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

Introduction to Mobile Architecture

School of Computer and Information Sciences
Indira Gandhi National Open University

Commonwealth *of* Learning

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.

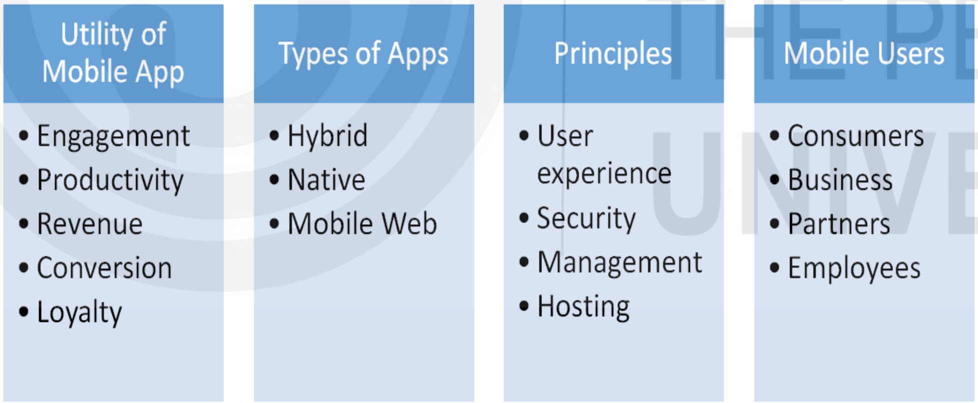
1.1 OBJECTIVES

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
  - Consumer engagement with richer user experience
  - Productivity through efficient flows
  - Driving incremental revenue through user stickiness
  - Customer conversion
  - 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:

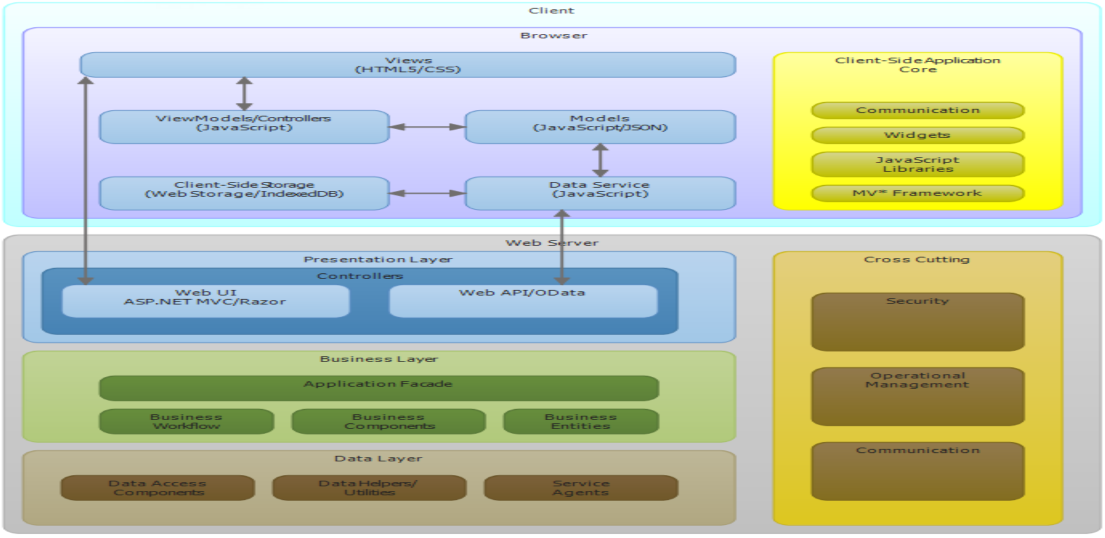


Fig. 1.2: MVC Web Application

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.

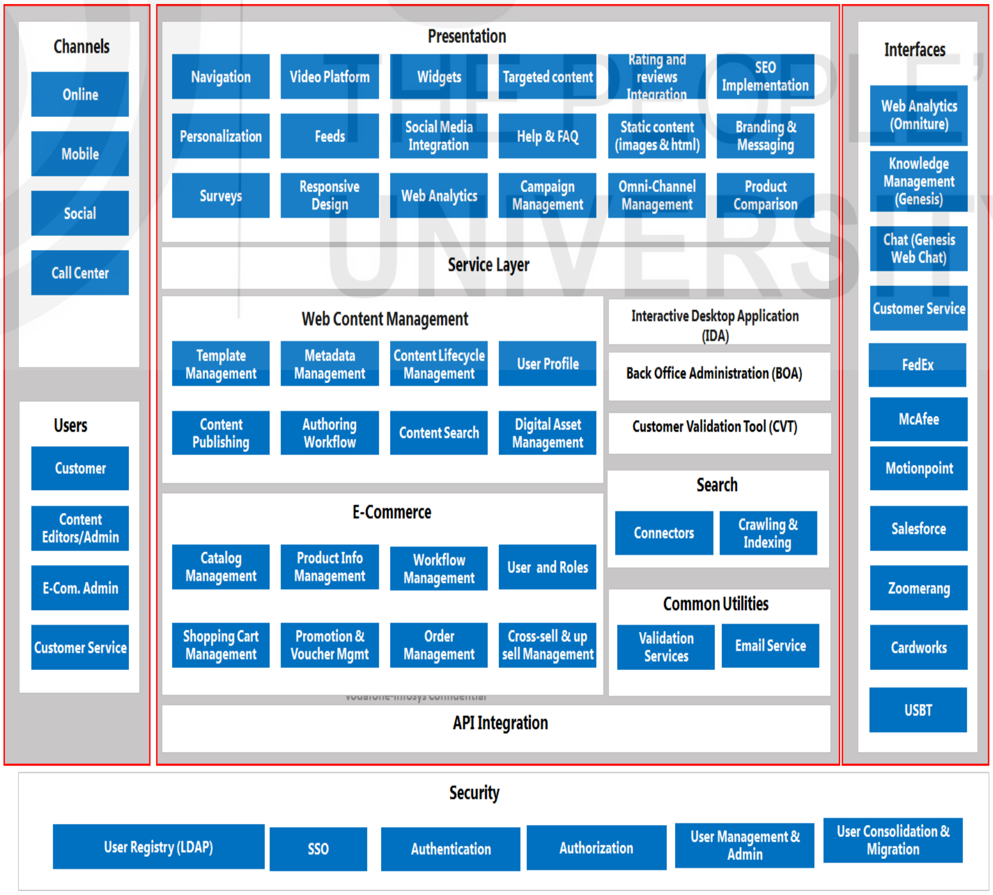


Fig. 1.3: N-tier Enterprise Web Application

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.

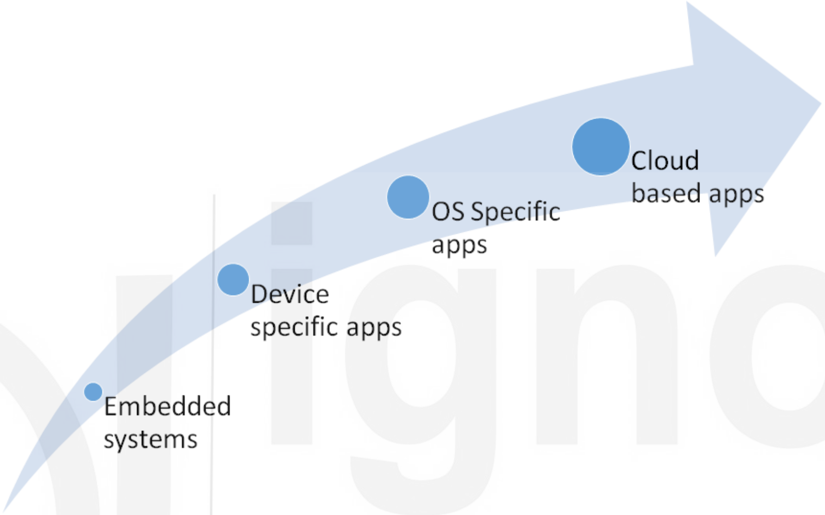


Fig. 1.4: Evolution of mobile app

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:

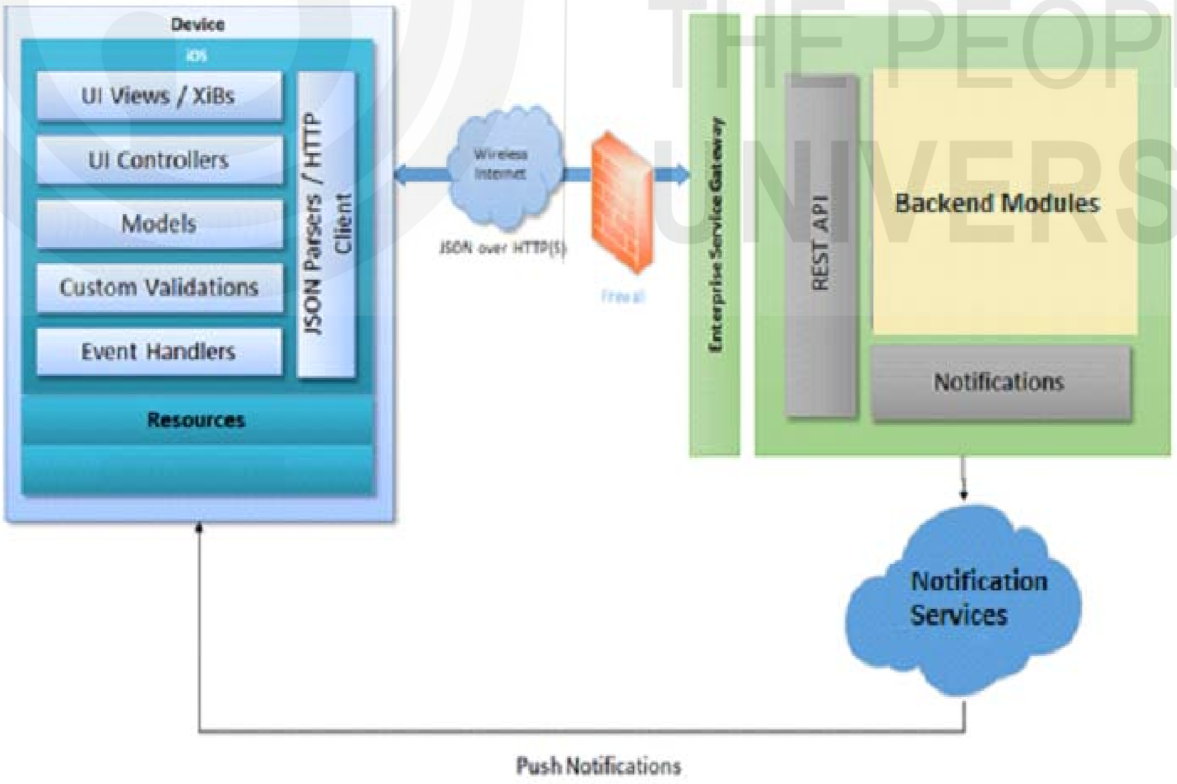


Fig. 2.1: iOS based native Mobile App

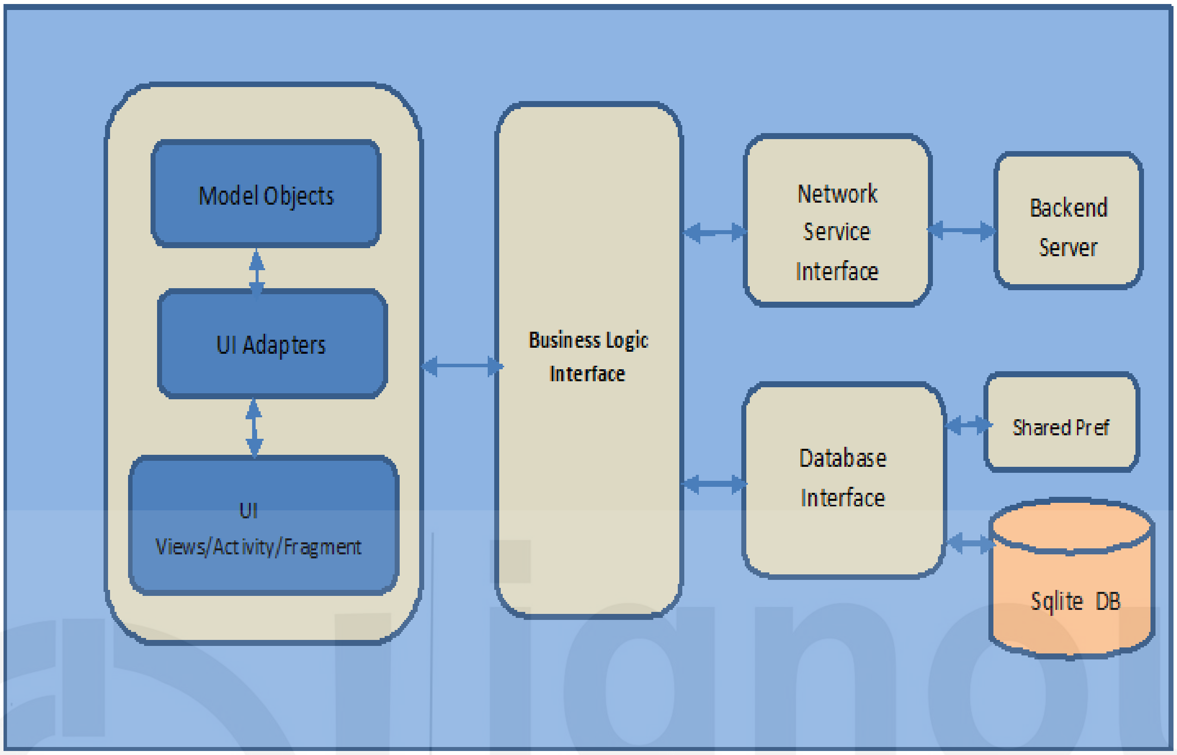


Fig. 2.2: Architecture of Android based native Mobile App

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.

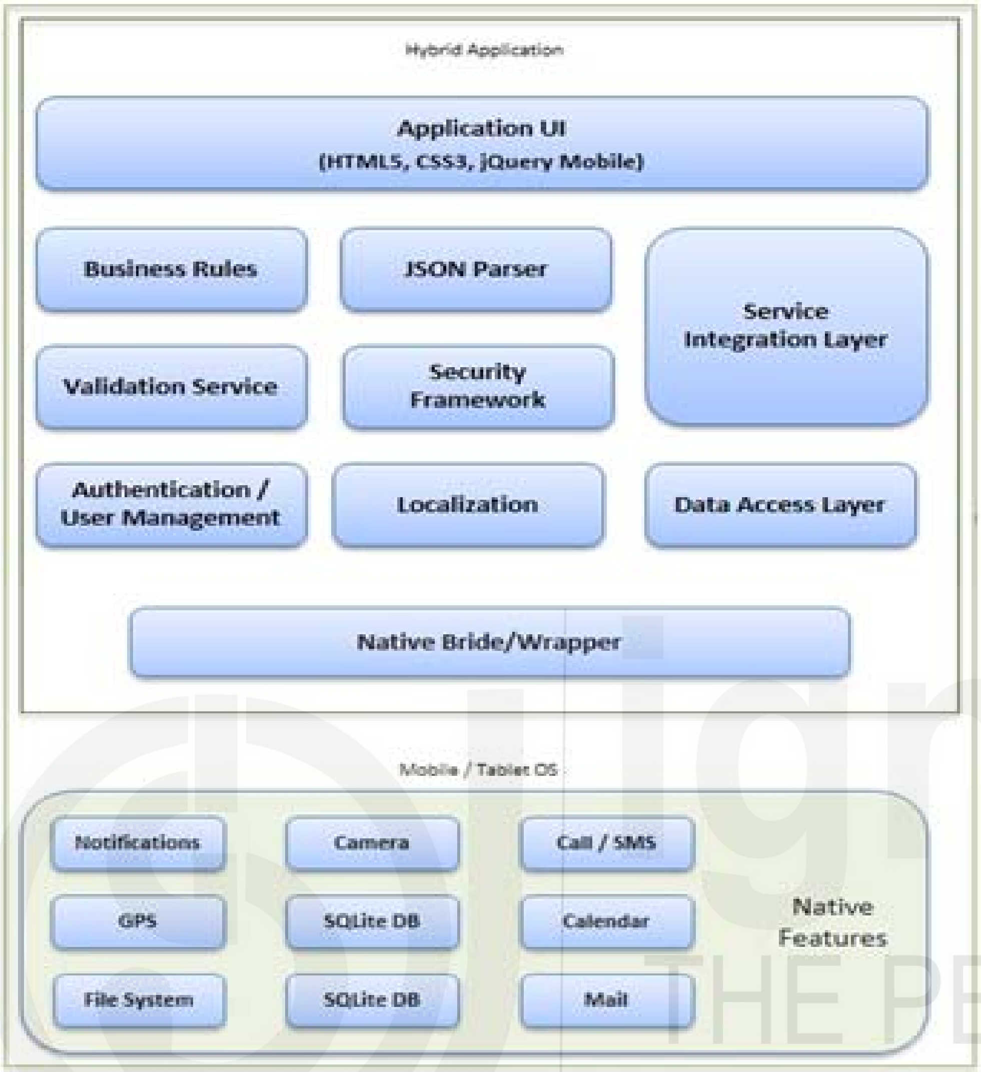


Fig. 2.3: Architecture of a hybrid Mobile App

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.

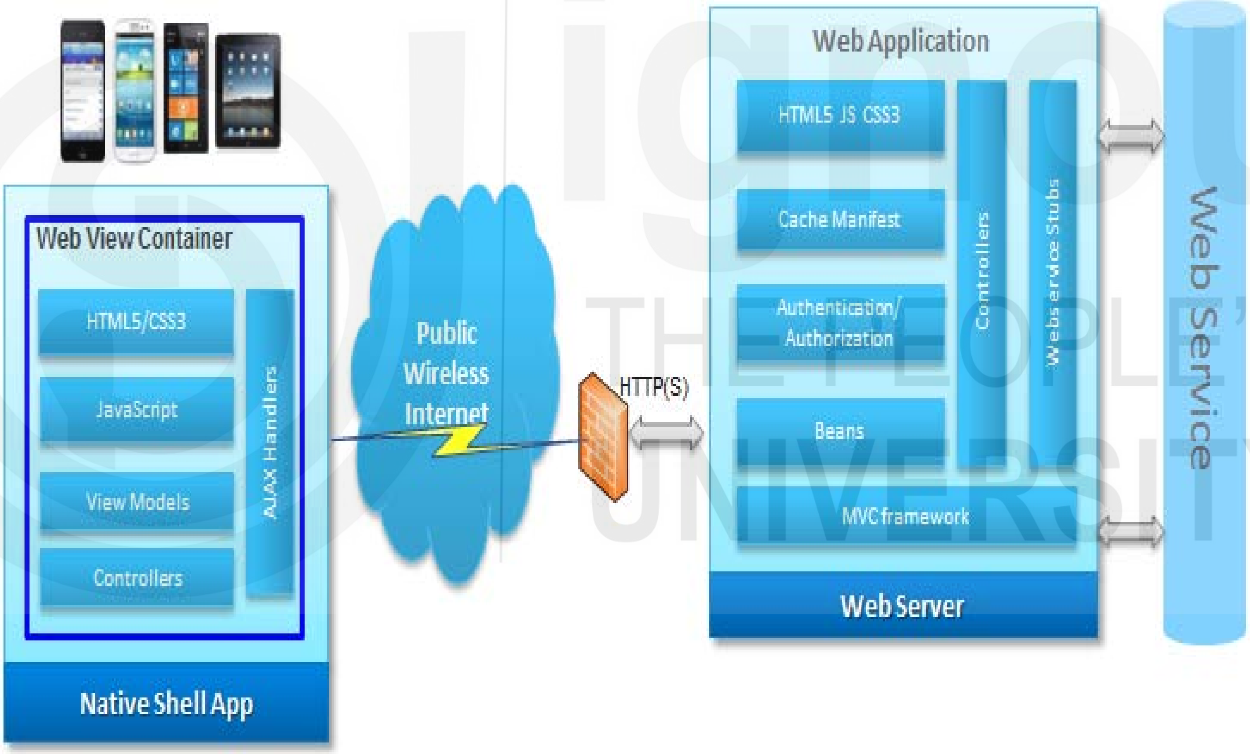


Fig. 2.4: End to End Flow of a hybrid Mobile App

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.

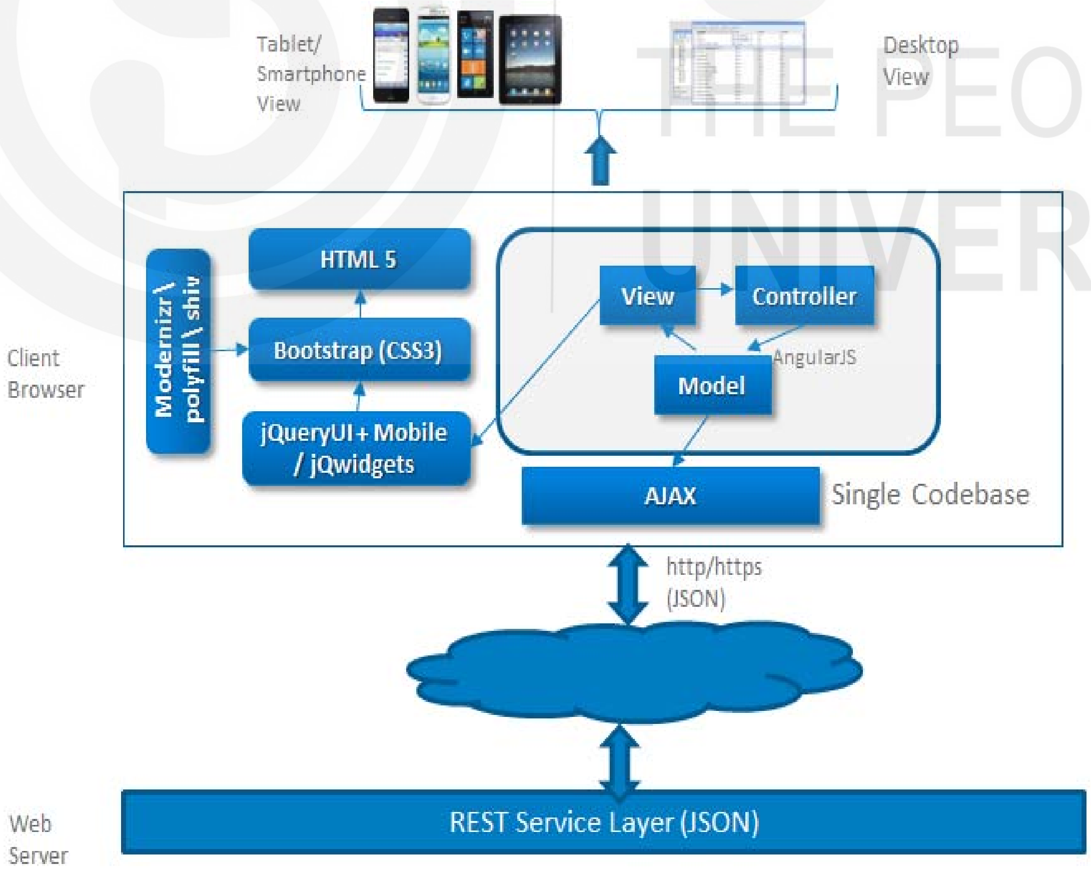


Fig. 2.5: Mobile web app using RWD

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:

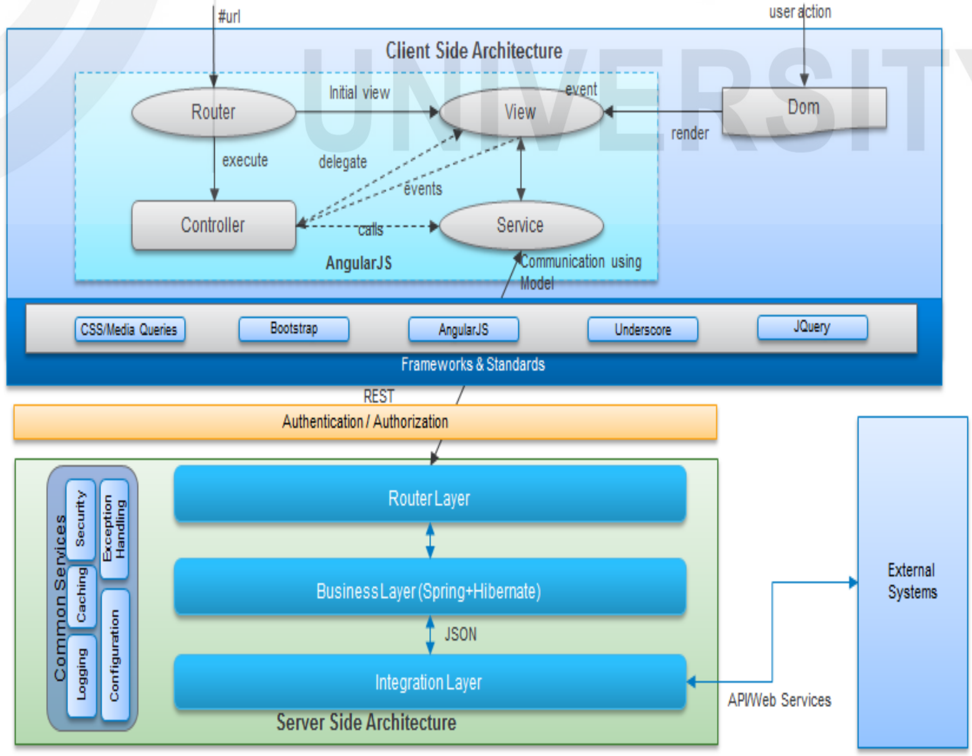


Fig. 2.6: MVC architecture of mobile web app

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.

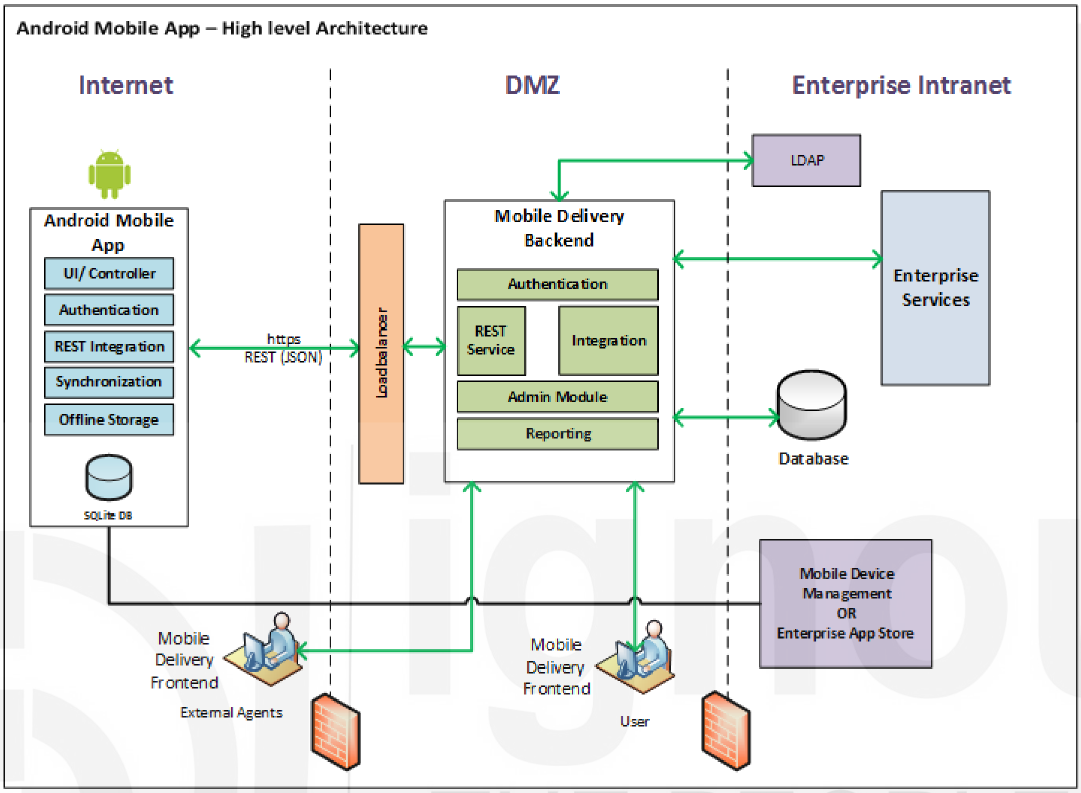


Fig. 3.7: Android App High level architecture

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.

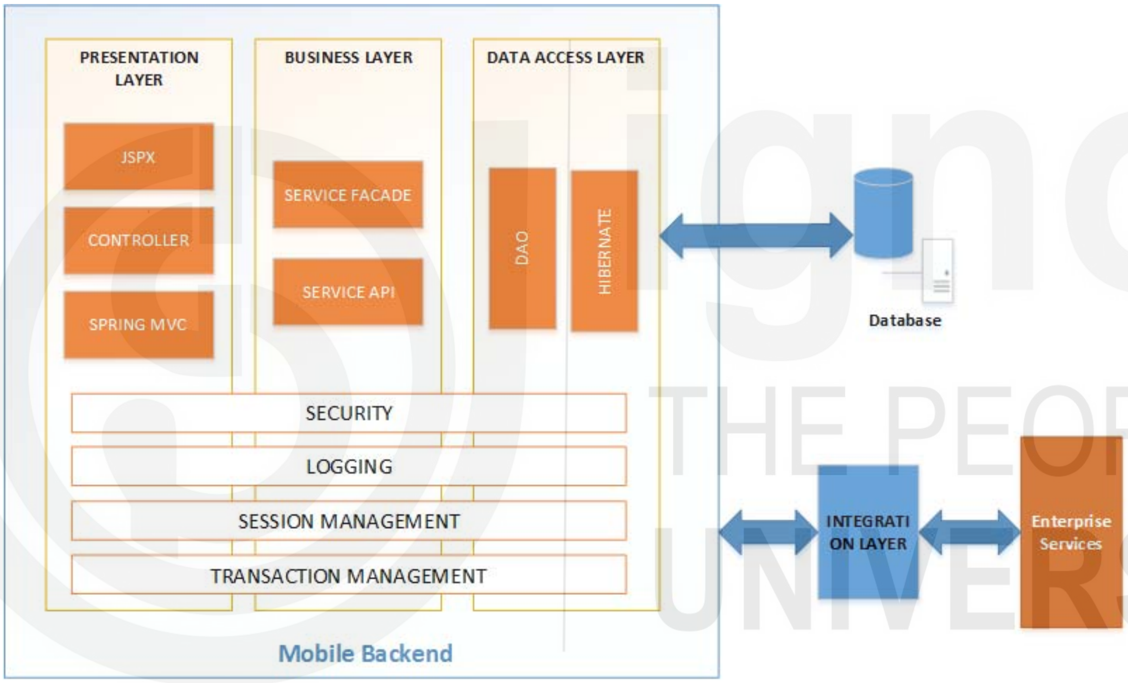


Fig. 2.8: Mobile backend components

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.

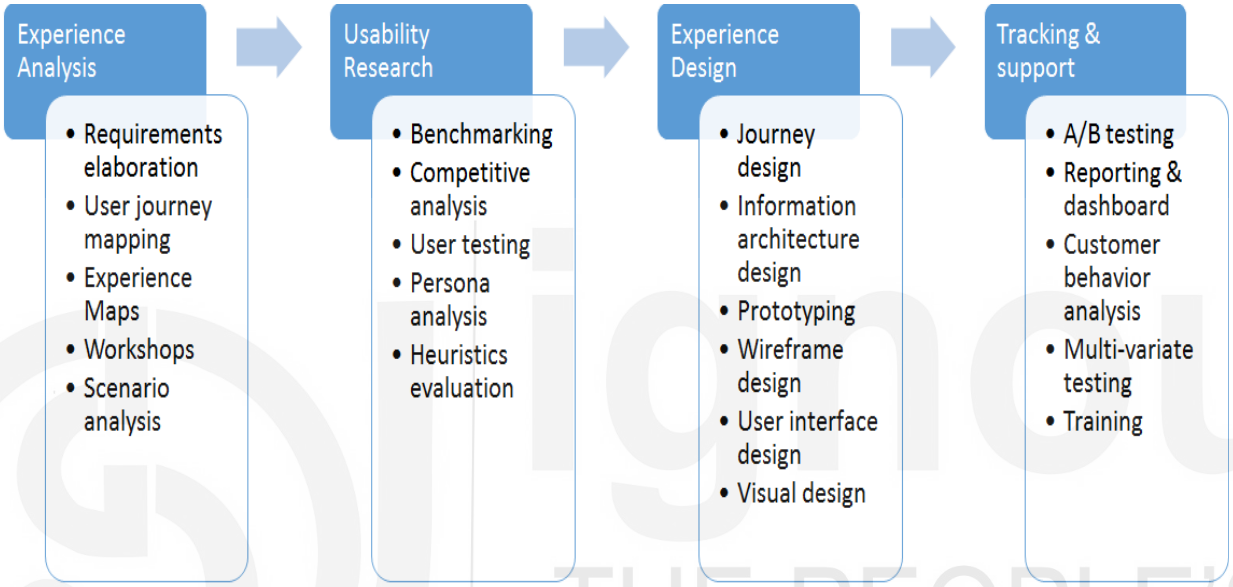


Fig. 3.1: User Interface Design Process for Mobile Apps

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.

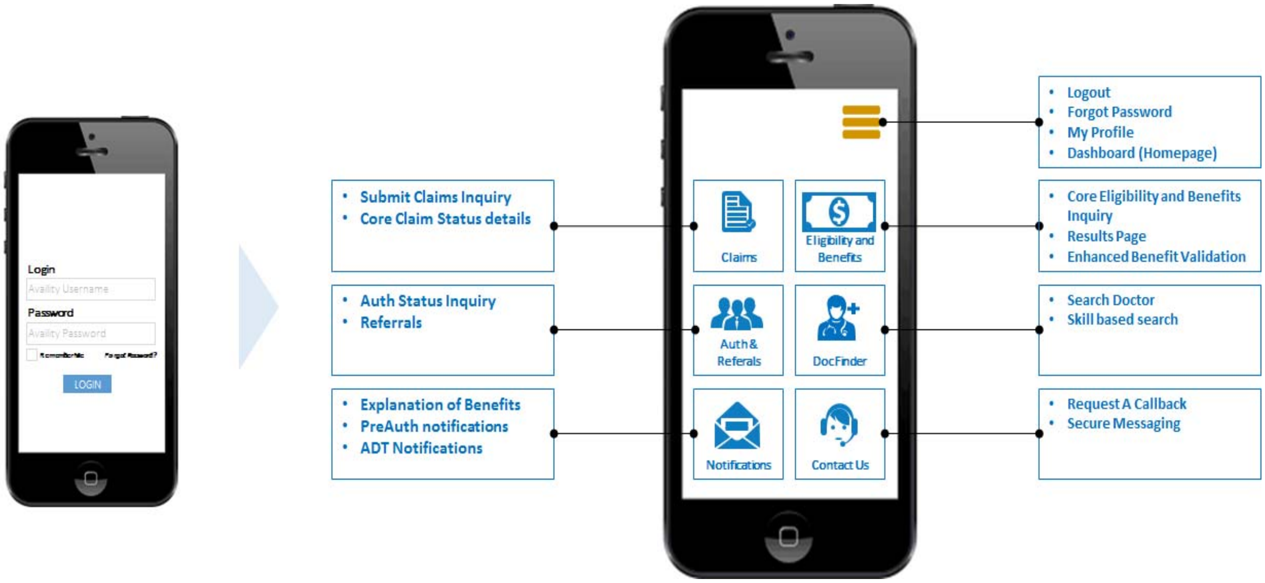


Fig. 3.2: Mobile app user interface of an Insurance Company.

3.5 DEPLOYMENT

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

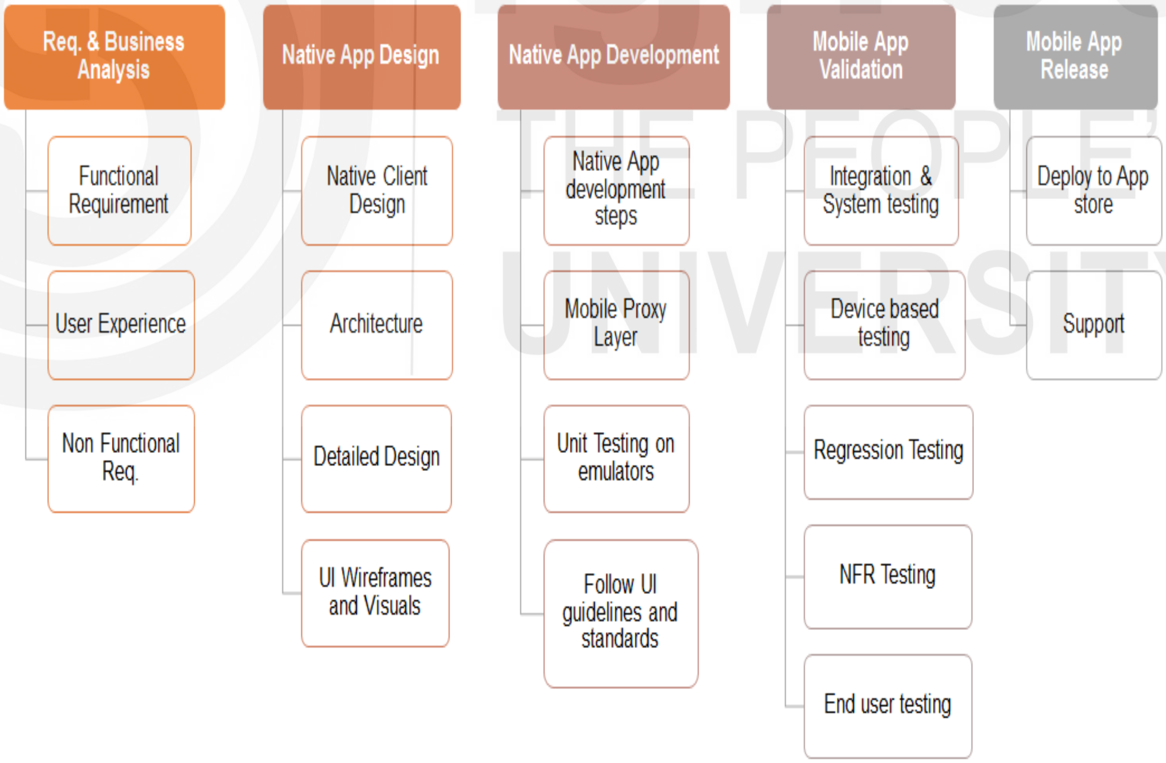


Fig. 3.3: Steps involved in Mobile app deployment

- **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

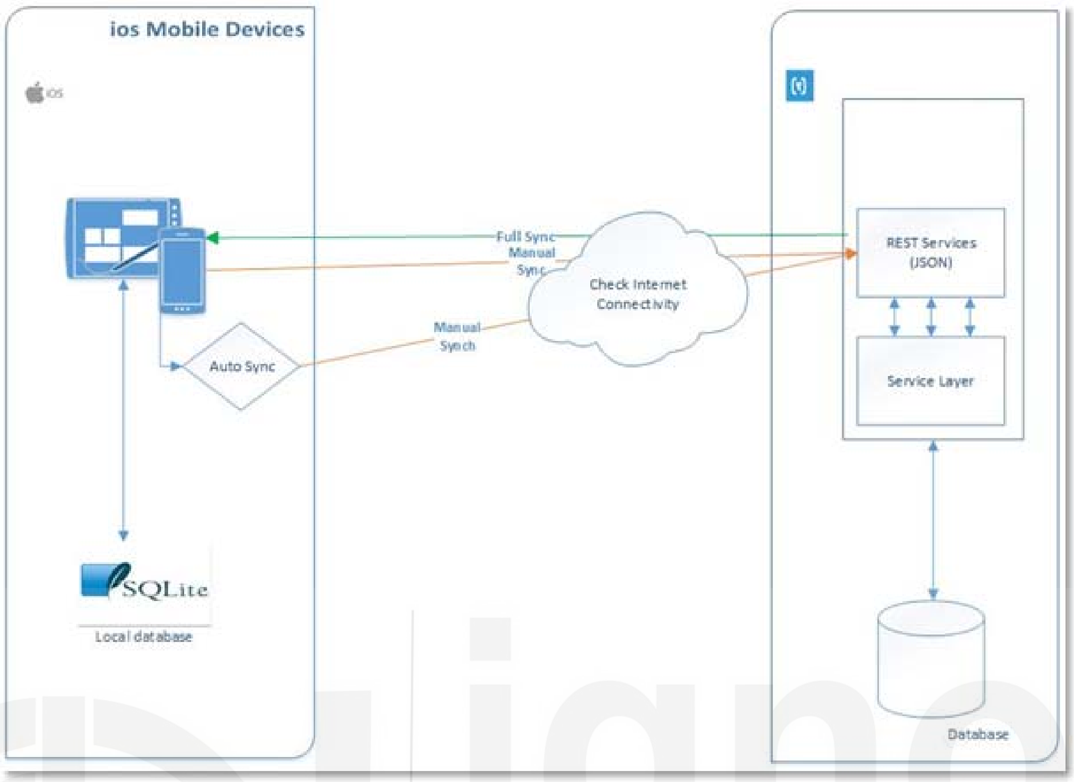


Fig. 3.4: Process of App synchronization

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