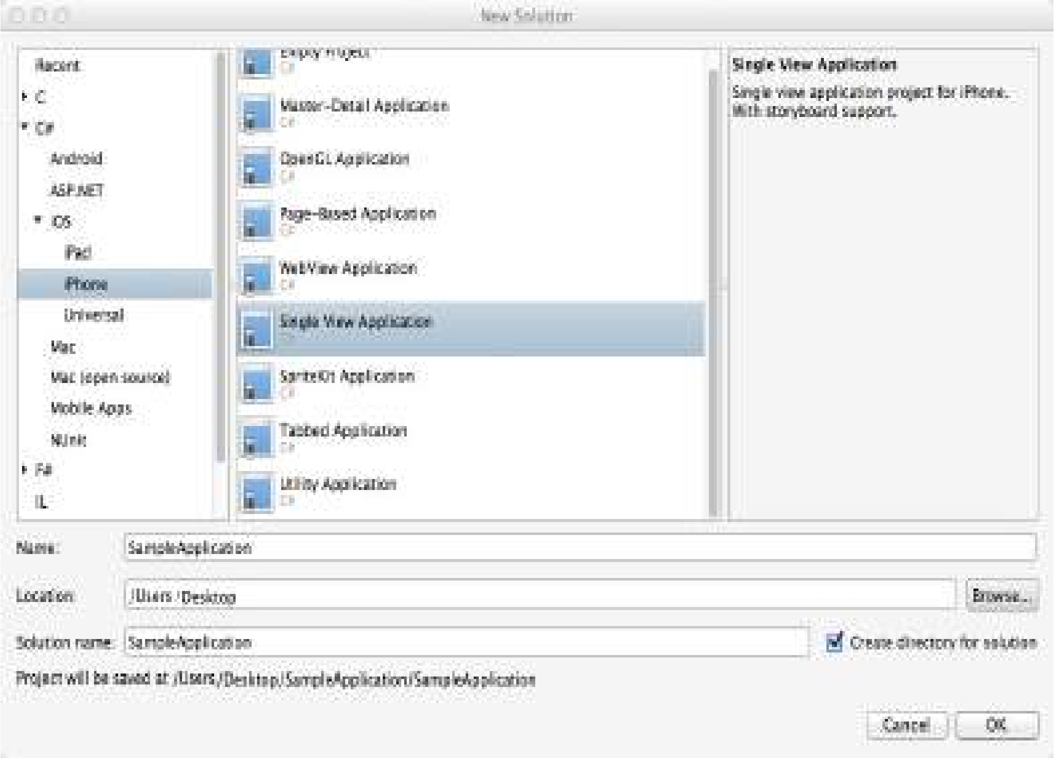
What's better than F# week? F# weekend! We're announcing another contest to commemorate this programming language in a new contest. Earlier this week, we started F# week off by announcing our...

Awesome Samples to Jumpstart F# Mobile Development

Throughout the week we have seen a wide assortment of F# samples and applications that you can use to jump start your F# mobile development. Here are a few new resources available to developers, including...

Figure 13.2 : Xamarin for iOS

2) Select the app type. For example, in the following screenshot, we have selected "Single View Application" (Figure 13.3).



Single View Application

Single view application project for iPhone. With storyboard support.

Name: SampleApplication

Location: /Users /Desktop

Solution name: SampleApplication ☑ Create directory for solution

Project will be saved at /Users/Desktop/SampleApplication/SampleApplication

Figure 13.3: App selection in Xamarin

3) Use ViewController.cs to open the Xcode based application (Figure 13.4)

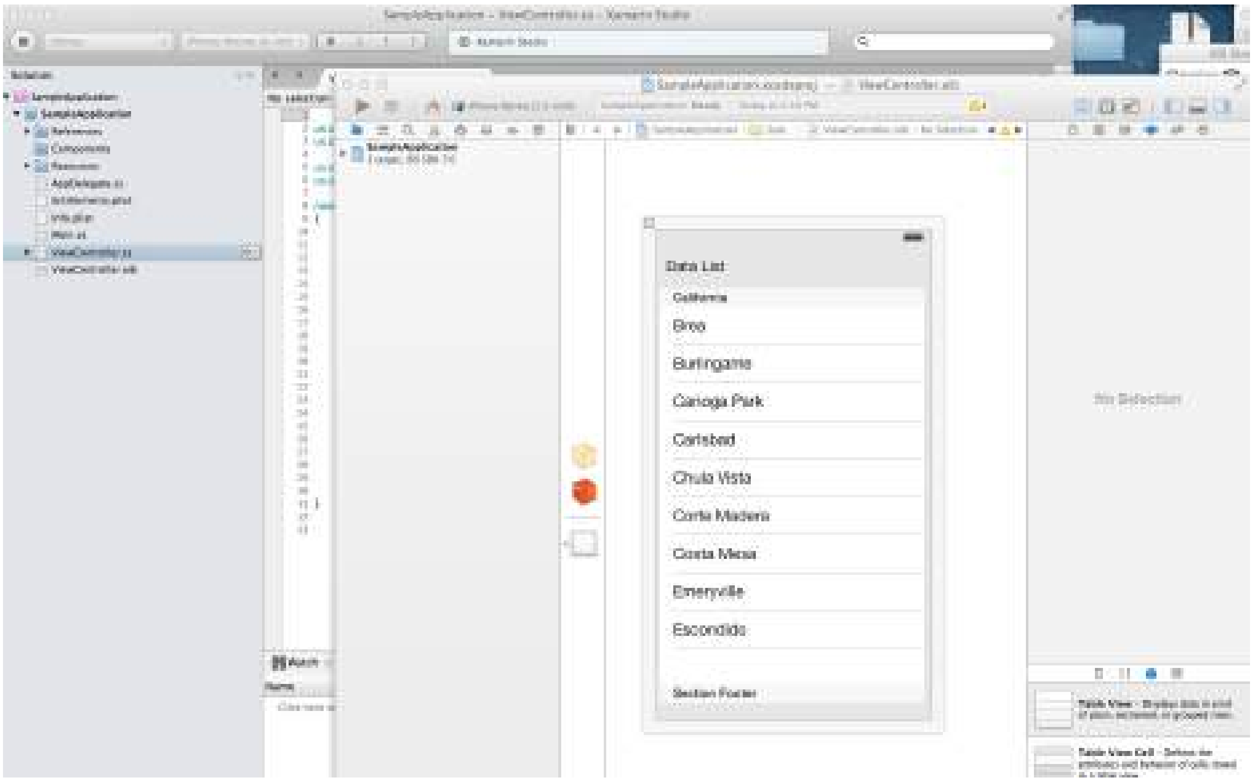
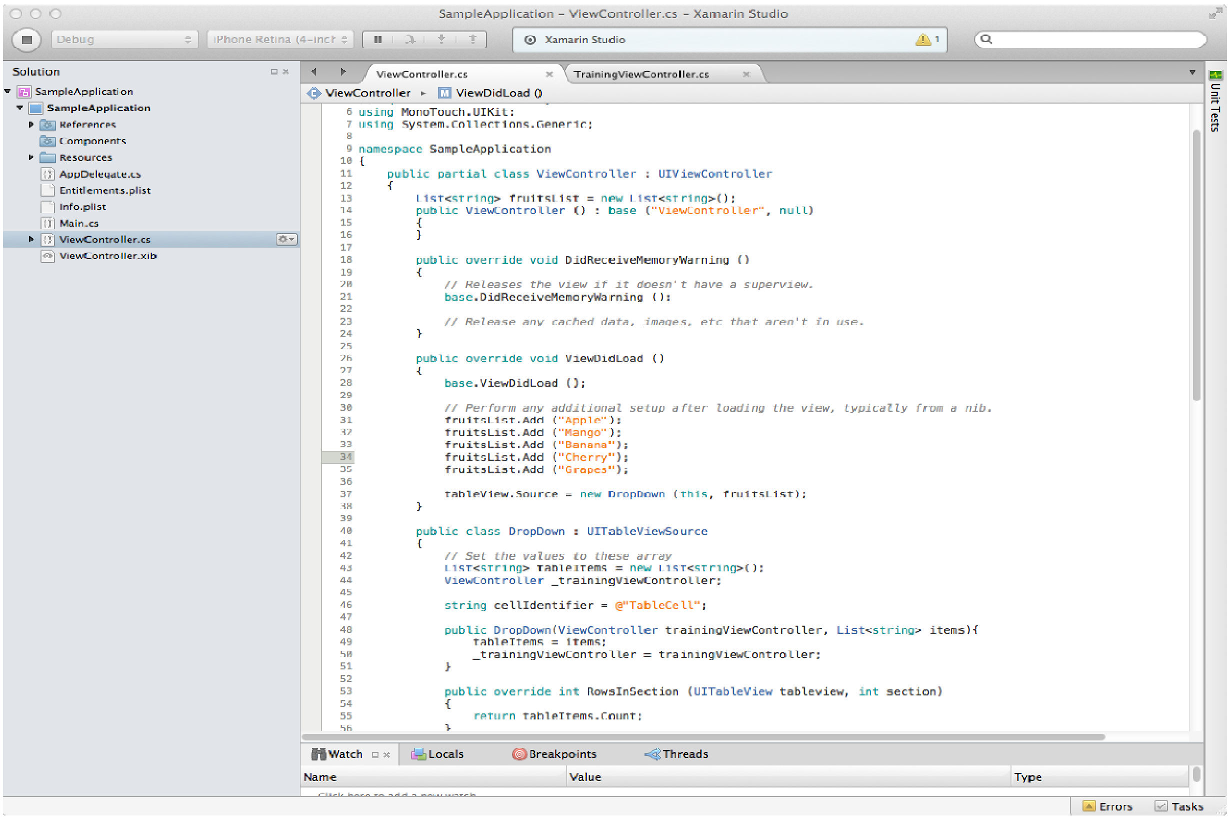


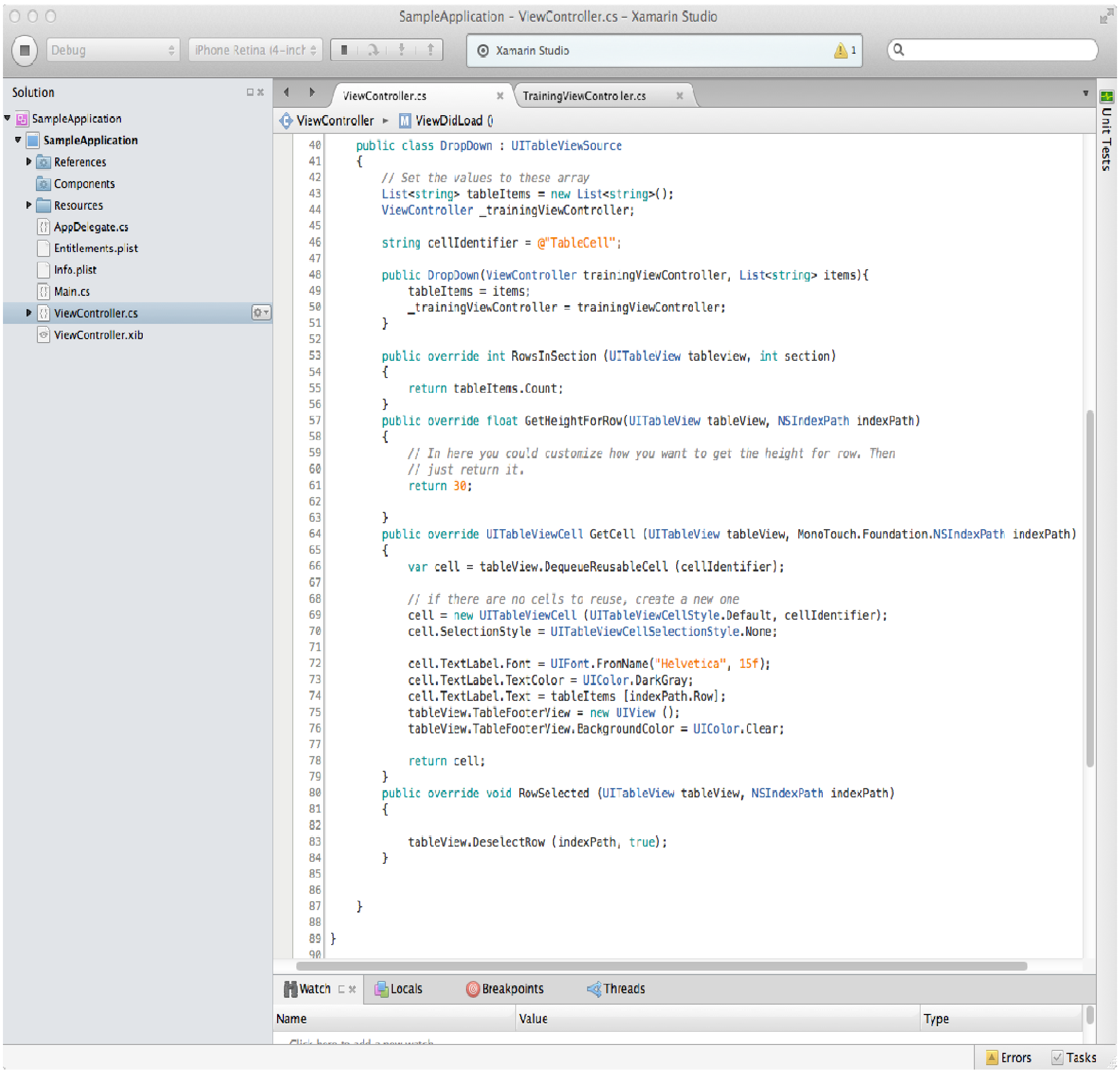
Figure 13.4: View Controller in Xamarin

4) Add needed elements. In this sample application we have added UILabel and UITableView. In ViewController.cs, add data list to UITableView (Figure 13.5 and Figure 13.6).



```
using MonoTouch.UIKit;
using System.Collections.Generic;

namespace SampleApplication
{
    public partial class ViewController : UIViewController
    {
        List<string> fruitsList = new List<string>();
        public ViewController () : base ("ViewController", null)
        {
        }

        public override void DidReceiveMemoryWarning ()
        {
            // Releases the view if it doesn't have a superview.
            base.DidReceiveMemoryWarning ();

            // Release any cached data, images, etc that aren't in use.
        }

        public override void ViewDidLoad ()
        {
            base.ViewDidLoad ();

            // Perform any additional setup after loading the view, typically from a nib.
            fruitsList.Add ("Apple");
            fruitsList.Add ("Mango");
            fruitsList.Add ("Banana");
            fruitsList.Add ("Cherry");
            fruitsList.Add ("Grapes");

            tableView.Source = new DropDown (this, fruitsList);
        }

        public class DropDown : UITableViewSource
        {
            // Set the values to these array
            List<string> tableItems = new List<string>();
            ViewController _trainingViewController;

            string cellIdentifier = @"TableCell";

            public DropDown(ViewController trainingViewController, List<string> items){
                tableItems = items;
                _trainingViewController = trainingViewController;
            }

            public override int RowsInSection (UITableView tableview, int section)
            {
                return tableItems.Count;
            }
```

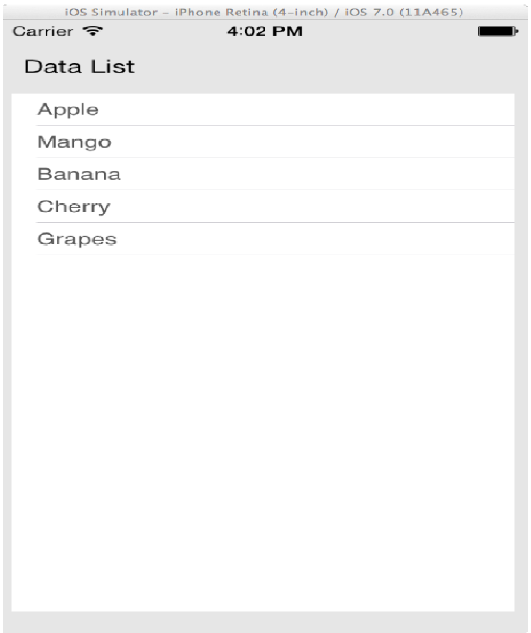
Figure 13.5: App Code in Xamarin



```
    public class DropDown : UITableViewSource
    {
        // Set the values to these array
        List<string> tableItems = new List<string>();
        ViewController _trainingViewController;

        string cellIdentifier = @"TableCell";

        public DropDown(ViewController trainingViewController, List<string> items){
            tableItems = items;
            _trainingViewController = trainingViewController;
        }

        public override int RowsInSection (UITableView tableview, int section)
        {
            return tableItems.Count;
        }
        public override float GetHeightForRow(UITableView tableView, NSIndexPath indexPath)
        {
            // In here you could customize how you want to get the height for row. Then
            // just return it.
            return 30;

        }
        public override UITableViewCell GetCell (UITableView tableView, MonoTouch.Foundation.NSIndexPath indexPath)
        {
            var cell = tableView.DequeueReusableCell (cellIdentifier);

            // if there are no cells to reuse, create a new one
            cell = new UITableViewCell (UITableViewCellStyle.Default, cellIdentifier);
            cell.SelectionStyle = UITableViewCellSelectionStyle.None;

            cell.TextLabel.Font = UIFont.FromName("Helvetica", 15f);
            cell.TextLabel.TextColor = UIColor.DarkGray;
            cell.TextLabel.Text = tableItems [indexPath.Row];
            tableView.TableFooterView = new UIView ();
            tableView.TableFooterView.BackgroundColor = UIColor.Clear;

            return cell;
        }
        public override void RowSelected (UITableView tableView, NSIndexPath indexPath)
        {

            tableView.DeselectRow (indexPath, true);
        }


    }

}
```

Figure 13.6: App code in Xamarin

5) When we run the application in iOS simulator, the output is displayed as shown in Figure 13.7



Data List

Apple
Mango
Banana
Cherry
Grapes

Figure 13.7 : iOS Simulator

13.4 PHONEGAP

HTML5, JavaScript and CSS are most popular technologies and standards to develop cross-platform mobile apps. PhoneGap is one of the most popular cross-platform development tools that uses HTML 5 compatible browser in the native app and provides JavaScript functions to access native platform capabilities. PhoneGap provides HTML 5 browser and a hybrid framework exposing APIs to access native mobile features. PhoneGap builds and runs an application for a device. PhoneGap is the original distributor of Apache Cordova. As HTML 5, JavaScript and CSS are standard technologies, it would be possible to easily develop and reuse the code and deploy it to various platforms using this technology. The trade-off is the performance overhead that occurs in this process.

PhoneGap provides APIs to access native mobile features such as camera, GPS, accelerometer, contacts, sensors etc. The basic architecture of PhoneGap is shown in Figure 13.8.

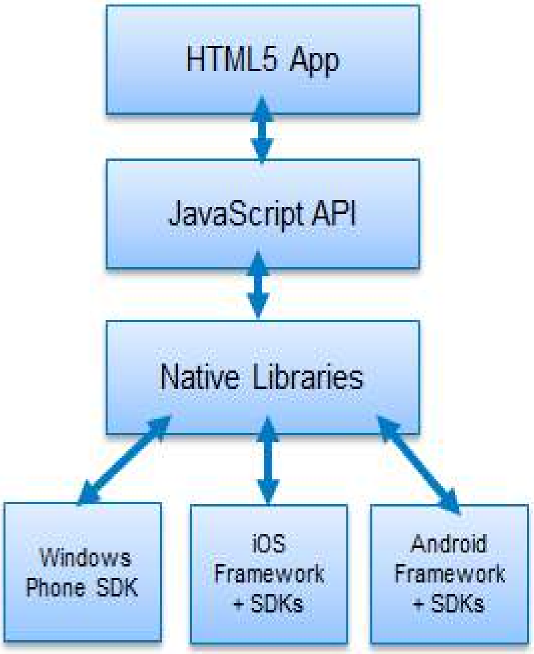


Figure 13.8: Architecture of PhoneGap

13.4.1 Developing an Android App in PhoneGap

The following are steps involved in developing a simple Hello World Android App in PhoneGap:

1) Download the PhoneGap libraries

2) Create a new Android project with empty activity (Figure 13.9). Then, its followed by a new Android application (Figure 13.10).

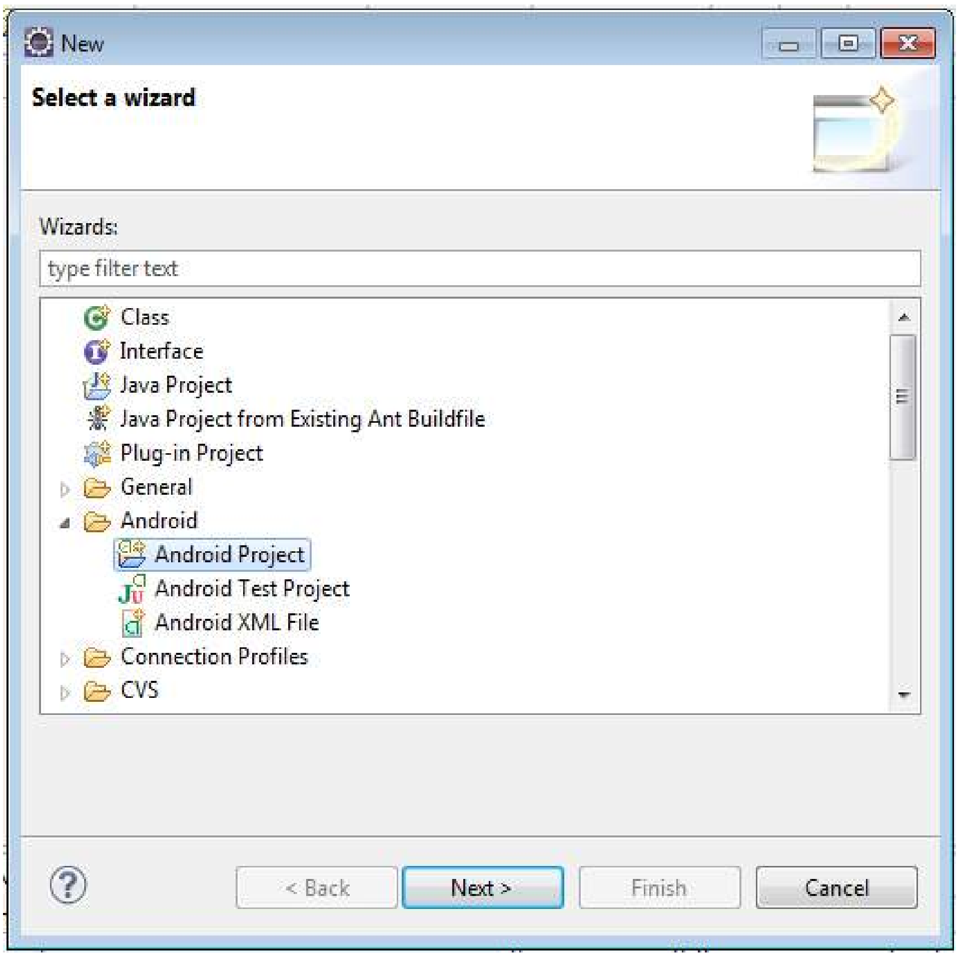
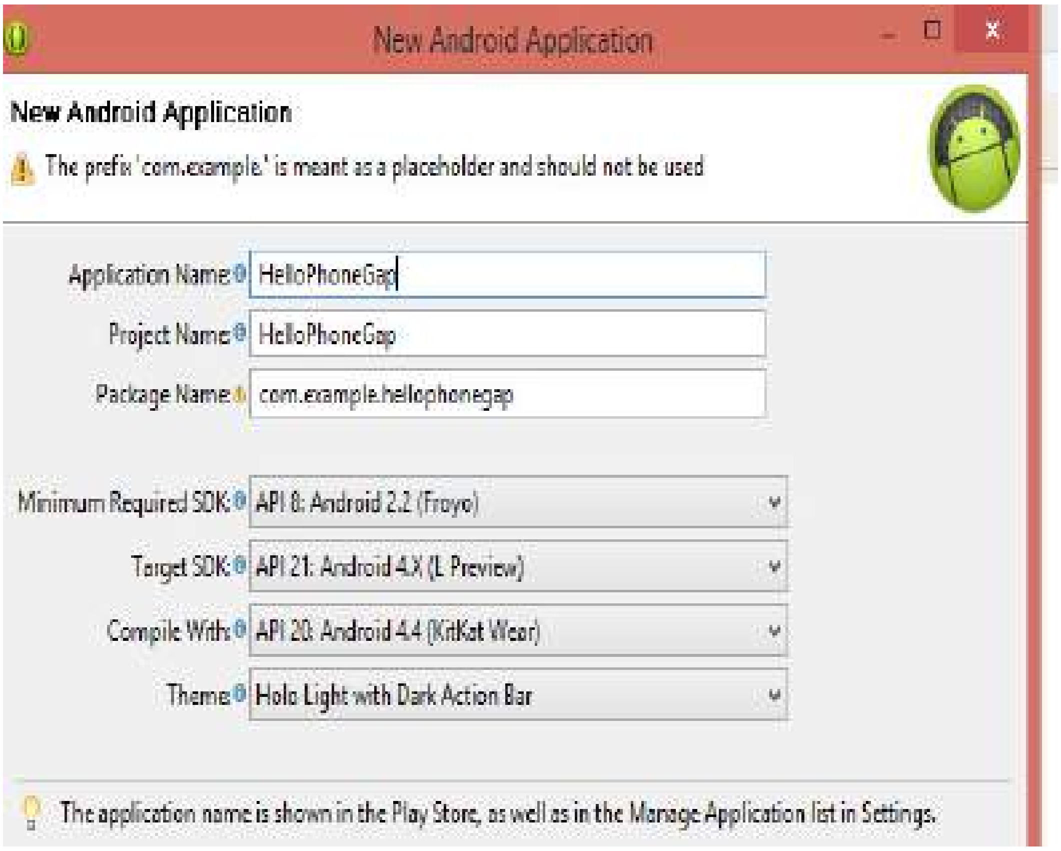


Figure 13.9 : Creation of a new Android project



New Android Application

The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: HelloPhoneGap

Project Name: HelloPhoneGap

Package Name: com.example.hellophonegap

Minimum Required SDK: API 8: Android 2.2 (Froyo)

Target SDK: API 21: Android 4.X (L Preview)

Compile With: API 20: Android 4.4 (KitKat Wear)

Theme: Holo Light with Dark Action Bar

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

Figure 13.10 : New Android Application

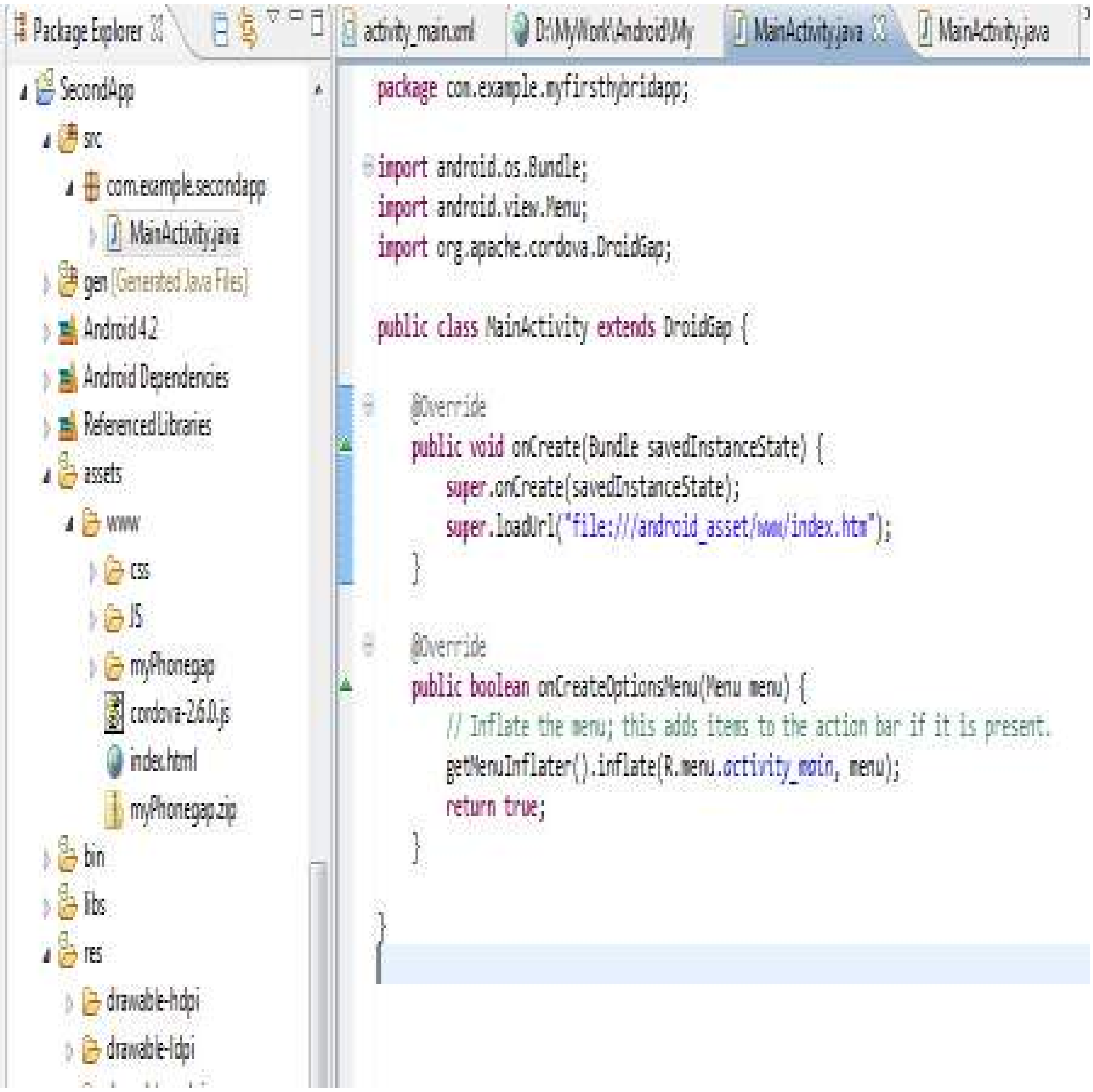
3) Create “www” folder within assets and we need to create all HTML, JS, and CSS in this folder.

4) Move the “cordova.jar” from PhoneGap libraries into the project’s lib folder and “cordova.js” into the newly created “www” folder. Copy xml folder to “res” folder.

5) Create and edit the index.html file in “www” folder. Add the reference of the Cordova.js in the index html

<script type="text/javascript" charset="utf-8" src="cordova-2.6.0.js"></script>

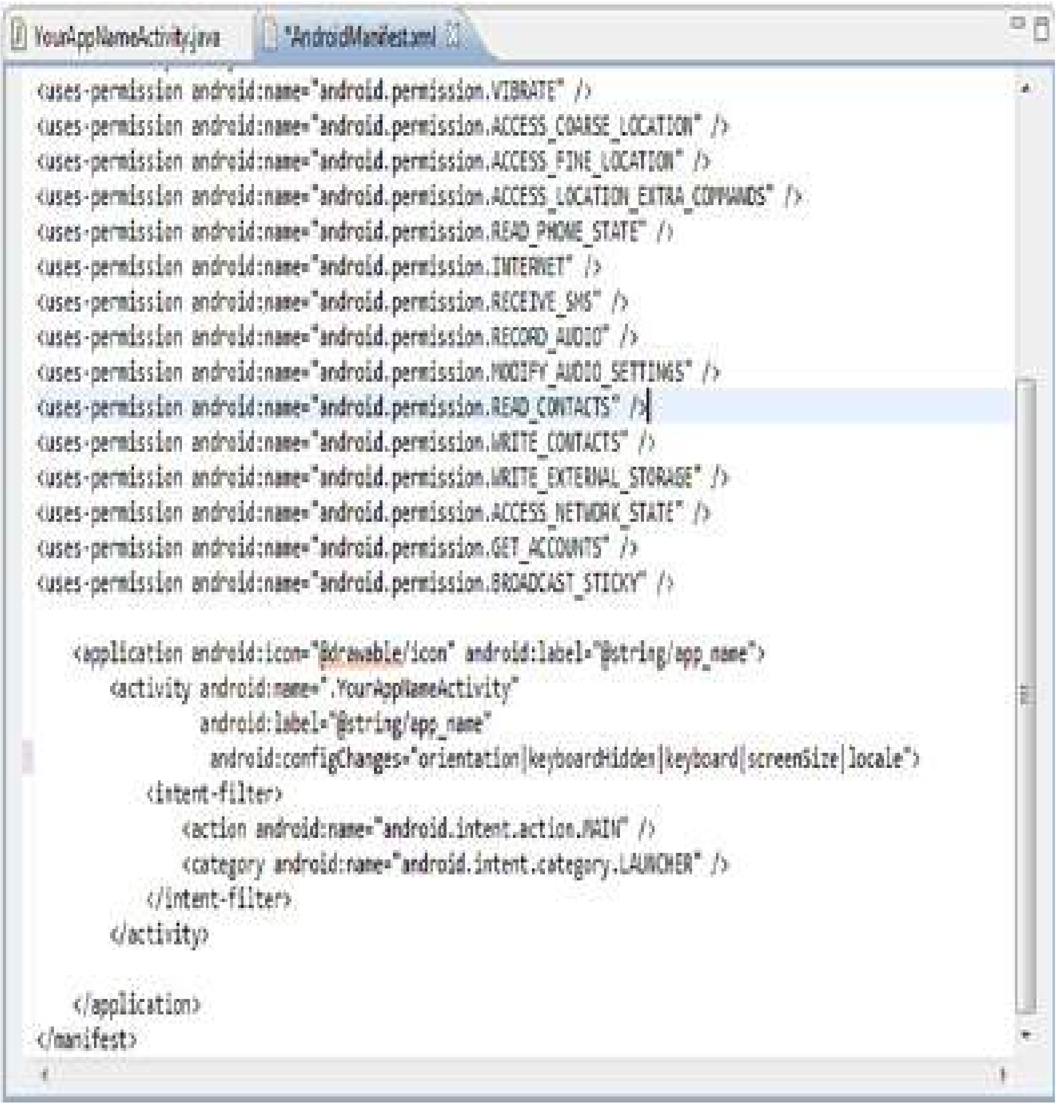
6) Edit the activity file and load the html file as shown in Figure 13.11



```
package com.example.myfirsthybridapp;

import android.os.Bundle;
import android.view.Menu;
import org.apache.cordova.DroidGap;

public class MainActivity extends DroidGap {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        super.loadUrl("file:///android_asset/www/index.htm");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }

}
```

Figure 13.11 : Activity file for Android app

7) Add needed metadata in “AndroidManifest.xml” and provide suitable permissions (Figure 13.12).



```
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />

  <application android:icon="@drawable/icon" android:label="@string/app_name">
    <activity android:name=".YourAppNameActivity"
          android:label="@string/app_name"
           android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>

  </application>
</manifest>
```

Figure 13.13 : Android app output in emulator

8) On running the application in emulator, output is shown in Figure 13.13

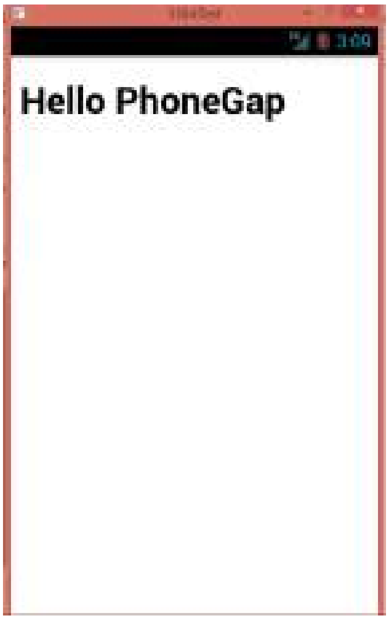


Figure 13.13 : Android app output in emulator

13.4.2 Developing an iOS App in PhoneGap

The following are steps for developing an iOS app in PhoneGap:

1) Download the PhoneGap library from official website (http://phonegap.com/)

2) Create an iOS project in lib/iOS folder of the extracted content (Figure 13.14).

3) Use the shell command to create a simple Cordova iOS project.

4) Open the project using the <projectname>.xcodeproj file.

5) After running the app, the basic application is rendered as shown in Figure 13.15 in simulator

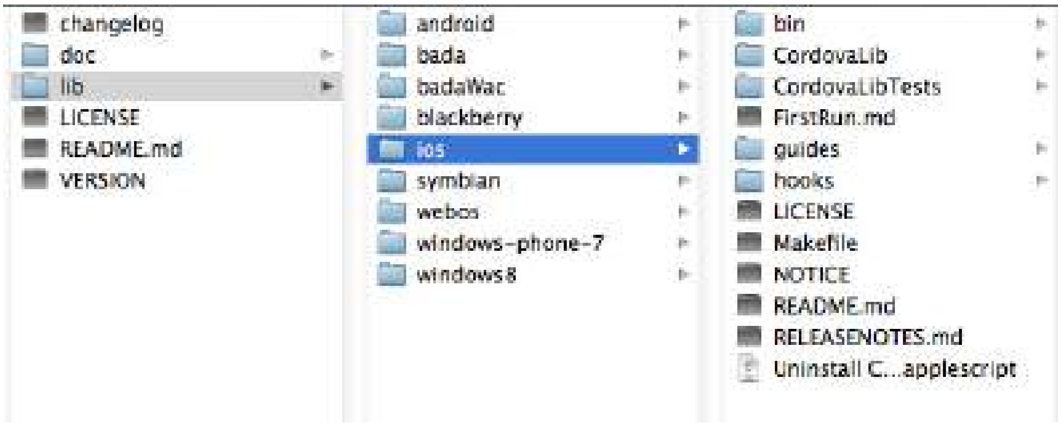


Figure 13.14 : iOS project in lib/iOS folder

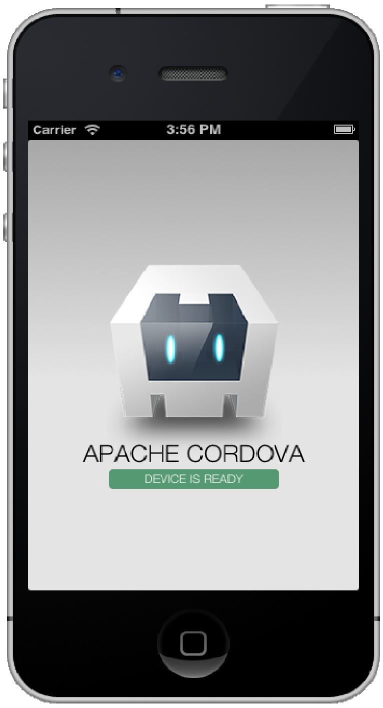


Figure 13.15 : Output of iOS app in simulator

☞ **Check Your Progress 1**

1) Xamarin uses __ platform

2) Xamarin SDK uses _____ for iOS and _____ for Android

3) ____ contains platform specific implementation details in a typical app

4) Android APIs can be installed from the _____ in Xamarin

5) _____ opens the XCode application in Xamarin studio for iOS

13.5 ADVANTAGES AND DISADVANTAGES OF CROSS PLATFORM MOBILE APP DEVELOPMENT TOOLS

As discussed earlier, there is a huge diversity in the mobile platform space. There are various mobile devices with different form factors and device capabilities. Various mobile platforms need specialized developers in varied skill sets. Cross platforms play a key role in addressing this challenge.

The following are some of the advantages of cross platform mobile app development tools

Given below are various advantages of cross platform mobile app development tools:

- Cross-platform app development tools provide native look and feel and we can reach wider audience quickly.
- We can use HTML standards (such as in PhoneGap) or a single source code (such as C# in Xamarin) and reuse/share it for various mobile platforms. It would be efficient to manage the mobile apps due to single code base.
- Due to single code base, the cost would be reduced with minimal learning curve.
- Unit testing and bug fixing would be easier due to single code base.
- Updates, maintenance and provisioning of apps to various mobile platforms would be easy.
- Cross platform tools provide uniform look and feel across mobile platforms.
- Time to market for mobile apps will be reduced.

The following are some of the disadvantages of cross platform app development tools:

- Some of the unique/specific device and platform features cannot be exploited within cross platform tools.
- Development of advanced graphics such as 3D graphics is a challenge in these apps.
- Designing a truly exceptional user experience on all mobile platforms would be challenging as each mobile platform has its own unique screen layout, and elements.

- Integrating with native cloud services, notifications, settings, and storage services would be a challenge.
- Mobile apps developed with cross-platform tools will not be able to leverage all security features of the native platform and hence are relatively more vulnerable to security attacks.
- Performance would be challenge sometimes leading to slow rendering issues.

☞ **Check Your Progress 2**

1) HTML, CSS, Javascript are copied to _____________ folder while creating a PhoneGap app.

2) ___________ file contains the permissions in Android app developed using PhoneGap.

3) _____________ leads to minimized cost, effort and learning curve in apps developed from cross platform tools.

13.6 SUMMARY

In this unit, we started discussing Xamarin and PhoneGap tools. We looked at detailed steps for developing Android apps and iOS apps in Xamarin. We also looked at key features and advantages of Xamarin. We then looked at steps for developing a simple Android app and an iOS app using PhoneGap. Lastly, we saw the key advantages and disadvantages of cross platform app development tools.

13.7 SOLUTIONS/ANSWERS

Check Your Progress 1

1) .net

2) MonoTouch, MonoDroid

3) Application Layer

4) SDK manager

5) ViewController.cs

Check Your Progress 2

1) www

2) AndroidManifest.xml

3) single code base

13.8 FURTHER READINGS

References

1) http://developer.xamarin.com

2) http://stackoverflow.com/questions/tagged/monodroid+or+monotouch+or+xamarin?sort=active

3) http://blog.xamarin.com

4) https://developer.android.com

5) https://university.xamarin.com/resources/working-with-android-emulator

6) http://phonegap.com/

UNIT 14 PUBLISHING TOOLS AND DEVELOPER PROGRAM

Structure

14.0 INTRODUCTION

The official channel for distribution of mobile apps is the marketplace stores. Each vendor has a specific app store wherein certified mobile apps are available. The app stores provide the much needed governance for the apps ensuring their certification compliance, security compliance, content policy compliance and other regulations. The app stores is the single interface for app developers to post and market their apps.

The app stores also provide intuitive interface for the end users to easily search, discover the apps through search UI, categorization, recommendation, sorting by rating/popularity and other criteria.

In this unit, we will look at the details of app stores for Android, iOS and Windows apps.

14.1 OBJECTIVES

After going through this unit, you should be able to understand

- key prerequisites and steps for publishing Android apps to Google Play store;
- key prerequisites for publishing and detailed steps for publishing iOS apps to Apple app store; and
- key prerequisites for publishing, best practices and detailed steps for publishing Windows apps to Windows store.

14.2 GOOGLE PLAY STORE

Google play store is the marketplace for Android mobile apps. Once the Android app is developed, tested and ready to be published, we need to publish it to the Google play store so that users can download and use it.

Google also offers other media such as digital music, video, books, magazines, TV programs through play store. Google play store forms a single unified distribution channel for Google's digital offerings.

Google play store has over 2.7 million Android apps as on 2017.

14.2.1 Prerequisites for Publishing Android Apps to Google Play Store

The following are the essential prerequisites for publishing Android apps to Google play store:

- A fully developed Android app as per the specification.
- A .apk file which packages the app.
- Eclipse IDE connected to Internet.
- A valid credit card to pay the Google developer license fee and register for Google Developer program at https://console.developers.google.com/
- Two representative screenshots of the app.

The following are the steps that need to be followed for publishing Android app from Eclipse IDE to Google play store:

1) Open the Android app in Eclipse

2) Right click on the app and select Export and "Export Android Application" option (Figure 14.1)

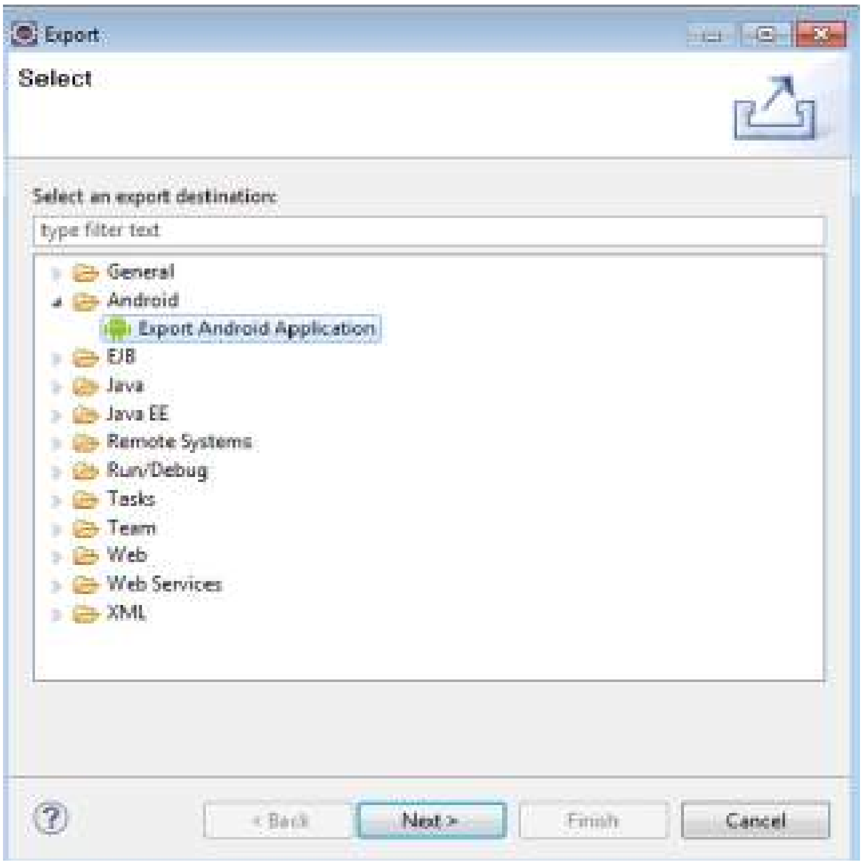
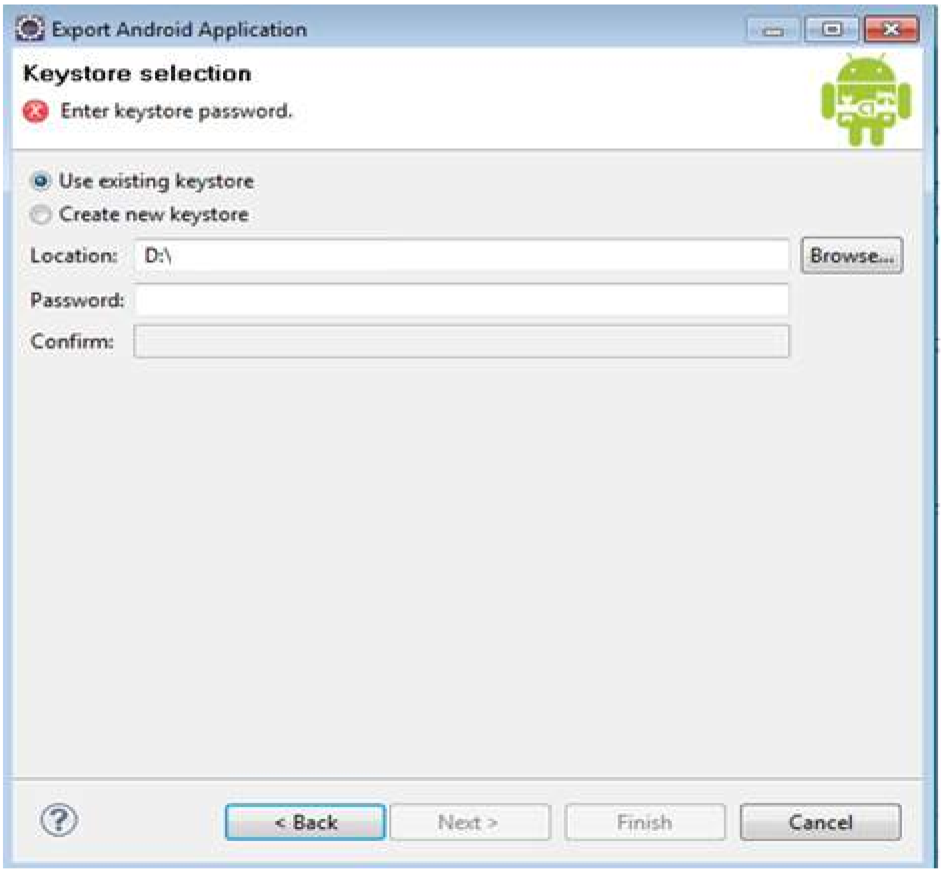


Figure 14.1 : Export Android application

3) Eclipse IDE will check for errors and if there are no errors we can continue to the next step.

4) In the next step we need to use the key store values (create keystore if not done already). This includes storing key store values to local file and entering password (Figure 14.2 and Figure 14.3).



Export Android Application

Keystore selection

Enter keystore password.

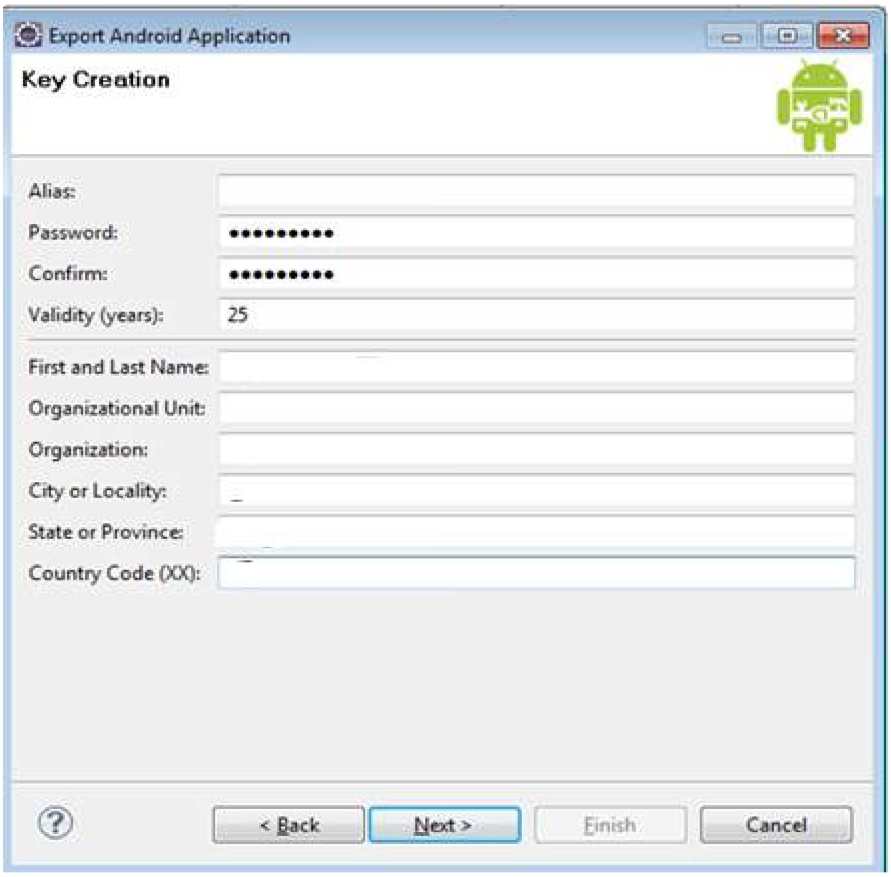
● Use existing keystore
○ Create new keystore

Location: D:\

Password:

Confirm:

Figure 14.2 : Selection of key store



Export Android Application

Key Creation

Alias:

Password: ••••••••••

Confirm: ••••••••••

Validity (years): 25

First and Last Name:

Organizational Unit:

Organization:

City or Locality:

State or Province:

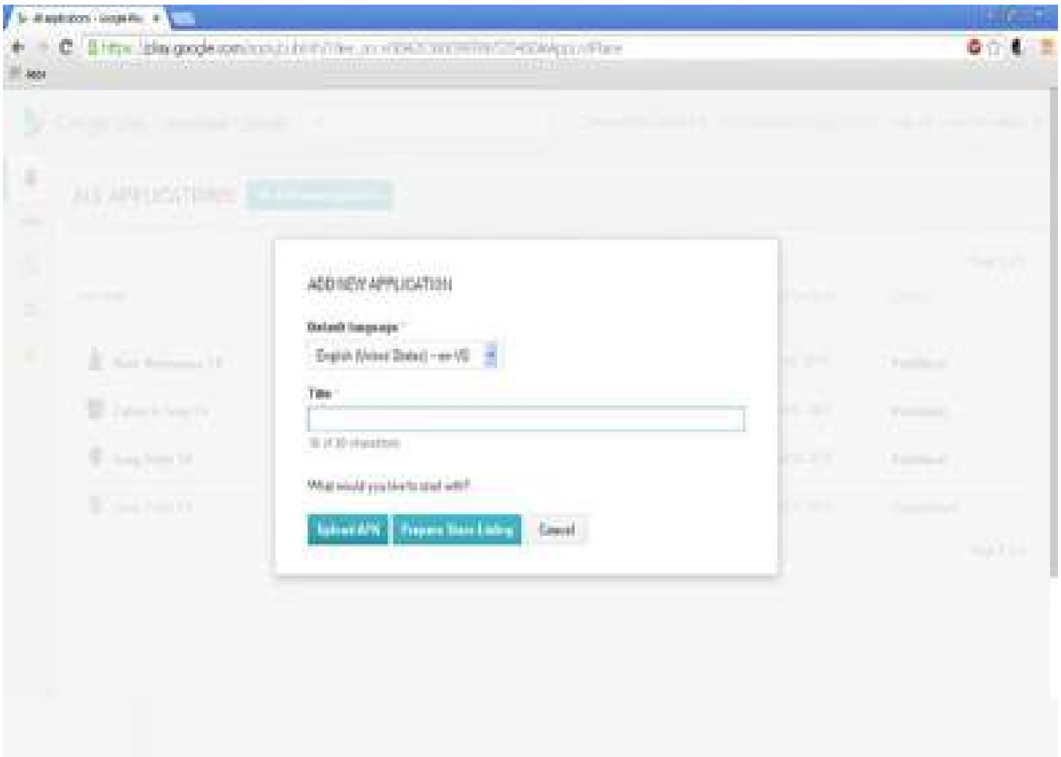
Country Code (XX):

Figure 14.3 : Password creation

5) Exporting the .apk file to a valid destination.

6) Open Google Play Store (https://play.google.com/apps/publish) and log into Google developer console.

7) Select "Add new application" and select the application (Figure 14.4)



ADD NEW APPLICATION

Default language *

English (United States) – en-US

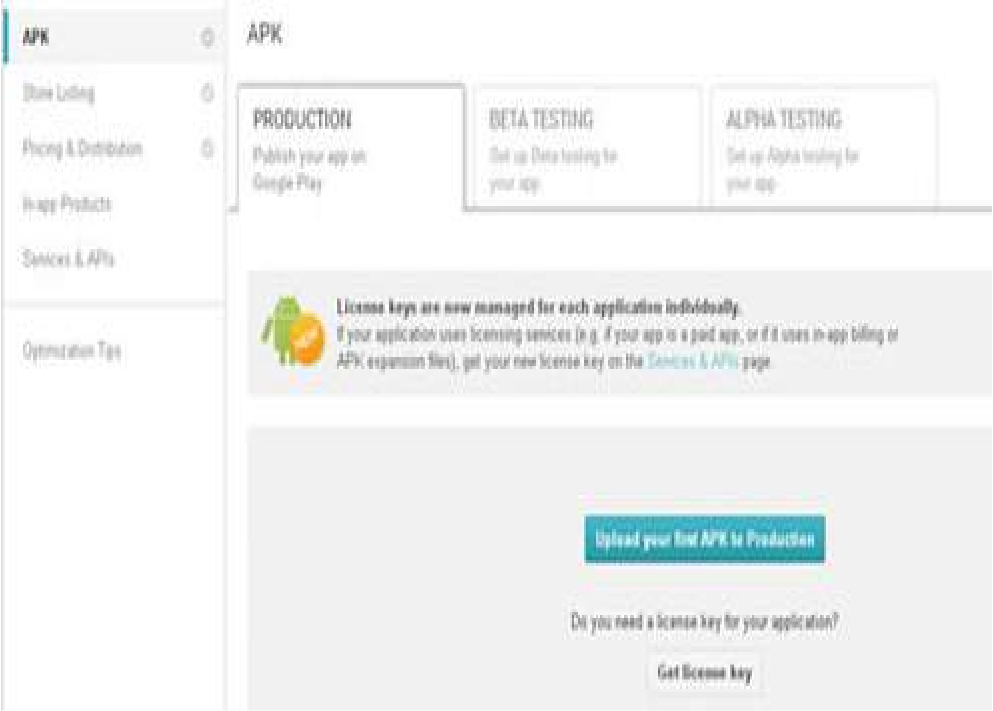
Title *

0 of 30 characters

What would you like to start with?

Figure 14.4 : Adding New Application

8) Upload .apk file to production (Figure 14.5)



License keys are now managed for each application individually.
If your application uses licensing services (e.g. if your app is a paid app, or if it uses in-app billing or APK expansion files), get your new license key on the Services & APIs page.

Do you need a license key for your application?

Figure 14.5: Upload .apk file

9) Once the upload is successful we need to add other details such as app description, screenshots, app icon, app categorization, contact details, privacy policy, maturity level details and such. We need to provide Website, email, and phone where customers can contact developer.

10) In the “Pricing and distribution” section we can specify if the app is free or paid, distribution countries, compliance to content guidelines and US export laws and such. For content ratings, guidelines can be referred at (https://support.google.com/googleplay/answer/6209544?hl=en-IN) (Figure 14.6).

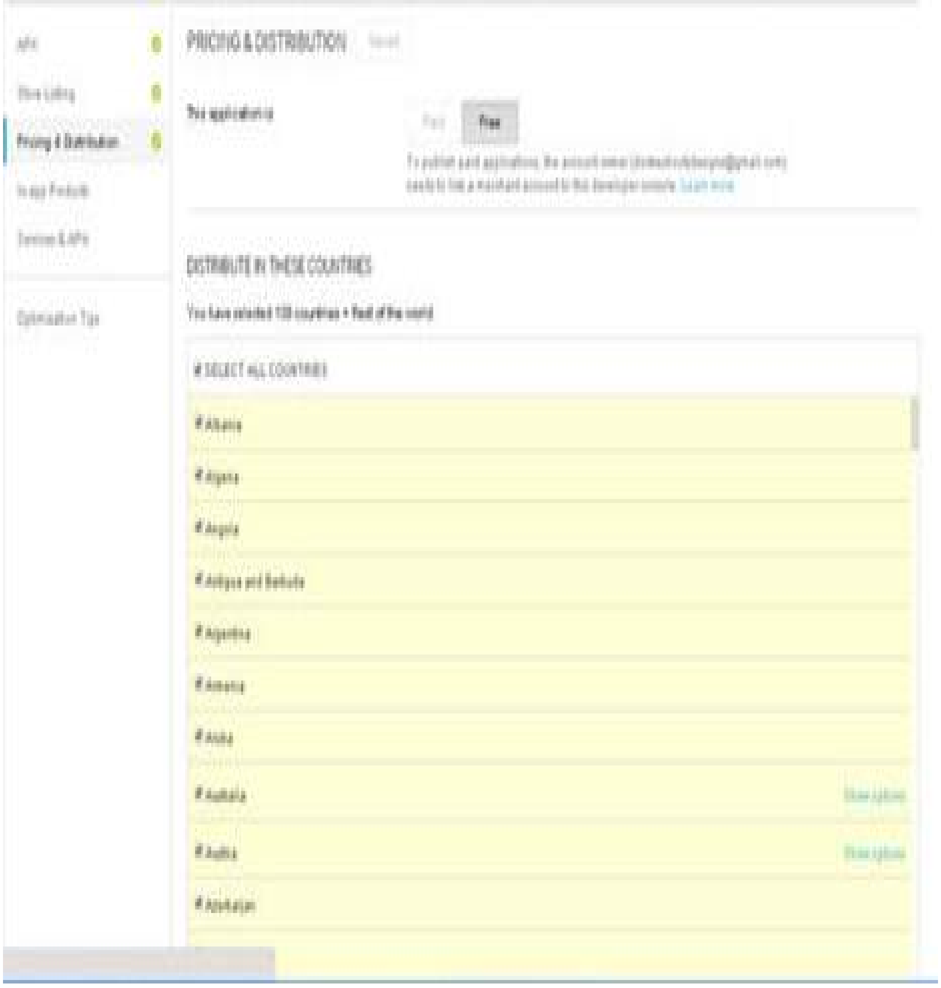


Figure 14.6 : Pricing and Distribution

11) After all the steps are completed, we can select “Publish this app” which publishes the app to Google play store and it will be live in few hours.

14.3 APPLE APP STORE

App Store is the official app distribution platform developed by Apple. It hosts all apps for Apple devices such as iPhone, iPad, Apple TV, Apple watch etc. As on 2017 Apple app store has more than 2.2 million apps.

14.3.1 Prerequisites for Publishing Apps to Apple App Store

The following are the prerequisites for publishing apps to Apple app store:

- Register at iTunes connect at https://itunesconnect.apple.com
- Register at provisioning portal at https://developer.apple.com/iphone

The following are high level steps for distributing iOS apps to Apple app store:

1) Register as iOS and Apple Developer
2) Create profile, certificate and register devices
3) Submit app for approval

Its possible to register as Apple Developer for free. Then, you can enroll for paid iOS developer program. Its possible to enroll as individual or as a company.

The following is the process for creating a profile, certificate and register devices:

All apps downloaded from Apple app store have signed Apple certificate. Apple devices verify the signature before executing the app.

Apple provides iOS provisioning portal for performing these activities. The portal creates “profiles”, “code signing entities” that enable Xcode to sign the apps so that Apple devices can securely verify the apps’ authenticity. There are mainly two kinds of profile: development profile (that restrict the devices on which the app runs) and distribution profile (wherein Apple has to sign the apps). Details of distribution certificate are available at https://developer.apple.com/ios/manage/distribution/index.action

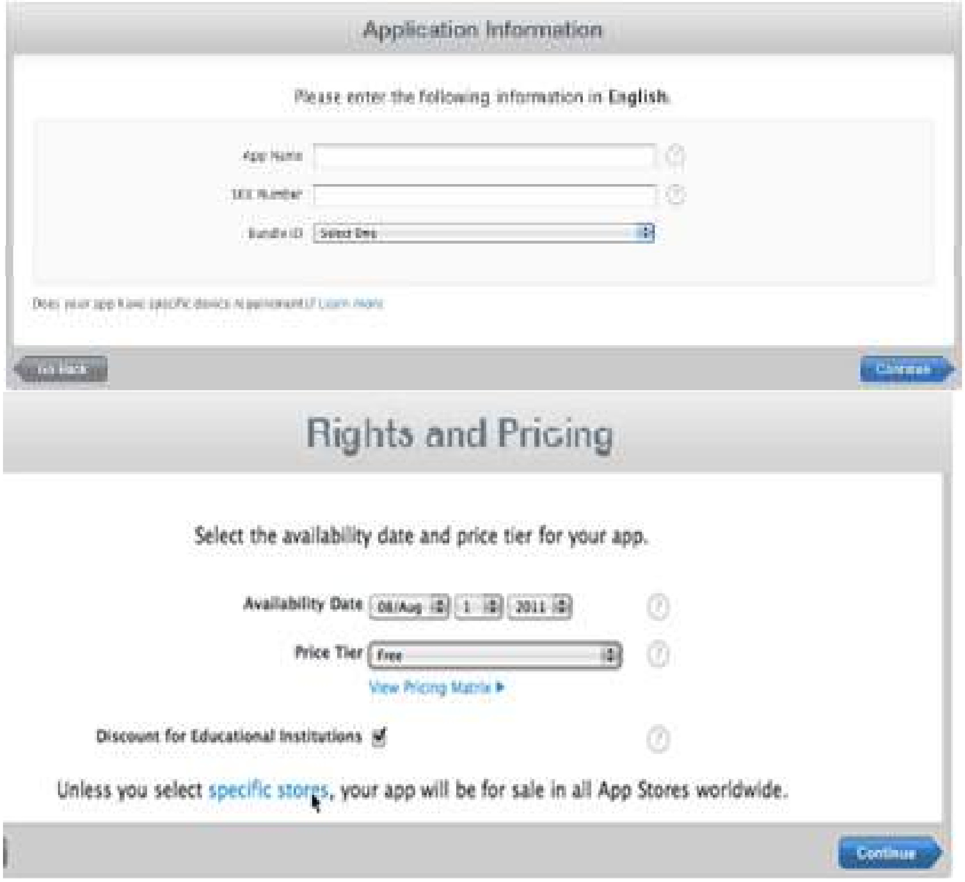
Provisioning portal is used for creating certification profiles and for registering the devices.

The following are high level steps involved in this stage:

1) Create and download the certificates for development and distribution.
2) Using the Unique Device Identifier (UDID), register the Apple device.
3) Create the device-specific unique App Id.
4) Create development and distribution profile using the profile name, App id, certificate and device.

The following are the steps to submit app to the app store:

1) Open the URL https://itunesconnect.apple.com and sign digital contracts.
2) Fill up contact details, bank information and tax forms.
3) Provide following details for the app (Figure 14.7).
   a) App’s Name, Description, app image icon (512px by 512px), at least one screenshot
   b) Company name
   c) Rights
   d) Availability date
   e) Price (paid or free)



Application Information

Please enter the following information in **English**.

App Name

SKU Number

Bundle ID: Select One

Does your app have specific device requirements? Learn more

Rights and Pricing

Select the availability date and price tier for your app.

Availability Date: 08/Aug 1 2011

Price Tier: Free

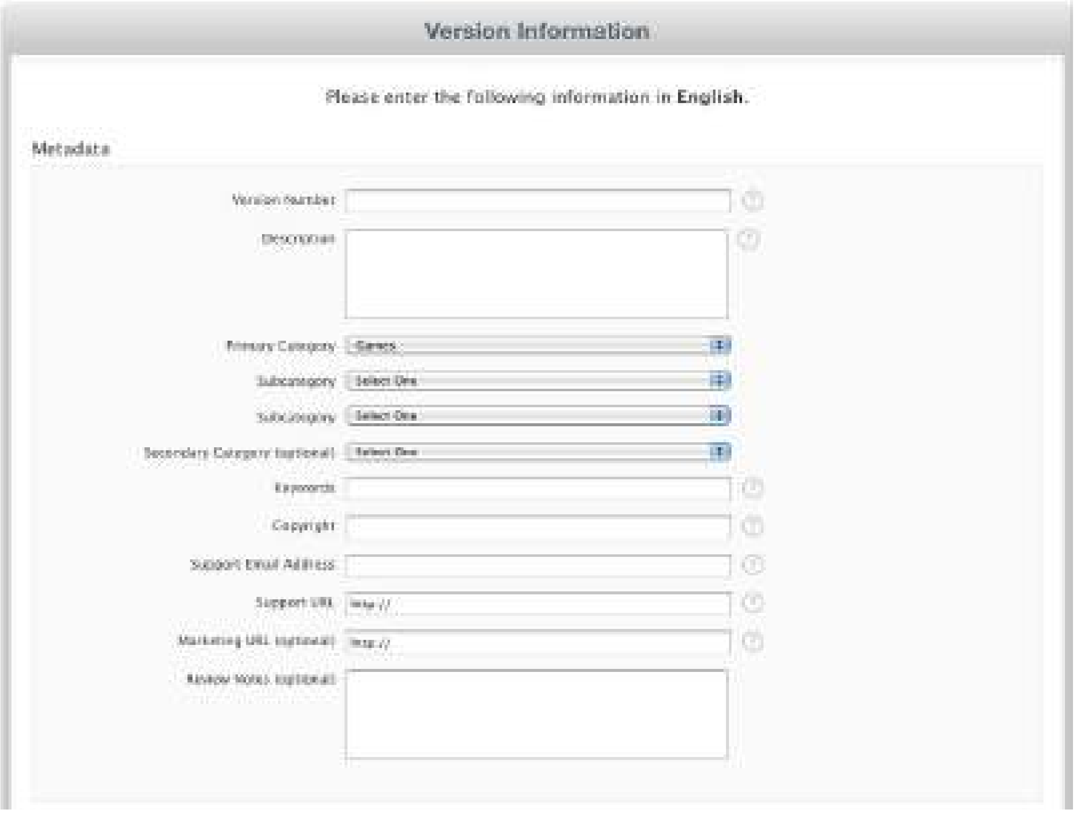
View Pricing Matrix ▸

Discount for Educational Institutions ☑

Unless you select specific stores, your app will be for sale in all App Stores worldwide.

Figure 14.7 : Rights & Pricing

2) Provide the app’s version information that includes app description, keywords, primary description, secondary description, copyright, support email address, and support URL, marketing URL, review notes and such (Figure 14.8).



Version Information

Please enter the following information in **English**.

Metadata

Version Number

Description

Primary Category: Games

Subcategory: Select One

Subcategory: Select One

Secondary Category (optional): Select One

Keywords

Copyright

Support Email Address

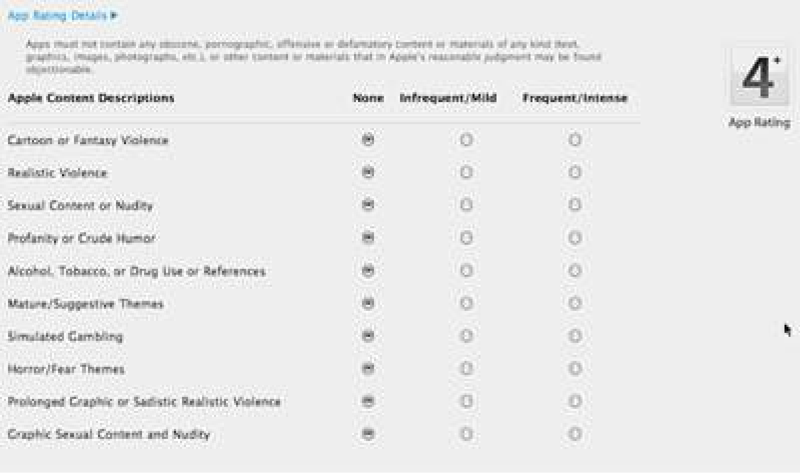
Support URL: http://

Marketing URL (optional): http://

Review Notes (optional)

Figure 14.8 : App version information

3) Provide the app rating details as shown in Figure 14.9



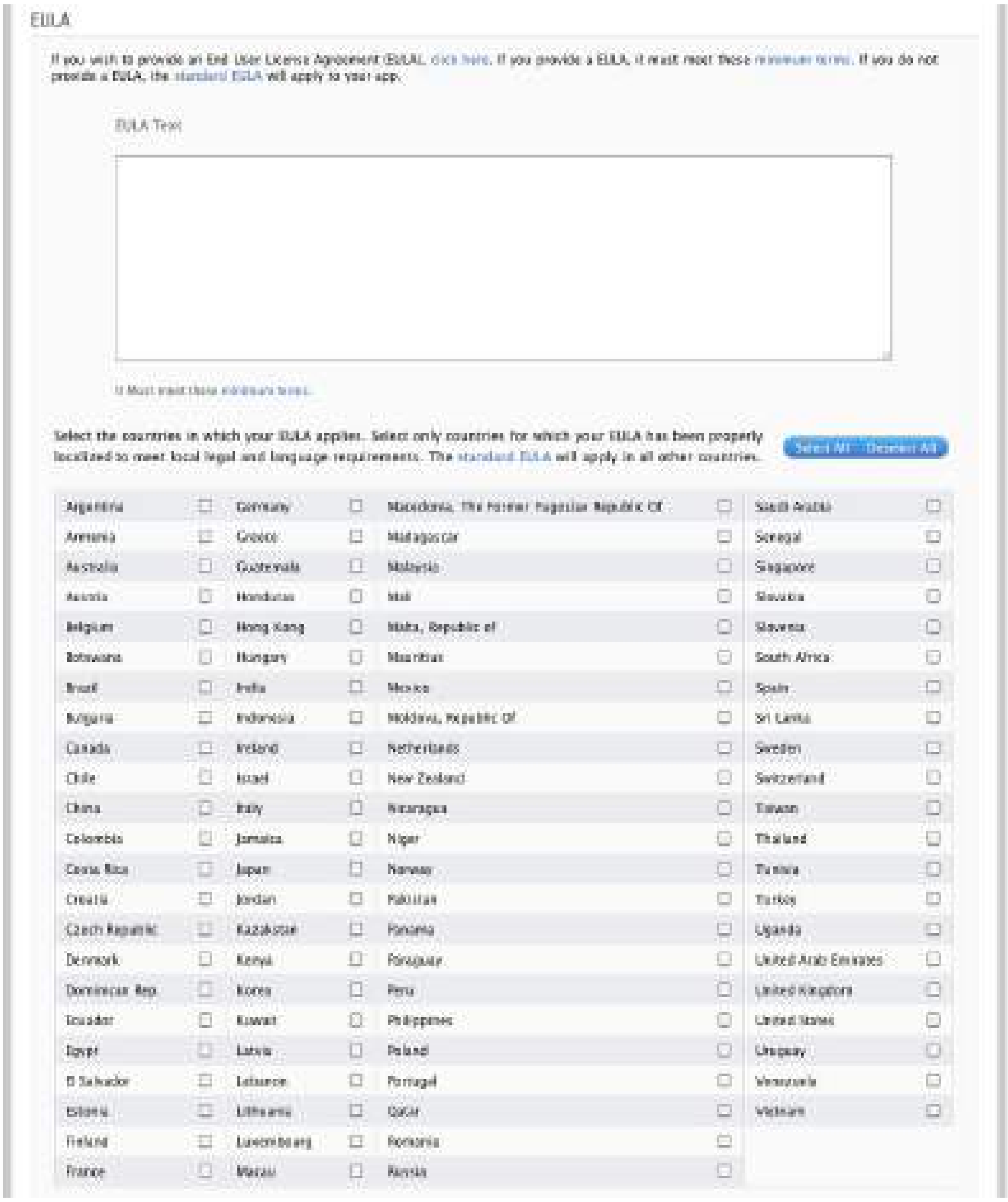
App Rating Details ▸

Apps must not contain any obscene, pornographic, offensive or defamatory content or materials of any kind (text, graphics, images, photographs, etc.), or other content or materials that in Apple's reasonable judgment may be found objectionable.

Apple Content Descriptions	None	Infrequent/Mild	Frequent/Intense
Cartoon or Fantasy Violence	●	○	○
Realistic Violence	●	○	○
Sexual Content or Nudity	●	○	○
Profanity or Crude Humor	●	○	○
Alcohol, Tobacco, or Drug Use or References	●	○	○
Mature/Suggestive Themes	●	○	○
Simulated Gambling	●	○	○
Horror/Fear Themes	●	○	○
Prolonged Graphic or Sadistic Realistic Violence	●	○	○
Graphic Sexual Content and Nudity	●	○	○

Figure 14.9 : App rating information

4) Provide the End User License Agreement (EULA) information (Figure 14.10)



EULA

If you wish to provide an End User License Agreement (EULA), click here. If you provide a EULA, it must meet these minimum terms. If you do not provide a EULA, the standard EULA will apply to your app.

EULA Text

It Must meet these minimum terms.

Select the countries in which your EULA applies. Select only countries for which your EULA has been properly localized to meet local legal and language requirements. The standard EULA will apply in all other countries.

Argentina	□	Germany	□	Macedonia, The Former Yugoslav Republic Of	□	Saudi Arabia	□
Armenia	□	Greece	□	Madagascar	□	Senegal	□
Australia	□	Guatemala	□	Malaysia	□	Singapore	□
Austria	□	Honduras	□	Mali	□	Slovakia	□
Belgium	□	Hong Kong	□	Malta, Republic of	□	Slovenia	□
Botswana	□	Hungary	□	Mauritius	□	South Africa	□
Brazil	□	India	□	Mexico	□	Spain	□
Bulgaria	□	Indonesia	□	Moldova, Republic Of	□	Sri Lanka	□
Canada	□	Ireland	□	Netherlands	□	Sweden	□
Chile	□	Israel	□	New Zealand	□	Switzerland	□
China	□	Italy	□	Nicaragua	□	Taiwan	□
Colombia	□	Jamaica	□	Niger	□	Thailand	□
Costa Rica	□	Japan	□	Norway	□	Tunisia	□
Croatia	□	Jordan	□	Pakistan	□	Turkey	□
Czech Republic	□	Kazakhstan	□	Panama	□	Uganda	□
Denmark	□	Kenya	□	Paraguay	□	United Arab Emirates	□
Dominican Rep.	□	Korea	□	Peru	□	United Kingdom	□
Ecuador	□	Kuwait	□	Philippines	□	United States	□
Egypt	□	Latvia	□	Poland	□	Uruguay	□
El Salvador	□	Lebanon	□	Portugal	□	Venezuela	□
Estonia	□	Lithuania	□	Qatar	□	Vietnam	□
Finland	□	Luxembourg	□	Romania	□		
France	□	Macau	□	Russia	□		

Figure 14.10: App EULA information

5) Provide various app screenshots (Figure 14.11).

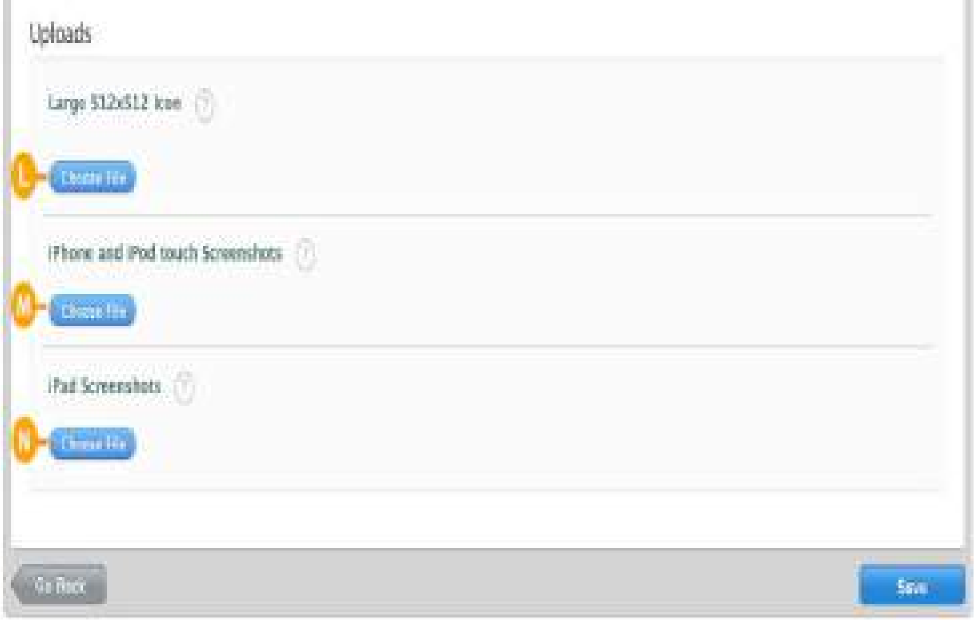


Figure 14.11 : App screenshots

6) After all the details are filled, app can be uploaded.

7) Apple reviews the app and upon successful review, it will be available in the app store.

☞ **Check Your Progress 1**

1) _____________ file is the packaged Android app file that can be deployed to Google Play store.

2) ______________section in Google play store provides fields for specifying the price details of the Android app.

3) _____________ information of the app is used for registering the Apple device.

14.4 WINDOWS STORE

Windows store is the official distribution channel for certified windows apps. Microsoft revamped the UI and brought the concept of apps from Windows 8 onwards to Microsoft devices. Developers can deploy, distribute and sell apps from Windows store. Users can discover apps belonging to various categories from Windows store. Visual Studio is used for developing Windows apps.

Windows 8 apps use various intuitive features such as full screen page layout, touch enabled features, communication with other apps for seamless user experience. Most of the apps uses tiles layout which can be used for pushing updates and live alerts.

Windows apps can be created either by any of the following groups of technologies:

- HTML 5, CSS 3 and JavaScript
- C#/Visual Basic/C++ and XAML

14.4.1 Pre-requisites for Publishing Apps to Windows Store

The following are the prerequisites for publishing apps to Windows store:

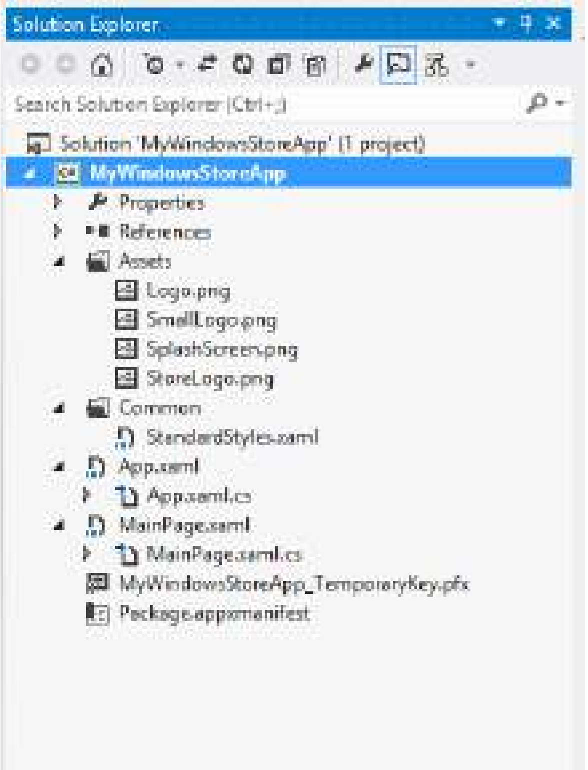
- Windows developer account should be created at https://appdev.microsoft.com/StorePortals/en-us/Account/Signup/Start
- License agreements should be accepted and the subscription fee should be paid.

The following are some of the best practices that would help Windows app to get certification:

- It should adhere to the Microsoft's UX guidelines. MS Visual Studio provides in built templates that conform to the specified UX guidelines.
- It should adhere to Microsoft's content policy. The advertisement should not interfere with end user experience.
- It should publish its data and content privacy policy.
- The launch time of 5 seconds or lesser and suspend time of 2 seconds or lesser should be adhered to.
- It functions should be accessible when the resolution is 1024 x 768 or above.
- It should provide technical support information.
- Windows App Certification Kit can be used to test the compliance level of the app.

The following are for publishing apps to Windows store:

1) Develop the app using any of the technologies mentioned earlier. A sample app project is shown in Figure 14.12.



```
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'MyWindowsStoreApp' (1 project)
  MyWindowsStoreApp
    Properties
    References
    Assets
      Logo.png
      SmallLogo.png
      SplashScreen.png
      StoreLogo.png
    Common
      StandardStyles.xaml
    App.xaml
      App.xaml.cs
    MainPage.xaml
      MainPage.xaml.cs
    MyWindowsStoreApp_TemporaryKey.pfx
    Package.appxmanifest
```

Figure 14.12: Windows app project

2) Prepare the app for publication. Provide all the app details such as app description, display name, app capabilities (such as access requirements for contacts, gallery, GeoLocator etc.), version, declarations, title images, logos (including small logo, wide logo, store logo, badge logo) as shown in Figure 14.13.

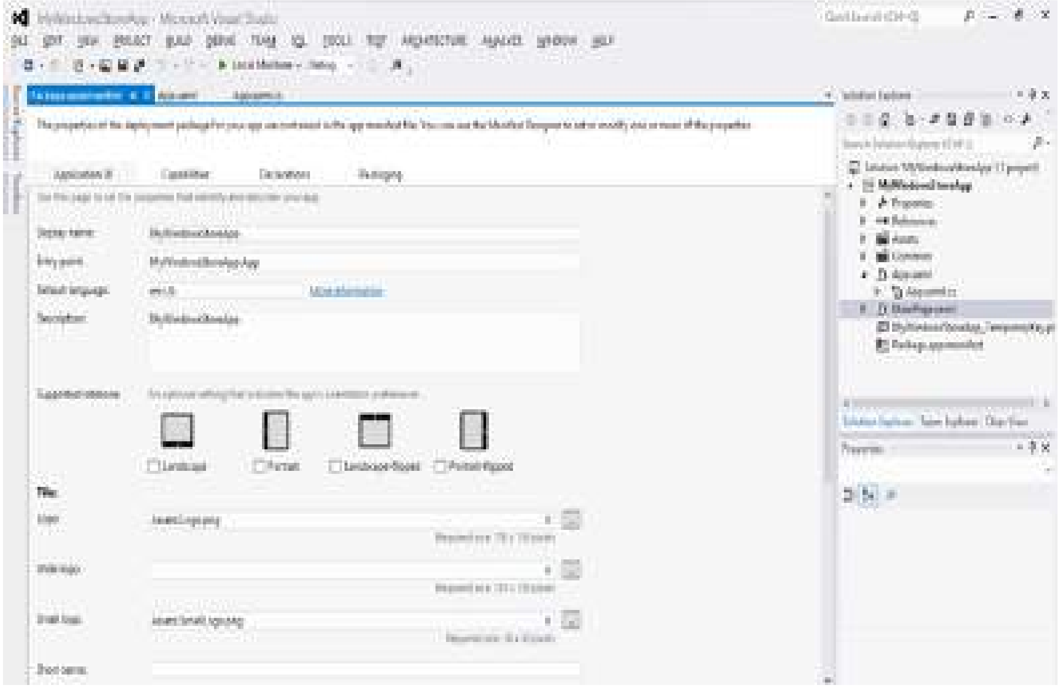
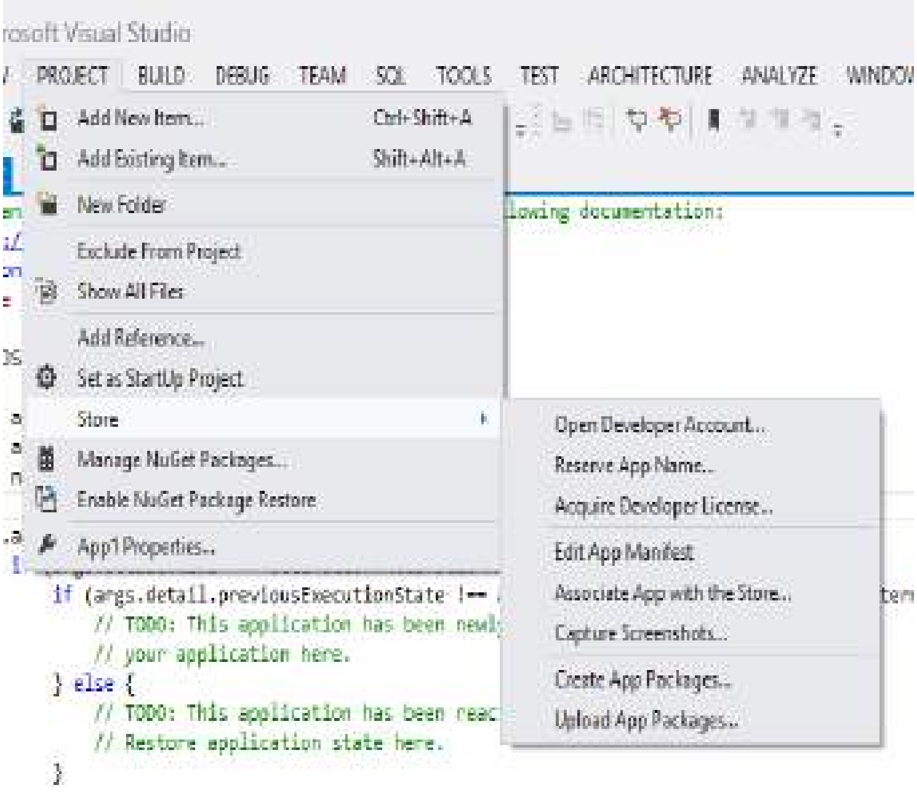


Figure 14.13: Configuring Windows app

3) Create the app package using project → store → create app package. While packaging the app, select the option to package the app for Windows store as shown in Figure 14.14.



```
llowing documentation:

    if (args.detail.previousExecutionState !==
        // TODO: This application has been newl
        // your application here.
    } else {
        // TODO: This application has been reac
        // Restore application state here.
    }
```

Figure 14.14: Creating a detailed app profile

4) Execute App certification SDK to test the technical compliance of the app to Microsoft guidelines.

5) Create a detailed app profile through project → store → Reserve app name or through online at http://go.microsoft.com/fwlink/?LinkId=256672&clcid=0x409

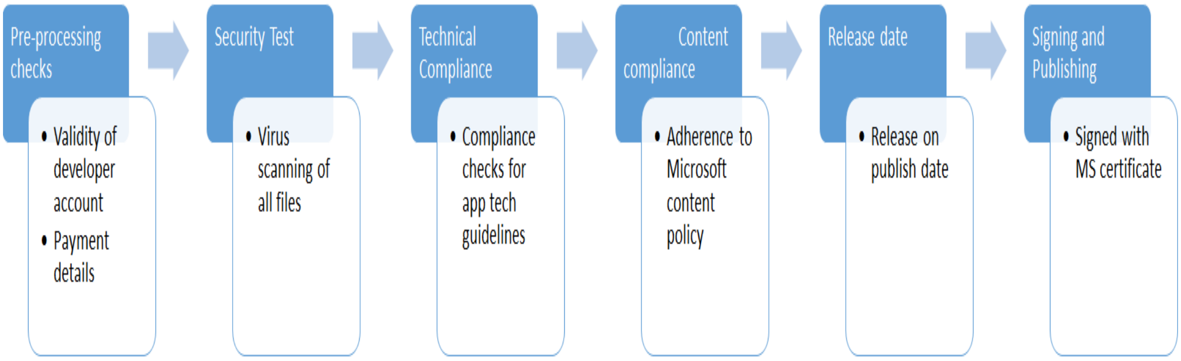
   In this step, specify the unique name for the app, age rating, app description, payment details (free or paid), distribution region and other details (Figure 14.14).

6) Upload the app package through project → store → upload app package or through URL http://go.microsoft.com/fwlink/?LinkId=256671&clcid=0x409. This submits the app to Microsoft for certification.

7) Developer can check the details such as downloads, ratings, reviews, conversion, usage details, payment etc. at http://go.microsoft.com/fwlink/p/?LinkId=223974.

After the app is submitted to the Windows store, developers can view the certification status of the app at http://go.microsoft.com/fwlink/p/?LinkId=223974.

The high level app certification steps are specified in Figure 14.15.



Pre-processing checks
- Validity of developer account
- Payment details

Security Test
- Virus scanning of all files

Technical Compliance
- Compliance checks for app tech guidelines

Content compliance
- Adherence to Microsoft content policy

Release date
- Release on publish date

Signing and Publishing
- Signed with MS certificate

Figure 14.15: Windows app certification steps

☞ **Check Your Progress 2**

1) The two development technologies for windows app are ____________ and __________________ .

2) ______________________ can be used to test the technical compliance for the windows app.

3) Content compliance check includes checking the adherence of app content to ____________________ .

14.5 SUMMARY

In this unit, we discussed various mobile app stores that can be used for deployment and distribution of mobile apps. The pre-requisites and publishing steps for Android apps in Google Play store, iOS apps in Apple app store and

Windows apps in Windows store are explained. We also discussed some best practices for Windows app development so that it gets certification.

14.6 SOLUTIONS/ANSWERS

Check Your Progress 1

1) .apk

2) Pricing and distribution

3) UDID

Check Your Progress 2

1) HTML 5, CSS 3 and JavaScript and C#/Visual Basic/C++ and XAML

2) App certification SDK

3) Microsoft content policy

14.7 FURTHER READINGS

References

1) http://developer.android.com/
2) http://developer.android.com/guide/publishing/publishing.html#market
3) http://developer.android.com/guide/publishing/app-signing.html
4) http://developer.android.com/guide/publishing/preparing.html
5) http://developer.apple.com/
6) https://developer.apple.com/support/ios
7) https://itunesconnect.apple.com
8) https://developer.apple.com/iphone
9) http://www.windowsstore.com/developers/windows
10) http://msdn.microsoft.com/library/windows/apps/br211361.aspx
11) http://msdn.microsoft.com/en-us/library/windows/apps/hh974581.aspx
12) http://msdn.microsoft.com/en-us/library/windows/apps/hh974578.aspx
13) http://msdn.microsoft.com/en-us/library/windows/apps/jj657972.aspx
14) http://msdn.microsoft.com/en-us/library/windows/apps/hh694062.aspx
15) http://msdn.microsoft.com/en-US/windows/apps/br229512
16) http://msdn.microsoft.com/en-us/library/windows/apps/hh921583.aspx
17) http://msdn.microsoft.com/en-us/library/windows/apps/hh694083.aspx

UNIT 15 MONETIZATION

Structure

15.0 INTRODUCTION

One of the key success factors for mobile apps may be overall revenue generated from the app apart from number of downloads etc. As app development needs significant investment and planning, monetization and revenue generation plans form part of fundamental aspects of mobile app strategy. We need to carefully explore all avenues of revenue generation and chose the most optimal one that aligns with overall business strategy.

In this unit,we will look at various avenues for monetization through mobile apps. We have detailed various ways to generate revenue from Android apps, iOS apps and Windows apps.

15.1 OBJECTIVES

After going through this unit, you should be able to understand:

- various monetization ways for Android apps such as free model, paid model, freemium model, affiliate model, commission model, white labeling model and publishing by Amazon app store;
- various ways to monetize iOS apps such as services model, sponsorship model, and Facebook ads model;
- various ways for monetizing Windows apps such as affiliate ads, community ads, and donation model; and

15.2 MONETIZING ANDROID APPS

Android platform provides great opportunity for monetization as it is one of the most popular mobile platforms. The following are some of the popular ways in which Android app developers can derive revenue out of apps:

15.2.1 Free Distribution Model with Ads

Offering Android apps for free is one of the popular ways to monetize the app. Free version of the apps feature advertisements (ads). About half of the available Android apps feature advertisements. App developers would derive revenue based on the click-through rate (CTR). CTR basically is the ratio of the ads users clicked to total number of displayed ads. For each ad click, the developer would get a specific amount. There are other events wherein the advertiser pays the developer such as ad view, ad click, ad conversion, ad events and such. Size and placement of ads also play an important role in CTR. Full screen ads have high CTR but they could decrease the popularity of the app. Normally full screen ads are displayed in game apps when user jumps from one level to another. Other ways to increase the CTR through ads are given below:

- Include rich media content (such as video content) in the ads
- Display the relevant ads relevant for the user’s context and based on user’s history of transactions.
- Include interactive content such as questions, teasers in the ad text. Provide the incentive for the user through ads.
- The ads must be non-intrusive and should not interfere or severely impact user app experience.

AdMob from Google and InMobi are some of the popular mobile ad platforms.

15.2.2 Freemium Distribution Model

Freemium is a free app with some premium content containing in-app purchases. In a typical gaming app, though the app is free, some of the key essentials needed for the full gaming experience (such as currency, higher levels, unlocking a feature, extended features, virtual coins etc.) are paid entities. Users need to purchase within the app (termed as in-app purchase) to access the premium content. This is also a popular model to monetize the app. Normally, freemium apps need more programming effort and maintenance effort. The following are some of the key points for designing an effective freemium app:

- Market the app so that it is popular in its category.
- Identify the key essentials of app and provide premium access to it.
- Decide a one-time purchase or a recurring subscription purchase option.

15.2.3 Paid Distribution Model

End users need to pay for these apps while purchasing from Google play store. App developers can fix the price and the distribution region. The following are some of the best practices for designing a successful paid app:

- Market the app in all the distributed regions.

- Release the demo app which is a limited free version of the app to generate interest among the users.
- Design a compelling user experience.
- Start with a free app and gauge the user feedback and response and then launch a paid app once the free app reaches critical mass.
- Promote the app through social networking sites, emails and SMS.

15.2.4 Affiliate Partnership Model

The apps which have lot of content would become affiliates to derive revenues. Affiliate ads drive the user traffic to ad sites. With a strong content strategy, the app can act as a brand marketing channel for the products and services.

15.2.5 Commission Model

If apps form the medium of transaction, app developers would receive a commission for each transaction that happens through that app. The marketplace apps, retail apps, e-commerce apps which bring buyers and sellers to the same platform are typical examples which work in this model.

15.2.6 White Labeling Model

App developers can sell their source code to various companies with similar needs. Companies can reskin and rebrand the app for their own needs after purchasing the app's source code.

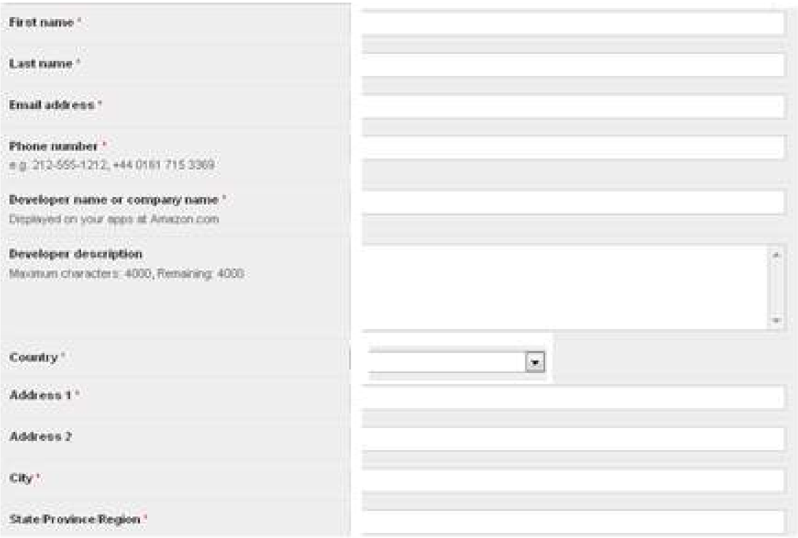
15.2.7 Publishing to Amazon App Store

Android apps can also be submitted to Amazon app store. This provides wider reach to Amazon customers and better app governance provided by Amazon. This would further help to gain additional revenue from Android Apps.

The following are high level steps for distributing Android apps from Amazon app store are given below:

1) Create a developer account at https://developer.amazon.com (Figure 15.1).

2) After successful registration and login, provide all the details such as profile information (name, address, company name etc.), customer support email etc. (Figure 15.2).



First name *

Last name *

Email address *

Phone number *
e.g. 212-555-1212, +44 0161 715 3369

Developer name or company name *
Displayed on your apps at Amazon.com

Developer description
Maximum characters: 4000, Remaining: 4000

Country *

Address 1 *

Address 2

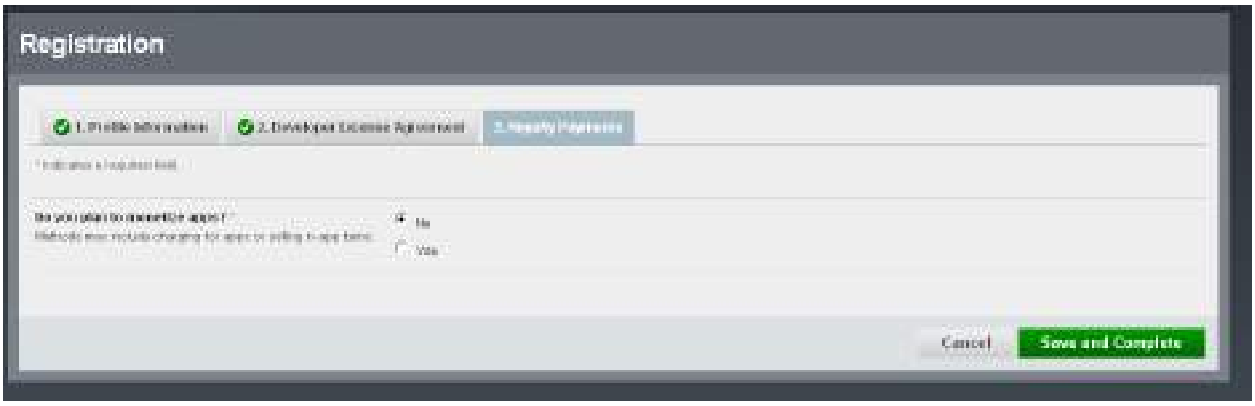
City *

State/Province/Region *

Figure 15.1 : Account creation at http://www.amazon.com

3) Accept developer license agreement and print the agreement.

4) Provide the royalty payment details



Registration

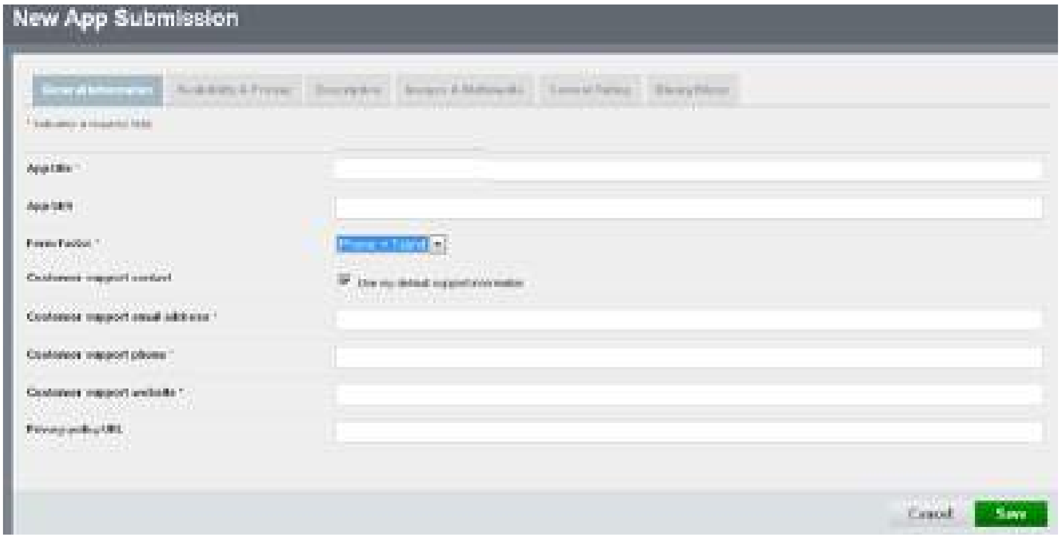
✓ 1. Profile Information | ✓ 2. Developer License Agreement | 3. Royalty Payments

* Indicates a required field

Do you plan to monetize apps? *
Methods may include charging for apps or selling in-app items.
● No
○ Yes

Figure 15.2 : Developer registration at Amazon

5) Add the app details such as app tile, form factor, support details, availability and pricing details etc. (Figure 15.3 and Figure 15.4)



New App Submission

General Information | Availability & Pricing | Description | Images & Multimedia | Content Rating | Binary File(s)

* Indicates a required field

App title *

App SKU

Form Factor * Phone + Tablet

Customer support contact ☑ Use my default support information

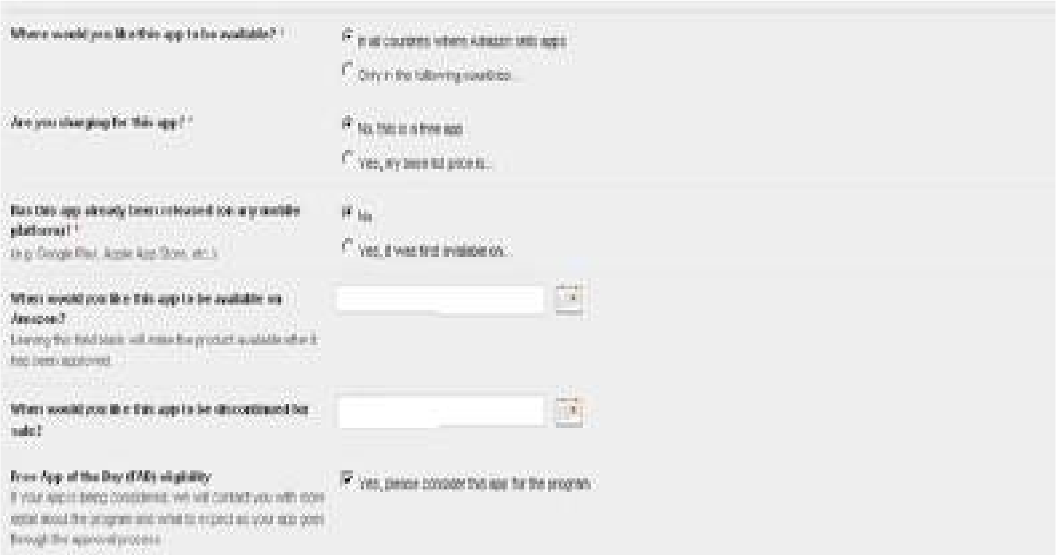
Customer support email address *

Customer support phone *

Customer support website *

Privacy policy URL

Figure 15.3 : New app submission



Where would you like this app to be available? *
● In all countries where Amazon sells apps
○ Only in the following countries...

Are you charging for this app? *
● No, this is a free app
○ Yes, my base list price is...

Has this app already been released on any mobile platforms? *
(e.g. Google Play, Apple App Store, etc.)
● No
○ Yes, it was first available on...

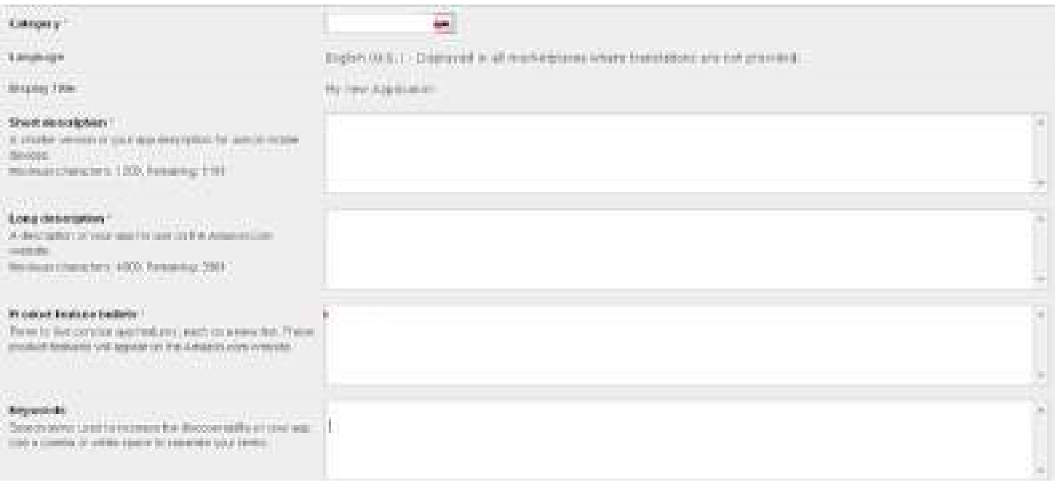
When would you like this app to be available on Amazon?
Leaving this field blank will make the product available after it has been approved.

When would you like this app to be discontinued for sale?

Free App of the Day (FAD) eligibility
If your app is being considered, we will contact you with more detail about the program and what to expect as your app goes through the approval process.
☑ Yes, please consider this app for the program

Figure 15.4 : Details about app

6) Provide additional app details such as category, language, short description, long description, keywords etc.(Figure 15.5).



Category *

Language — English (U.S.) - Displayed in all marketplaces where translations are not provided

Display Title — My New Application

Short description *
[illegible]
Maximum characters: 1200, Remaining: [illegible]

Long description *
[illegible]
Maximum characters: 4000, Remaining: [illegible]

Product feature bullets *
[illegible]

Keywords
[illegible]

Figure 15.5 : Description of app

7) Provide the app screenshots and content rating and then, submit the app binary.

15.3 MONETIZING IOS APPS

Most of the monetization techniques that were discussed for Android apps such as free distribution model, freemium model, paid module, affiliate model, commission model are applicable for iOS apps as well.

The following are some more monetization techniques for iOS apps:

Apps for mobile-first strategy

iOS apps can be used to implement the mobile-first strategy. Apps can be used to showcase and promote the company’s products and services. E-commerce apps can be used to sell the products, services and merchandise.

Revenue through services

If the app has valuable data or business logic, the key features of the app can be exposed as services for third party consumers. Apps can expose analytics functions, demographics data, user interests (subject to privacy policies), and geo-location data as services.

Sponsorship apps

App developers can secure the sponsorship for their apps based on the ad content, themes, events, services and topics. For instance, a movie app can secure the sponsorship for promoting a recently released movie from distributors.

Using Facebook ads

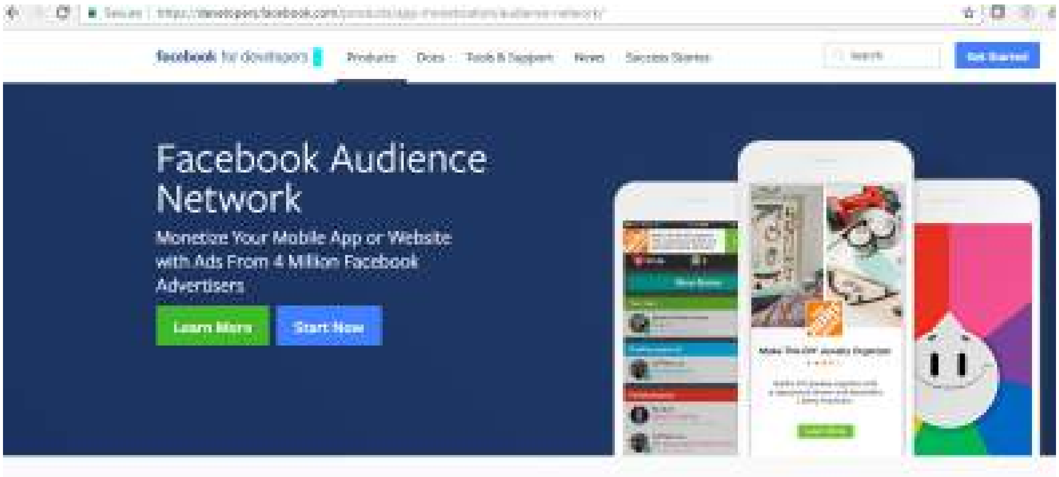
iOS developers can leverage the “Facebook audience network” to monetize their iOS (and Android) apps through Facebook ads. Developers can implement the Facebook’s audience network feature in their apps. When the end user is using the app, unless the user has chosen “opt out feature”, or if the corresponding user does not exist in Facebook, an ad will be shown.The user information is mapped to the Facebook user information for displaying ads.

Primarily, there are three types of Facebook ads which are explained below:

- **Banner ads**: These ads can be place anywhere in the app. Developers can also set the refresh time for this ad. The size of banner ads is restricted to max height of 50 px for mobile and 90 px for table with configurable width.
- **Interstitial ads**: These are full screen ads normally used for games.
- **Native ads**: Developers get flexibility with these ads in terms of location, look,feel and size. The ad title should contain maximum of 30 characters with cover image of 1200 x 627 px and the body text is restricted to 90 characters.

The following are the detailed steps for implementing Facebook ads in iOS apps:

1) Go to https://developers.facebook.com/products/app-monetization/audience- network/ and apply for the audience network (Figure 15.5)

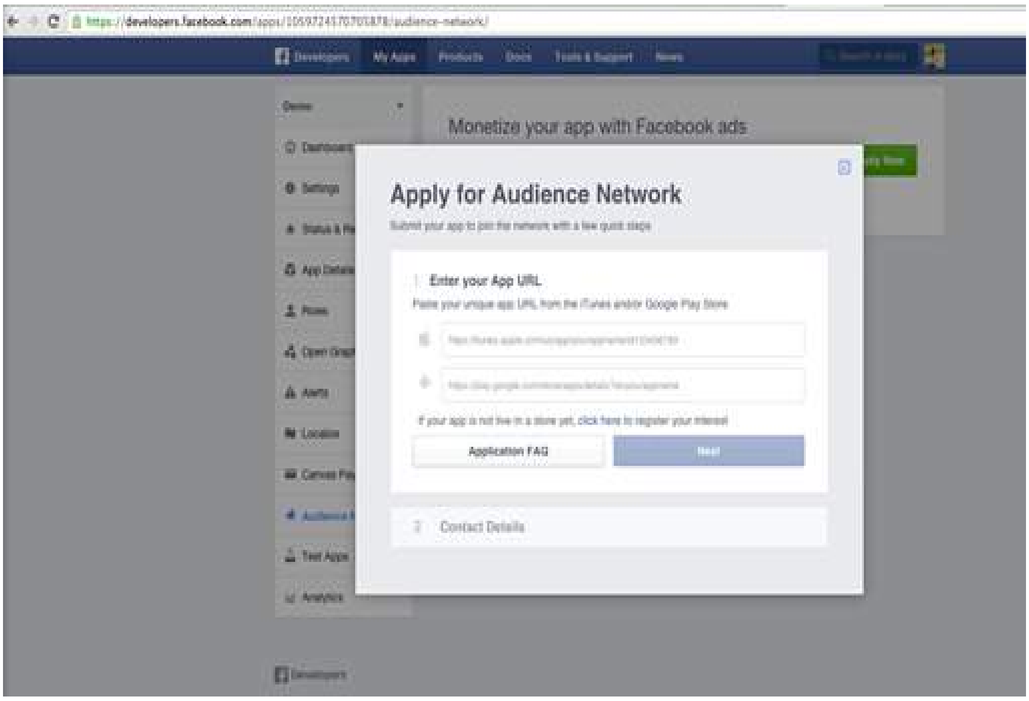


Facebook Audience Network

Monetize Your Mobile App or Website with Ads From 4 Million Facebook Advertisers

Figure 15.5 : Facebook Audience Network

2) Provide the app details and user details to apply for Facebook audience network (Figure 15.6)



https://developers.facebook.com/apps/

Monetize your app with Facebook ads

Apply for Audience Network

Submit your app to join the network with a few quick steps

1 Enter your App URL

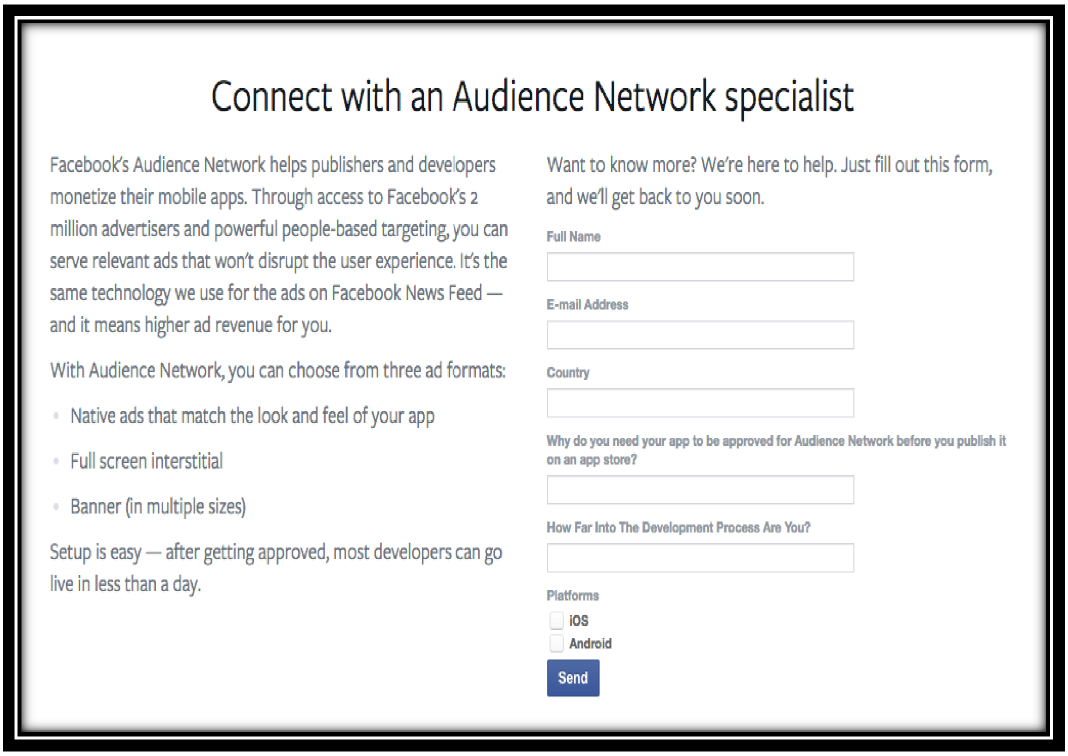
Paste your unique app URL from the iTunes and/or Google Play Store

If your app is not live in a store yet, click here to register your interest

2 Contact Details

Figure 15.6: Applying for audience network

3) Then, connect with an Audience network specialist (Figure 15.7).



Connect with an Audience Network specialist

Facebook's Audience Network helps publishers and developers monetize their mobile apps. Through access to Facebook's 2 million advertisers and powerful people-based targeting, you can serve relevant ads that won't disrupt the user experience. It's the same technology we use for the ads on Facebook News Feed — and it means higher ad revenue for you.

With Audience Network, you can choose from three ad formats:

- Native ads that match the look and feel of your app
- Full screen interstitial
- Banner (in multiple sizes)

Setup is easy — after getting approved, most developers can go live in less than a day.

Want to know more? We're here to help. Just fill out this form, and we'll get back to you soon.

Full Name

E-mail Address

Country

Why do you need your app to be approved for Audience Network before you publish it on an app store?

How Far Into The Development Process Are You?

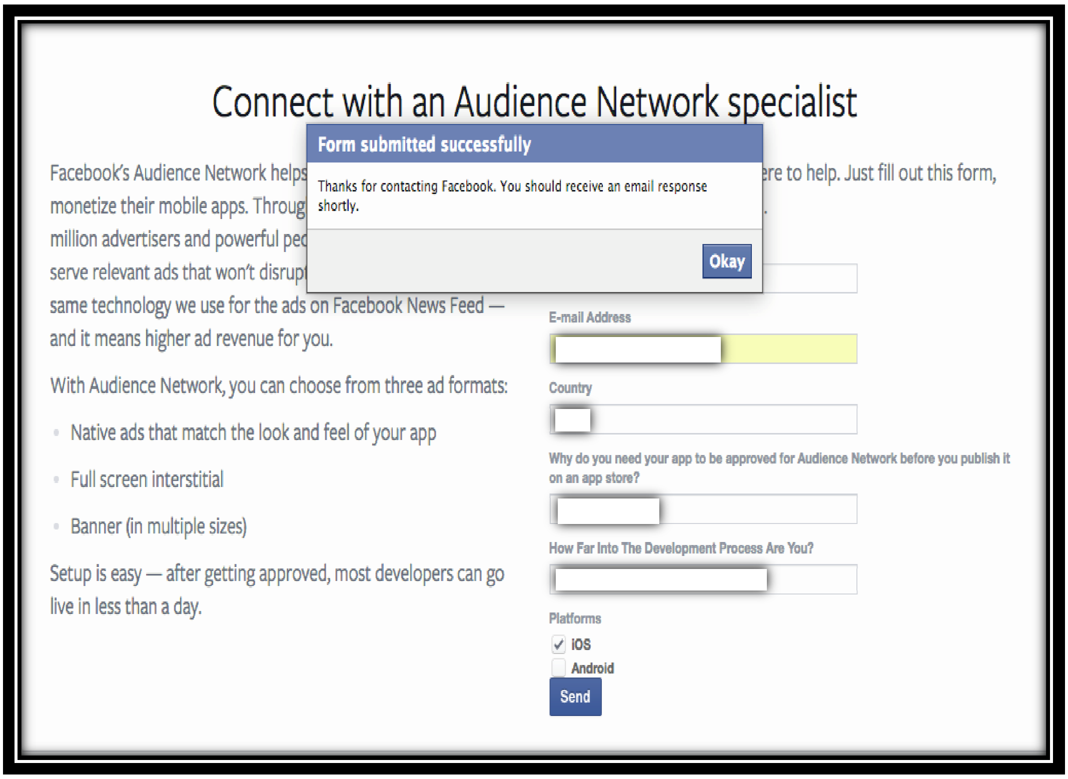
Platforms

□ iOS

□ Android

Figure 15.7 : Connecting with Facebook Audience Network Specialist

4) After submitting all the details, the submission is acknowledged as shown in Figure 15.8.



Connect with an Audience Network specialist

Form submitted successfully

Thanks for contacting Facebook. You should receive an email response shortly.

Facebook's Audience Network helps
monetize their mobile apps. Throug
million advertisers and powerful peo
serve relevant ads that won't disrup
same technology we use for the ads on Facebook News Feed — and it means higher ad revenue for you.

re to help. Just fill out this form,

With Audience Network, you can choose from three ad formats:

- Native ads that match the look and feel of your app
- Full screen interstitial
- Banner (in multiple sizes)

Setup is easy — after getting approved, most developers can go live in less than a day.

E-mail Address

Country

Why do you need your app to be approved for Audience Network before you publish it on an app store?

How Far Into The Development Process Are You?

Platforms

☑ iOS

□ Android

Figure 15.8 : Acknowledging the submission

5) After registration, the app developer can access the audience network section (Figure 15.9).

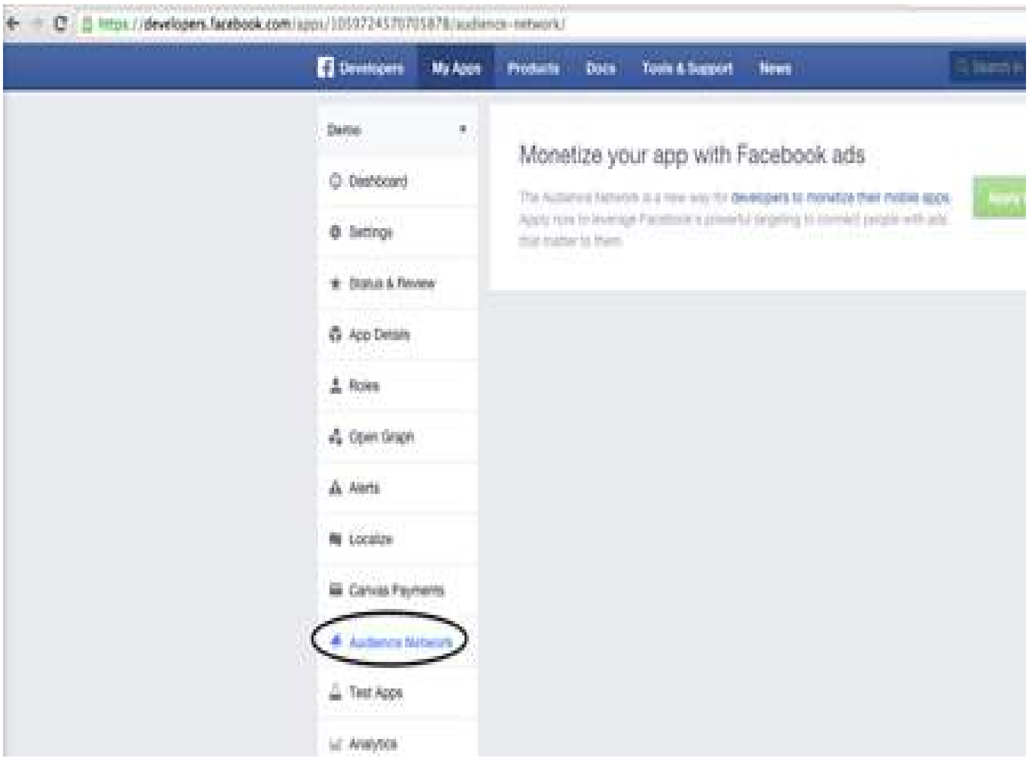


Figure 15.9 : Accessing the Facebook Audience Network

6) Download and include the "Audience network library" which is part of Facebook iOS SDK. The FBAudienceNetwork.framework , AdSupport, StoreKit and CoreMotion can be added to Xcode project (Figure 15.10).

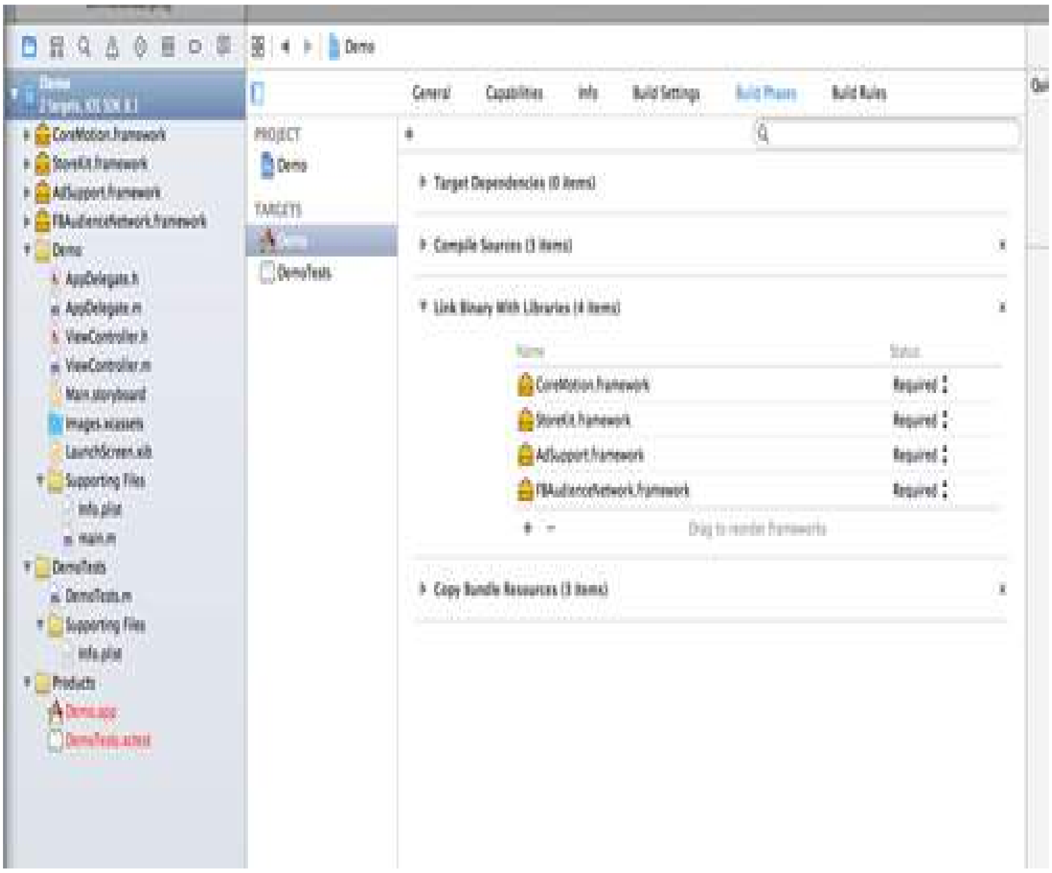


Figure 15.10 : Audience Network Library

7) Use placement id that identifies each Facebook ad unit and add the ad units into app using the APIs.

☞ **Check Your Progress 1**

1) CTR is the ratio of the ads ___________ to ______________.

2) Marketplace apps, retail apps, e-commerce apps are example of ___ model.

3) Selling app source to other companies is done in ___________ model.

4) Exposing key app functionality as ______ is a monetization model.

5) The ads that are placed at top or bottom of the screen are called ____________________.

6) _________________ ads occupy full screens.

7) _________________ library in Facebook iOS SDK is used for ads.

15.4 MONETIZING WINDOWS APPS

Paid, free, freemium, white labeling monetization models discussed earlier can be used for Windows apps as well. In addition, the following are other popular monetization techniques for Windows apps:

Affiliate ads from Microsoft

Windows app developers can leverage the Microsoft affiliate ads to show promotions for music, games from Windows store. Based on the conversion rate, commission will be paid to the app developers.

Ads from Community

Windows app developers can select this option to get ads published from other developer communities.

Donation model

Not-for-profit and charity companies normally design apps that seek donations from people. They can post the app to Windows store.

15.5 CHOOSING RIGHT MOBILE AD STRATEGY

As ads play role in app monetization, let us look at some of the key concepts in this regard.

15.5.1 Defining the Success Metrics

The first stage is to define the Key Performance Indicators (KPI) and metrics for mobile ads. Two popular metrics are Click through rate (CTR) and Cost per thousand impressions (CPM).

- CTR is total number of ads clicked by user divided by total number of displayed ads. In this case, the revenue model will be based on CTR.
  - CTR = (ads clicked by user)/ (total ads displayed)

- CPM is the effective cost per 1000 impressions.
  - CPM = (Advertising cost/Number of impressions) X 1000

Once we define the revenue model and the metrics we can accordingly use the ads.

15.5.2 Mobile ad Selection Based on Business Strategy

In this step, we will select the ad that is appropriate for business strategy. Key ad selection criteria are as follows:

- Based on the business needs, we can select the ad type: banner ad, full screen ad or custom ad.
  - Banner ads are normally placed at the top or bottom of the screen
  - Full screen ads (or Interstitial ads) have high CTR rate but carry high risk of user dissatisfaction. This is usually placed at the end of user flow (such as end of game) or when user navigates from one flow to another (such as user going from one game level to another).
  - Ads can be laid out in custom format with custom size and formats based on the business needs. For instance, ads can be placed in list format or in custom panel format.
  - Ads can be pushed to user through notification feature.
  - Form ads can be designed to collect the user information such as email address and contact details.
  - Many product companies are promoting their products through videos that showcase their product.
- Place the ad strategically so that it does not interfere with end user experience and at a location where it is minimally invasive and least distracting for the user experience.
- The content or rich media content within the ad should be relevant and responsive for high CTR.
- Experiment with various ad formats through A/B testing and multi-variant testing to fine tune the ad size and placement.
- Understand the ads that are effective and continuously fine tune the ad size and placement strategy.
- Ensure that ads are compliant with the marketplace policies, content policies and other regulations.

15.5.3 Monitor Ads

The usage and effectiveness of ads should be monitored through analytics dashboard to understand the CTR and CPM metrics. We can also gather other metrics such as clicks, views, and conversion etc.

☞ **Check Your Progress 2**

1) __________ can be used to show the Windows store promotions.

2) Ads published from other developers can be shown through ______ ads in Windows apps.

3) Cost per impression is ___________________.

4) __________________ ads are pushed to users.

15.6 SUMMARY

In this unit, we started with discussion on various monetization techniques for popular mobile platforms. We looked at key monetization models such as paid model, free model, freemium model, affiliate model, white labeling model and Amazon app store publishing modelfor Android apps. We discussed services model, sponsorship model and Facebook ads model for iOS apps. We studied details of affiliate ads, community ads and donation model for Windows apps model. Finally, we had a deep dive into Mobile ad strategy by looking at key success metrics, ad selection process and monitoring process.

15.7 SOLUTIONS/ANSWERS

Check Your Progress 1

1) users clicked, total number of displayed ads

2) commission model

3) White labeling

4) Services

5) banner ads

6) Interstitial

7) Audience network

Check Your Progress 2

1) Microsoft affiliate ads

2) Community

3) effective cost per 1000 impressions

4) Notification

15.8 FURTHER READINGS

References

1) https://developer.amazon.com/

2) https://developer.amazon.com/help/faq.html

3) http://www.amazonappstoredev.com/

4) https://blogs.windows.com/buildingapps/2017/01/30/monetize-game-app-playtem-ads/#lWTsWQwg9E5xFTht.97

5) https://msdn.microsoft.com/en-us/library/windows/apps/hh202939(v=vs.105).aspx

6) https://docs.microsoft.com/en-us/windows/uwp/publish/monetize-with-ads

7) https://docs.microsoft.com/en-us/windows/uwp/monetize/

8) https://developer.microsoft.com/en-us/store/monetize/ads-in-apps

9) https://developer.microsoft.com/en-us/store/monetize

10) https://developer.apple.com/iad/monetize/Implementing-iAd-in-Your-iOS-Apps.pdf

11) https://developer.apple.com/app-store/business-models/