



Diploma Thesis

Faculty of economics and management

Mobile application development and testing

Supervisor: - Doc.Ing Vojtech Merunka, PH.D

Author: - Yogin Vaniya

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

©Year of submission 2019-2020 CULS

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Vaniya Yoginkumar Hasamukhlal

Systems Engineering and Informatics
Informatics

Thesis title

Mobile application development and testing

Objectives of thesis

The goal of this thesis is to analyze and develop a mobile phone application on android platform related with the fire safety. The main idea of the application will be safety in the fire situation. Everyone can use this application in the case of fire. They can just login to application and directly go via map and find the available fire vehicle and book it for them in any kind of emergency. Application will automatically take a geo position of a user and show in on the map. User can find available vehicle and book it. Moreover, a booking user also will able to see the navigation of the fire vehicle to the user's location. By this idea user can easily get help in any fire emergencies.

Methodology

The idea about requirements will be in the theoretical part. The second part will be the project. The application will contain several parts like profile, help, history, about us, feedback etc. User can just download the application and register only once. A security issue will be solved. By registering, users can fill their profile and see their profile in profile part at main home page. At home page they will find some several part like help, map, history, feedback so on. They can view the profile at profile page. In map they can click for help and application will take their geo location and it will show the map or location of user and they can see easily the nearby fire station or fire vehicle which will be free and they can easily book fire vehicle for them. They can easily look in history page that how many time they use the application. They can give the feedback via feedback part. Important information will be available in about us part. The application will be implemented in its first working prototype version.

The proposed extent of the thesis

60 – 100 pages

Keywords

Mobile application development; Software process management; UML

Recommended information sources

BURNETTE, E. (2009) Hello Android: Introducing Google's Mobile Development Platform, Pragmatic Bookshelf
BURTON, M. (2005) Android App Development for Dummies, John Wiley & Sons.
Udemy teaching and learning, connecting students, <https://www.udemy.com/>

Expected date of thesis defence

2019/20 WS – FEM (February 2020)

The Diploma Thesis Supervisor

doc. Ing. Vojtěch Merunka, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 25. 11.2019

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 25. 11. 2019

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 12. 03.2020

Declaration

I declare that I have worked on my diploma thesis titled “Mobile application development and testing” by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on **12 March 2020**

Yoginkumar Vaniya

Acknowledgement

I declare that the thesis “Mobile application development and testing” is developed independently under the guidance of my supervisor, Doc.Ing Vojtech Merunka, PH.D for which I am thankful to him and other person for their advice and support given during my research. I also declare that the use of literature and other information sources that are cited in the work are listed in the reference at the end of work. As the author of the referred thesis, I further declare that I didn't infringe the copyright of any third parties in the context of its creation.

Shrnutí

Tato práce obsahuje informace o vývoji mobilních aplikací na platformě Android; studii o vývoji mobilních aplikací a životním cyklu aktivit; popis jak proces o vývoji aplikací funguje a jaké různé složky musí být při vývoji libovolné aplikace pokryty. Tato práce navíc obsahuje informace o základních součástech, které aplikace obsahuje a popis prvků uživatelského rozhraní zobrazení aplikace. Tato práce též obsahuje informace o nejvíce užívaných věcech při vývoji aplikací, jakožto součásti uživatelského rozhraní, sada pro vývoj softwaru Android, menu, zásobník softwaru Android, Manifest, Java, Xml, kontejnery a mnoho dalšího.

Testování mobilní aplikace je jedním z nejdůležitějších kroků při jejím vývoji. Tato práce obsahuje informace o testování aplikací Android. Co se aplikací Android týče, testování je velmi velké téma pro diskuzi. Tato práce se snaží poskytnout informace o tom, jak uživatelé mohou testovat vyvíjené aplikace a jak probíhá běžný proces tohoto testování. Tato práce též poskytuje informace o testování na konkrétním příkladu, který je pracovním prototypem a testování poskytne normální testovací informace s testovacími daty a hlášením chyb.

Klíčová slova: : Vývoj aplikací Androidu, verze Androidu a jeho historie, sada pro vývoj softwaru Android, komponenty aplikace Androidu, zásobník softwaru Android, Životní cyklus aktivit, Manifest Java a Xml složka, Rozvržení, widgety uživatelského rozhraní, menu, kontejnery, datové úložiště, testování aplikace.

Summary

This thesis is conducted the information about the mobile application development on android platform. The study about the development of mobile application and activity life cycle. How development of application process works and what are the different components that needs to be covered for developing any application. Moreover this thesis contains the information about the basis parts that application contains and the explanation about the UI components for application display. Also this thesis covered the information about the most used things in application development like activities, UI components, Android SDK, menus, Android software stack, Manifest, Java, Xml, containers and many more.

The mobile application testing is one of the most important phase of mobile application development. This thesis covered the information about the testing of the android application. Testing is a very big topic for the discussion in android application. This thesis try to provide some information about how user can test the developed application and what is the common process do test any application. This thesis also provides information about the testing of the practical example application which is working prototype and testing provides the normal testing process information with test data and bug report.

Keywords: Android application development, Android versions and history, Android SDK, Android application components, Android software stack, Activity life cycles, Manifest Java and Xml file, Layouts, UI widgets, Menus, Containers, Data storage, Testing of an application.

Table of contents:

Table of Contents

| | |
|--|----|
| 1. Introduction..... | 1 |
| 1.1 What is Android ? | 1 |
| 1.2 Android version names | 1 |
| 1.3 Android version symbols. | 2 |
| 1.4 Need for Android | 3 |
| 2. Goal of Thesis | 4 |
| 3. Methodology of Thesis | 5 |
| 4. Understanding of Key Android Components | 6 |
| 4.1 Android software stack | 6 |
| 4.2 Android SDK | 8 |
| 4.3 Android application components | 10 |
| 5. Overview of types of Mobile Applications..... | 12 |
| 5.1 Native applications..... | 12 |
| 5.2 Web applications..... | 13 |
| 5.3 Hybrid applications | 14 |
| 5.4 Comparison of different applications..... | 14 |
| 6. Basics of Mobile Application Development on Android platform..... | 15 |
| 6.1 Activity..... | 15 |
| 6.1.1 What is Activity?..... | 15 |
| 6.1.2 Types of Activity in Android | 15 |
| 6.1.3 Explanation of Android Life Cycle | 19 |
| 6.2 Introducing Android Manifest file | 22 |
| 6.3 Importance of Android Java file and Xml file | 22 |
| 6.4 Layouts for the Android Applications..... | 23 |
| 6.5 Multiple UI Widgets for Android Applications | 25 |
| 6.6 Menus in Android | 30 |
| 6.7 Containers for Android Applications | 31 |
| 7. Messaging object and Display of Information in Mobile Applications..... | 33 |
| 7.1 Intents in Android Applications | 33 |
| 7.2 Notifications for Android Applications | 34 |

| | |
|---|----|
| 8. Different types of Data Storage for Android Mobile Applications | 38 |
| 8.1 Shared Preferences in Android..... | 38 |
| 8.2 Different Storage for Applications..... | 39 |
| 8.2.1 Internal Storage | 39 |
| 8.2.2 External Storage | 39 |
| 8.2.3 Cache Storage | 39 |
| 8.3 SQLite Database for Android Application..... | 40 |
| 8.4 SQL Databases for Back-end Development | 40 |
| 9. Practical Example of Mobile Application Development on Android platform..... | 42 |
| 9.1 Idea of Example Application | 42 |
| 9.2 Problem Definition..... | 42 |
| 9.3 History of technology and work-done..... | 43 |
| 9.4 System requirements and constraints | 43 |
| 9.5 System analysis | 45 |
| 9.6 Activity/Process in the application..... | 46 |
| 9.7 UML diagram and Screenshots of the applications UI | 47 |
| 10. Testing of An Android Mobile Application..... | 63 |
| 11. Conclusion | 74 |
| 12. Bibliography | 75 |
| 13. Appendix..... | 77 |

Table of Figures:

| No of Figure | Description | Page no. |
|---------------------|---------------------------------------|-----------------|
| 1.2.1 | History of Android versions | 2 |
| 1.3.1 | Android versions symbols | 2 |
| 4.1.1 | Android software stack | 6 |
| 6.1.1 | Different Android activities | 16 |
| 6.1.3 | Android activity life cycles | 20 |
| 7.2.1 | Inbox style notifications | 35 |
| 7.2.2 | Big text style notifications | 36 |
| 7.2.3 | Big picture style notifications | 37 |
| 9.7.1 | Use case diagram of application | 47 |
| 9.7.2 | Class diagram of application | 48 |
| 9.7.3 | Starting screen of application | 49 |
| 9.7.4 | Login screen of application | 50 |
| 9.7.5 | Register screen of application | 51 |
| 9.7.6 | Verify OTP screen of application | 52 |
| 9.7.7 | Make profile screen of application | 53 |
| 9.7.8 | Home screen of application | 54 |
| 9.7.9 | View profile screen of application | 55 |
| 9.7.10 | Edit profile screen of application | 56 |
| 9.7.11 | Fire screen of application | 57 |
| 9.7.12 | Help screen of application | 58 |
| 9.7.13 | About us screen of application | 59 |
| 9.7.14 | History screen of application | 60 |
| 9.7.15 | Feedback screen of application | 61 |
| 9.7.16 | Forgot password screen of application | 62 |

Tables:

| No of Table | Description | Page no. |
|--------------------|---|-----------------|
| 1 | Comparison of different mobile applications | 14 |
| 2 | Activity/Process in the application | 46 |

1. Introduction:

1.1 What is Android?

Android is a mobile operating system which is growing very rapidly in the world today. Android is the most common combination of Linux kernel and some other open source software. Android is the most used software stack for every mobile devices. This software stack mainly contains various parts which are operating system, middle ware and key applications. This operating is mainly designed to give touchscreen experience to the world. Most common devices uses this operating systems as touch screen experiences are mobile phones, laptops, tablets etc. Android is type of operating system which contains the large number of applications.

It was founded long years ago in October 2003 and it was developed by Andy Rubin, Rich Miner, Nick Sears and Chris White, in Palo Alto, California and later on it was purchased by Google in August, 2005. Since then it is subsidiary of Google Incorporation. The first commercially available smartphone running Android operating system was HTC Dream and it came into market as the T-Mobile G1 in the USA, on October 22, 2008. (History of Android, 2013)

1.2 Android Version Names:

Android operating system has a different versions. After some period of time Android change their versions and give a new latest version to the market for better user experience. Every versions of Android is names as a sweets name. The most first version of Android was known as Aestro. Later on after some time intervals Android introduced lots of different and better versions of the Android. The latest version of Android which is currently used in the market is known as Android Pie 9.0 and it is released on August 6, 2018. (Android overview, 2004)

| Version ↕ | Code name ↕ | Release date ↕ | API level ↕ |
|-----------|--------------------|-------------------|-------------|
| 10 | 10 | September 3, 2019 | 29 |
| 9 | Pie | August 6, 2018 | 28 |
| 8.1 | Oreo | December 5, 2017 | 27 |
| 8.0 | | August 21, 2017 | 26 |
| 7.1 | Nougat | October 4, 2016 | 25 |
| 7.0 | | August 22, 2016 | 24 |
| 6.0 | Marshmallow | October 5, 2015 | 23 |
| 5.1 | Lollipop | March 9, 2015 | 22 |
| 5.0 | | November 3, 2014 | 21 |
| 4.4 | KitKat | October 31, 2013 | 19 |
| 4.3 | Jelly Bean | July 24, 2013 | 18 |
| 4.2 | | November 13, 2012 | 17 |
| 4.1 | | July 9, 2012 | 16 |
| 4.0 | Ice Cream Sandwich | October 19, 2011 | 15 |
| 2.3 | Gingerbread | February 9, 2011 | 10 |

Figure-1.2.1

1.3 Android Versions Symbols:



Figure-1.3.1

1.4 Need For Android:

There are so many reasons that force us to use Android operating system. There are several main points that we can consider as a focus to use Android. Also we have some other reasons to use Android operating system. The mainly android is open source operating system which means programs source codes are useful for users and developers for both. Android has very large developer and community reach. Android can easily increase the marketing strategies of any company. Android provides very large inter app integration so user can easily use lot of application at the same time.

Android provides very less expensive development cost. Android operating system has very high amount of success ratio compare to other operating systems. Android has very rich application development environment which is user friendly and easy to use for all the users.

2. Goal of thesis:

The main goal of this thesis is to understand the process of android mobile applications development. This thesis is mainly focus on how the mobile applications are develop on the android platform. This thesis provides information about the what is an android in real-time, what is the past and upcoming future of the android, android life cycles, components, data resources and the designing and the developing an android application for the users with the proper testing of developed application. This thesis also include almost every necessary part that is most important in application development in android platform.

This thesis has one practical example of fire safety which explains an android mobile application development which is working prototype and provides the basic fundamentals of the android application development. This fire safety application is useful in fire emergency. User can log in to the application and go directly via map and find the available vehicle and easily take the help in fire cases. Application is able to take the geo place of the user and provides the nearby fire vehicle in the map which users can easily book for them for help. Application is also providing the live navigation of fire vehicle to the user place so that user can easily estimate the time period for help. This thesis is also providing some common testing procedure for the practical example application to understand the testing part of android application.

3. Methodology:

This android mobile application development and testing thesis include the method of all necessary android application development standard. This includes the each steps of application development from designing the application to the testing of the application. The practical example application is fire safety which has different parts which provides different functionality.

User can use this application by registering only once by mobile number which is a perfect way of user authentication. This application is using one time password (OTP) for the user authentication for security purpose. After registering, user is filling their profile which user can see in the profile part in the home page of the application. The home page of application has multiple parts like Profile, Fire, Help, About us, History, Feedback. Every part is providing the different services to the user. Profile part is providing the information about user profile details which user filled during registration process and now user can view and edit that details in this part. Fire is the main part which is the main purpose of the application which providing the map where users can see nearby fire vehicles and book it for them. Help is providing some contacts to the users who need some help while using the application. About us is nothing but some overview of the application. History is a part which gives information about how many times user booked a vehicle in the application and some details about the booking. Feedback in this application is providing the gateway where user can drop some feedback or message after using the application.

4. Understanding of Key Android Components:

4.1 Android Software Stack:

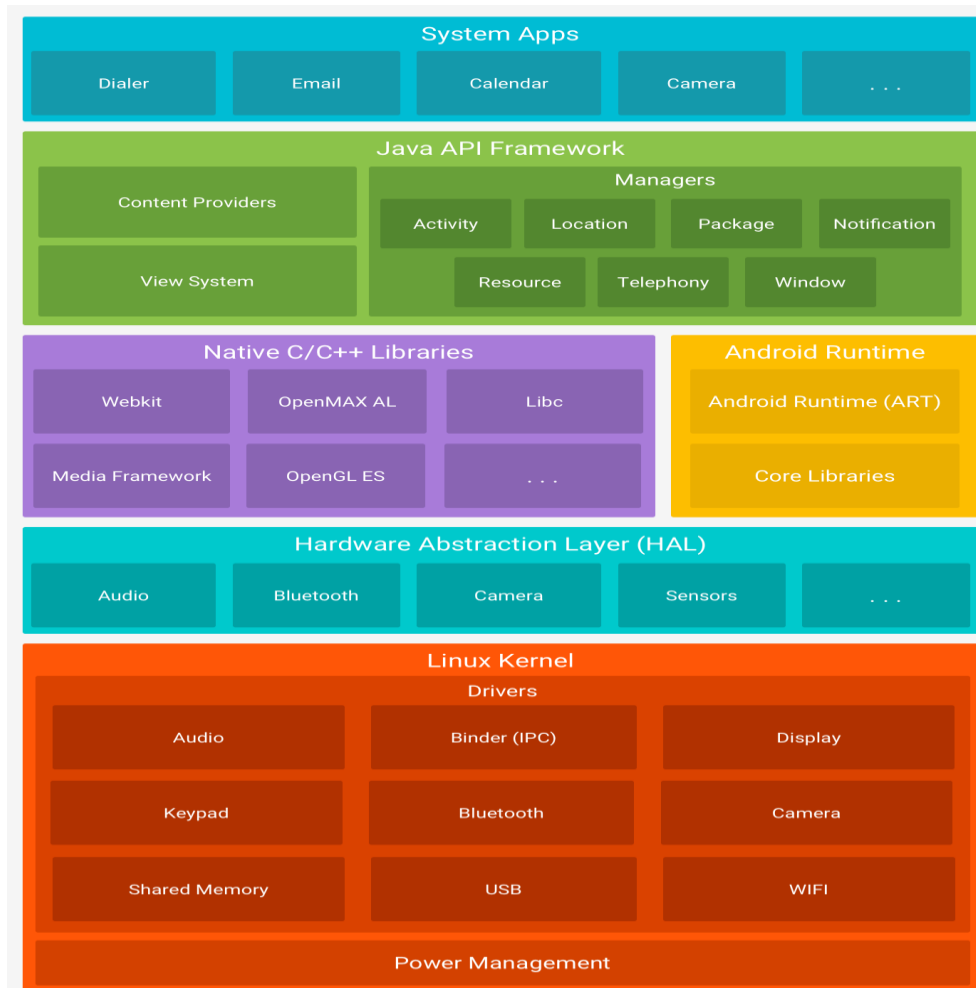


Figure-4.1.1

Android is considered as a software stack which contains a lot of things or parts which describes the total architecture of android operating system. Mainly it contains so many different parts but usually it categorised as main three parts which is operating system, middleware and key applications. (Burnette, 2010)

A. Linux kernel:

Basically in android development the main operating system is nothing but the Linux kernel. Linux kernel is more power full platform and itself has a large number of functions or services which makes android better than others. Most common feature or function that Linux kernel has is memory management which allocates the memory to new files and itself make memory free when a particular file is deleted or removed. Linux kernel has power management which used to give power to different devices like Camera, Bluetooth etc.

Linux kernel has function called driver management which automatically install the various types of drivers. Linux kernel has function called resource management which used to give the resources to every processes running. Because of this function android has ability to do multitasking at the same times which means android supports lots of applications at the same time. User can easily use more than one application the time. For example look pictures in gallery, listening to music, chatting on WhatsApp. (Burnette, 2010)

B. Hardware Abstraction Layer:

Hardware abstraction layer is also known as HAL which is a one kind of interface HAL is an interface which is used to establish the communication between the Linux kernel and the software APIs. HAL keeps the tracks of different actions carried in the systems by different vendors and always provides the API for every actions. Hal consists of vendor specific C and C++ implementation of multiple library modules in which each has access to different type of hardware. So when framework make call to use the hardware then only the android load library module for that hardware component.

C. Native C/C++ Library:

Native C/C++ library is also an important part of android software stack. Native C/C++ library is used by android to make interactions with Linux kernel via HAL. Android programming interface (API) is also a part of or related with the set of Native C/C++ libraries. There is number of API which uses the different set of Native C/C++ libraries. For example android.opengl library is using the OPENGL ES a C++ library. Native C/C++library is also used or accessed by the android APIs. (Burnette, 2010)

D. Android Runtime:

Android runtime is also popular as ART. Android runtime is nothing but a managed runtime which is used by number of applications in android or some system services on the android platform. Android runtime was specially created for android projects. Android runtime is used as a runtime to execute the Dalvik executable format and Dex bytecode specification. Mainly this part of the software stack has the two different things that is Dalvik virtual machines and code libraries. Nowadays Dalvik virtual machine is replaced by the latest ART. (Burnette, 2010)

E. Java API Framework:

Java API Framework is most important part of the android software stack. Java API Framework provides the set of APIs which helps developers to easily create different applications for android platform. Java API Framework used to reduce the amount of codes that developer needs to write to perform some operations and it saves lots of time for developer. Java API Framework is API set that helps developer to create simple class for particular actions and increase their speed of developing applications. For examples To change or know the state of wifi rather

writing the large amount of codes Java API Framework helps to create the class called wifiManager to check the all the tasks or actions related to the wifi. (Burnette, 2010)

F. System APPs:

System application is the top most layer of the android software stack. Android has some of core default applications for messages, emails, internet, contacts, calendar etc. Applications which is installed by default they don't have special status among the applications that user wants to install in their devices. Most of the time in system apps the third-party applications can become the default applications for the users. All the applications which use android frameworks use the android runtime and the libraries. And android runtime and native libraries always use the Linux kernel via HAL.

4.2 Android SDK:

In android, SDK means software development kit. Android SDK is a set of development tools which is used to develop an application on android platform. Android SDK and java programming language both are very useful to develop any application on android platform. Android SDK has everything that a single application want means development, testing and debugging an android application. Android SDK also used to write android programs on command prompt and this method of writing command is known as Integrated development environment (IDE). Android SDK mainly includes the following components which are most important for application development on android platform. (Android software development, 2006)

1) Android APIs:

API is nothing but the application programming interface. API is usually the collection of some protocols and part of rules for building the different types of the software application. API can be also considered as a collection of a software modules which comes together to make complete android SDK. API provides the information about how the different part or components of the software will be interact with each other on the application. During the programming of Graphical user interface this APIs are most important aspects. (Android SDK, 2005)

2) Development tools:

Android has different types of development tools to develop android applications. Android development tools allows developer to develop innovative and power full as well as interactive mobile applications on android platform. Android development tools generally known as two different type. Sometimes it is SDK tools and sometimes it is platform tool. Android SDK also has some tools which helps developer for compilation of application, debugging of application or running application. (Android SDK, 2005)

3) Android Emulator:

Android emulator is a virtual mobile phone inside the development tool. Android emulator is full functioned and full interactive android device emulator. Android emulator is simply an android virtual device (AVD) that look same as any android device or mobile phones Android emulator is mostly used to run and test application that developed on the software. Android emulator gives the same display of application like as it would be running on original mobile device of developer.

Android emulator has very different options and interfaces to run and test application on kinds of device emulators or with some default custom hardware features like Resolution, RAM, CPU etc. (Android SDK, 2005)

4) Documentation:

Documentation is simply a place which provides detailed information about everything related to the android application development. Main focus of documentation is provides the code level references information. Code level references contains the details about every important things like what kind of things are included inside the different types of packages and classes and developer can how to use this information. Documentation also has the information about how simply a developer can start developing an application and provides some fundamental requirements or information about the developing an application on android platform. (Android SDK, 2005)

5) Sample codes:

Sample codes is nothing but a simple example codes provided for developer to learn and understand how to code for simply make a normal application. Android SDK also has a sample codes for developers to create various kind of applications. SDK provides services to choose some different kind of the example application to help developer to understand the requirement of the project and helps to understand the android APIs. Sample codes in SDK also give some coding experience to the developer or the users who are newly started android application development. SDK manager also provides different types of the Android platform and by this platform users can easy get different types of sample applications. (Android SDK, 2005)

4.3 Android Application Component:

Android application has main four different types of components. Each of this component is most important for building a mobile application on android platform. Every components has their own and different services that helps developers to complete their application in working phase. These are most important building blocks that each android application contains. These components are part of android manifest files in which these components are briefly described as well as explained. There components are Activities, Services, Content providers and Broadcast receivers.

- **Activities:**

Activities are the main accentual part of the any mobile application. Activities are nothing but it's like presentation layer of an application which can be designed as our requirements. Activity is nothing but a single screen with which any user can interact. Developer can make the user interface of any application with many extensions with respect to activity class. Each activity is also a subclass of any class activity which defines in android SDK. It is possible that an application has more than one activity and that each activity can easily operates independently.

Every activities can be linked to another activities in the applications but it is necessary that the activity that create in the application it must need to be defined in the android manifest file. . (Burnette, 2010)

- **Services:**

Services are the process which is invisible to user in any application. A long running background processes which never requires any kind of user interaction is called as a service. This is a processes which always running in the backend and always keeps the data of application updated, makes the notifications available for the user and keep trying broadcasting the data and services. Services also sometimes performing the tasks which never required the application to be available or activate on particular time periods.

For example a service might play some songs in the background when user using the different application and it also fetch some data over the internet without disturbing the other activities. Each services is a subclass of service class in the android SDK. (Burnette, 2010)

- **Broadcast Receivers:**

Broadcast receiver is nothing but a one type of active intent or intent listener which is working in the application. Broadcast receiver is a kind of handler which usually handles the communications between the android operating systems and applications.

Broadcast receiver is a listener which try to make an application to listen a intent that try to match the requirements entered by the user of an applications.

Broadcast receiver never display any kind of user interface but it's make a notification to let know the user about any kind of broadcast events that occurs in the application.

For example user can see the sign when plug-in the headphone and gets the notifications when the battery is low this kind of things are examples broadcast receiver. (Burnette, 2010)

- **Content Providers:**

Content providers are the interface which shared the data with different applications. Content providers also manages the data of any applications and always try to interact with the SQL databases. Using content providers android can easily share the data directly between the different applications.

Content providers of one applications can be configured to use by the different applications. Content provider is the sub class of the class Contentprovider in android SDK. (Burnette, 2010)

5. Overview of types of Mobile Applications:

Mobile applications were designed to make the user experience very easy and more interactive. Specially, this applications are created to run inside the mobile devices like smartphones, smart pads and other devices. Nowadays mobile devices are became more useful and more popular among the world that is a reason behind the intensive growth of the mobile applications. Every day the number of mobile applications are increasing because everyone develops amount of mobile applications to make every task very easy. According the study the use of smart phones or mobile devices is rapidly increased everyday compared to desktop devices.

Most of the mobile devices and mobile applications are mostly need internet access to work fully functioned and some applications are also working in offline mode. As there are so many types of desktop applications also there is a various types of the mobile applications.

The type of mobile applications are usually categorised according to the different types of the programming languages used to develop the mobile applications. These mobile applications are mainly categorised as Native apps, Web apps and Hybrid apps. Every single type of the application has their different working methods as well as different features and some advantages and disadvantages of using it.

5.1 Native Applications:

Native applications are developed for the particular type of the device according to the user requirements. Native applications are a kind of system program which only created to use for particular platform or devices. Native application is the application which can be developed only using native programming language. Because of the reason that native applications are created for the particular platform or device they can have approach for using device specific hardware and software.

For IOS platform the native application can be developed using native programming language or swift, similarly for Android platform java can be used to develop native applications and C++ is used for the windows mobile phones. For making it available for users on play stores developer must need to develop native application in two different platform for Android and IOS. Usually this applications are running on system level so it can take more use of system services like fingerprint, camera, GPS etc. Applications which are available on the Google play are most of the native applications.

There are a lot of different advantages that native applications provides like it can give different user experience in different platform, it provides the best user experience to the user and better easy to use version for user, also its runs offline most of the time and give user access to use the different resources for the local devices. These is some disadvantages like native applications are very expensive because the development and maintenance are very high, user need to upload it on the Google play and share

the profit with them, and at last it's very difficult to transfer the applications to different platforms.

But anyways native applications are most used applications because it provides better user experience and high-speed performance also it has notification handler so most of the messaging applications are developed as native applications. (Types of Mobile Apps by Technology, 2017)

5.2 Web Applications:

Web application is the second type of the mobile application. Web applications are totally different from the native application. Usually this is the applications which needs some types of the remote server to storing the data of the application. Web applications need common internet application which resides on server and this application only works through the internet. This is a type of application which always need some remote servers and it is deliver on internet to the user through some kinds of web browser interface. Web applications are designed for some purposes and it can be used by anyone in the organisation. Web applications are almost same for every platform and user can easily use it because of its user-friendly nature.

Using web applications user get several benefits like saving the time of development as well as it can helps developer to use fewer resources. The best example of web app is Gmail which works same on every platforms and gives the better user experience in every devices also user can use this on web sites via login in the site it can give the same results everywhere.

Web applications have some advantages like they are very easy to develop and they don't have kind of installation requirements. The maintenance cost and development cost of this kind of applications are very less. Also they can be updated easily via any other requirements because compared to native applications the web applications have very less requirements and specifications.

There are a disadvantages like web applications need internet and server to use this kind of applications which costs little high. Using web applications have several kind of security risks as well as risk of hacking and data loss. Sometimes this applications are working little slowly because of slow internet or less bandwidth and it's not supports in some web browsers. (Types of Mobile Apps by Technology, 2017)

5.3 Hybrid Applications:

Hybrid applications are nothing but the combinations of the native applications and web applications. Also it is considered as a web application which covered or well embedded by the native application. The main motive is to give user the best experience and the advantage of both the native and web applications. When hybrid applications are downloaded and installed in the local devices they are connect and embedded by shell which provides the same user experience and the services via web browser that means user can experience both native as well as web applications.

This is the application which has several features like saving data, running offline, user can get access to the data resources and also users can download this applications in play stores. This applications are busy to develop because half of application are web-based application so it can be easy transferred in different platforms and rest part is native application which only need a special development and building.

The advantage of this application is that it gives the best user experience and it is easily transferable in different platforms. It's a combination of native and web applications so it's for this application to update without any kind of installation and it has same development like web application and all the features are provided same as native application.

The disadvantage of this application is same like web-based because also it has a part of web app that it has very slow response sometimes because of internet and servers also it has very low performance due to fewer features supports and compatible. (Types of Mobile Apps by Technology, 2017)

5.4 Comparison of Native app, Web app, Hybrid app:

| | Native App | Web App | Hybrid App |
|---------------------------|---------------|---------------|--------------|
| Available on | App Stores | Web Sites | App Stores |
| Speed of development | Very Slow | Fast | Medium |
| Expense of development | Very High | Very Low | Medium |
| Way of maintenance | Very Hard | Very Easy | Medium |
| Performance ability | Very Fast | Very Slow | Medium |
| Offline feature | Supported | Not-supported | Supported |
| Installation requirements | Required | Not-required | Not-required |
| Cross platform support | Not-supported | Supported | Supported |

6. Basics of Mobile Application Development on Android Platform:

6.1 Activity:

6.1.1 What is activity?

In android activity is a single screen with which user can able to interact. It is one type of screen provided for android application's user interface purpose. Activity is almost similar like the same desktop application has windows. Activity provides the window where the application possible to draw its UI.

If .java class needs to be an activity it must contain or extends as the activity class. Activity is managed by the activity stack. It is possible that application can contain more than one activity that means the application can contain more than one single screen.

Every activity in the application are interconnected or linked with each other and it is possible to share data and information between them. When new activity is started it placed on the top of the stack and become the running activity of the application. The previous activity remains under the new activity in the stack and it will not come to foreground again until the new activity running in the application. (Burton, 2015)

6.1.2 Types of activity:

There are different types of the activities used in the android application projects. Any platform that user uses for developing any application ask for which kind of activity user wants to use. Every single type of activity in the projects used for different types of purposes. Almost all the different type of activity has their own design patterns as well as different use for developing different applications. These all the activities are briefly explained below.

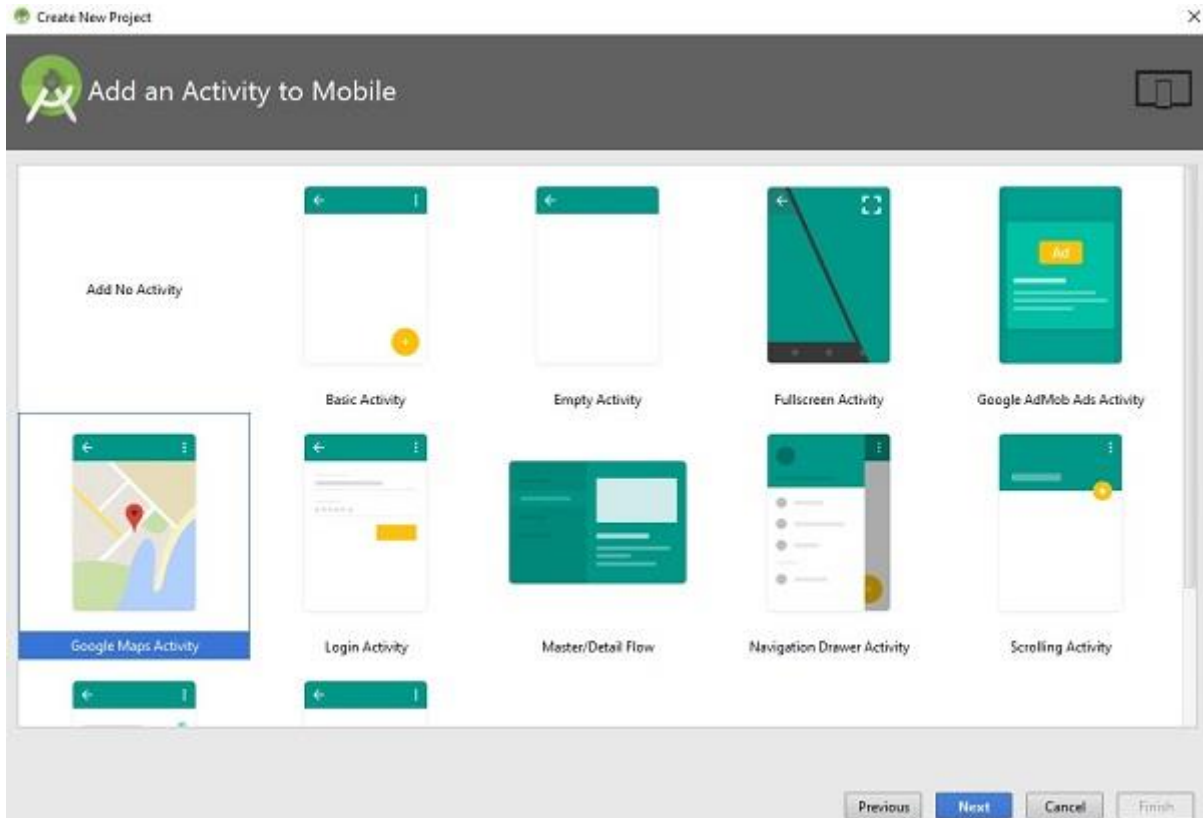


Figure-6.1.1

- **Basic activity:**

This is very simple type of the activity which is easy to use for the user who is developing the first application. This type of activity contains the simple app bar and the one floating action button. This activity also considered as the starting point of any project because this activity contains some by default UI components which helps the user to easily develop their application. Mainly this activity has the two different layout xml files in which one xml files used as activity file and other file used for separating the text contents. (Activities, 2015)

- **Bottom navigation activity:**

This is the most common bottom navigation activity used in most of the application today. Instagram is an application which is using this bottom navigation activity which is in the home page of mobile application where we can find the various navigate options. This activity provides the bottom navigation which helps the user to switch or transfer the most used pages or top-level views in the single touch on the bar. This type of activity is used when user want to create an application which has three or more top level destinations. This activity has the app bar and a single layout file which has a normal design of the navigation pages. (Activities, 2015)

- **Empty activity:**

This is a most common used activity in each types of the application. User uses this type of activity while creating any kind of the designing whether it is a simple UI designing or user wants to use different kinds of components and want to make it bit of extraordinary design. This activity creates simple layout file which is empty and user can design their UI as per requirements. This activity only has the single layout file with some simple texts. (Activities, 2015)

- **Full screen activity:**

Full screen activity basically used for providing the different types of views. Full screen activity provides the functionality to the user to create the application which switch the views. This activity helps app to switch between primary full screen view and the normal view which has the different standard UI controls. In this activity full screen view is selected by default and user can activate the standard view as per requirements by touching the screen of the application. This activity has the app bar and button in standard view but it is not functioning. One single layout file with the full screen view and one frame layout for the standard view components. (Activities, 2015)

- **Login activity:**

This type of activity used to prepare the basic login design for any user's application for make their work easy. Basically, the login screen has the required emails and password fields as well as login button. This activity use Async task for operating and handling network operations rather than using main user interface thread. This activity has process indicators which useful during the network operations. Single layout file with login required credentials like two edit texts for the email and password and one button for the login purpose. (Activities, 2015)

- **Master/Detail flow activity:**

This activity has the list on every items that used in the applications. This activity create an application which has the item list and display the detailed list of each item used in the application. This activity provides the new page with details of the each items when user clicked on the list button. This activity provides the screening layout according to the user mobile device. This application mainly has floating action button placed at each screen, backup resource layout files and toolbars. This activity present the list of item used and activity and fragments used for the detailed description of each item. (Activities, 2015)

- **Navigation drawer activity:**

This is also a most common type of activity used in number of application in the world today. The most popular example of this activity is Gmail application which provides the navigation drawer menu. This activity is nothing but a normal basic activity which is embedded with the navigation drawer menu. The navigation bar expands in the main application screen from the left side to right side on top of the home screen. This activity mainly contains the drawer layout which has details of

menus which user want to display in the navigation drawer. This activity has app bar and floating action button and simple layout file with navigation drawer and basic activity file. (Activities, 2015)

- **Scrolling activity:**

Scrolling activity is nothing but a one type of scroll view use in the application. This is an activity which is mostly used to create an application which has the collapsing toolbars and supports the scroll view for large amount of text in the application. When the user scroll the page down, the toolbar in the screen is automatically started collapsing and the floating action button is suddenly disappears. Mainly it has the collapsing toolbars instead of app bar and it has the floating action button. This activity has two layout files in which one has activity and other has a nested scroll view for the separating the texts. (Activities, 2015)

- **Setting activity:**

Setting activity is used to add setting menu in the application screen. Setting activity used to make an activity which can create the setting or user preference for any applications. This activity is most used in the user design and this activity also extends the preferenceactivity class. It basically has the layout file which used to extend the preferenceactivity class and the xml file which contains the some display setting and feature. (Activities, 2015)

- **Tabbed activity:**

Tabbed activity is a type of activity which is a mixture of some different parts. Tabbed activity is used when user want to make application which has number of features, Usually it is a type of activity which used to create the application which has number of sections, can able to work with swipe navigation and app bar. Tabbed activity included the sections which supports the navigation on both the direction like left swiping and right swiping. This activity has a components like app bar, fragment adapter which is extended version of adapter to create fragment for each section or class. This activity has two layout file in which one is used for activity where the other one is used for the making a fragments for different class. (Activities, 2015)

- **Google maps activity:**

Google maps activity used to insert a Google map inside the user application. This is a most common and used activity today because most of the application in the markets today are based on the maps or navigation which need the implementation of map inside the application. This activity rapidly used because this has the map and map is used to give a perfect location of the user or thing in the application. This activity used to offer Google map with the support map fragment. This activity usually use markers, polygon or polylines to display the interactive information to the user of the application. (Activities, 2015)

6.1.3 Explanation of Activity life cycle:

As it already describe that activity is nothing but a single screen with which user can interact and design the user interface of any application. With the help of activity user can able to put together all the UI components and widgets in the single screen of application. The android activity lifecycle is controlled by the class on android which is called as android.app.Activity class and all the methods of the activity life cycle are protected by this class of android.

The activity life cycle has the different 7 activity life cycle methods which can easily explain the way how this activity will work and behave at different level. The activity which created in any kind of application which usually goes through this different types of the stages which is called as a life cycle.

Any activity of an application has total 7 numbers it the call back methods or life cycle methods which are onCreate(), onStart(), onPause(), onRestart(), onResume(), onStop(), onDestroy(). The main methods of activity life cycle are briefly explained below. (Burton, 2015)

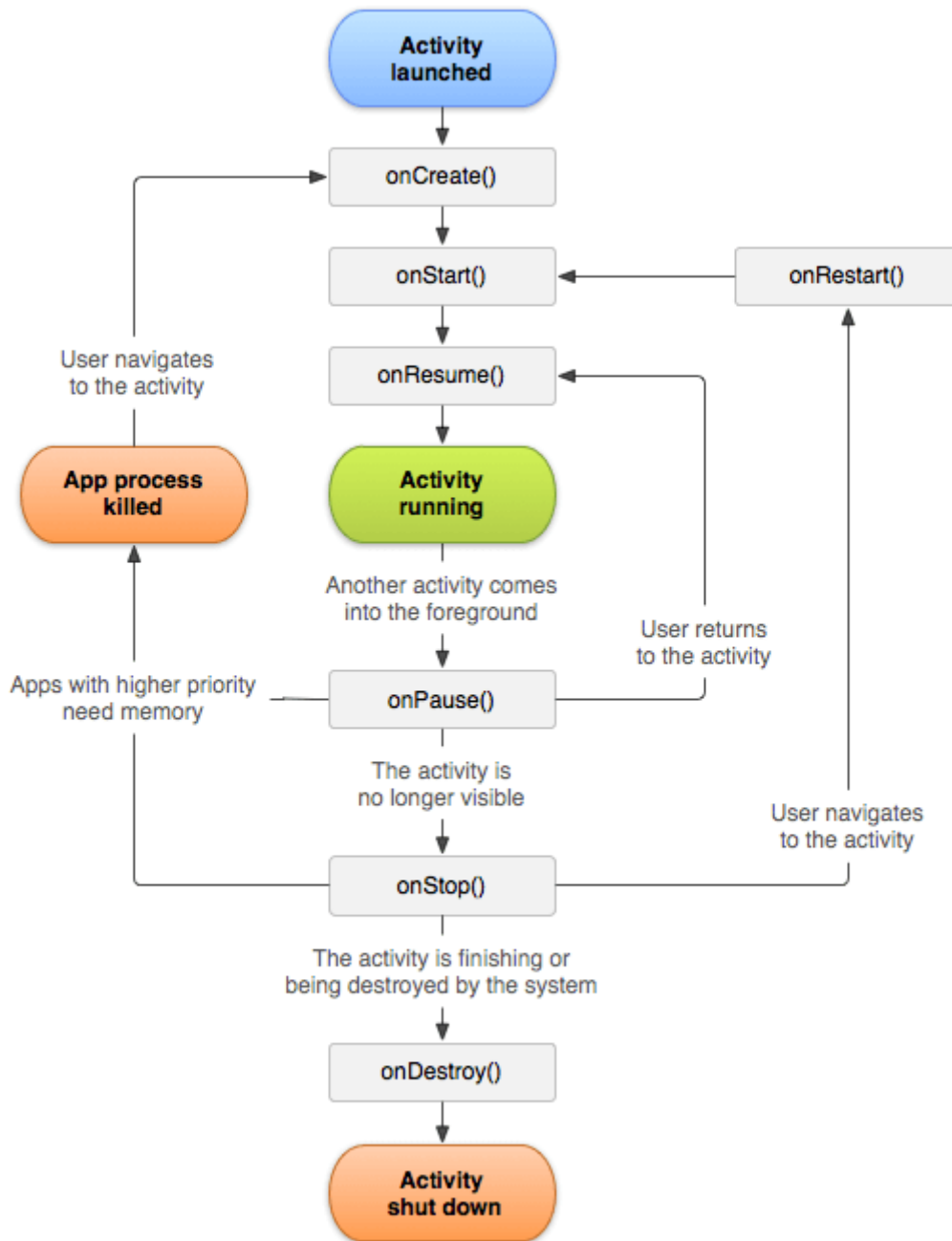


Figure-6.1.3

a) onCreate() :

This is the method called when usually the activity is first create, normally it is called when user open the application. This is the real place which gives user to do all the static set ups like user can do different things like binding the data or create different types of the views and other things. This methods also join with the bundle has some earlier information about the frozen state if they have any. This method is always followed by the `onStart()` method. (Burnette, 2010)

b) onStart() :

This method is always called when the activity is able to visible to the user of the application. This method is automatically taken place when the on/create method is done and activity can be suddenly available for the user. This method is followed by the onResume() method when any activity comes to the foreground. This method is also followed by the onStop() method if it becomes hidden for the user. (Burton, 2015)

c) onResume() :

This is the method which called when the activity will start communicating with the user of the application and already in foreground. This is point when the activity of user application is at the top most level of the activity stack. This is a time when user can start animation for the application or use the different exclusive access devices like camera and so on. This is the method which is always followed by the onPause(). (Burton, 2015)

d) onPause() :

This is the method usually called when activity will not be visible to any user. This is the method which called by the system is trying to start the activity which is already resumed. When the activity is on pause state it cannot able to take any kind of user input and it cannot able to execute a single code. This is used to undone the some parts which was already finished in the onResume() state. This method is followed by the onResume() if the activity will go back in front to use or followed by onStop() if the activity will be invisible to the user for use. (Burton, 2015)

e) onStop() :

This is the method which is called when the activity is no more visible to the user or the user will lower the current application. Normally this method is called because the activity is not visible for user and another activity is resumed and it's covered the contents of the previous application which is already running in the system. This is commonly used to be undone the parts which is already finished in the onStart() state. This is usually called sometimes when the new activity suddenly started and the existing activity is being brought in front or other reason is this activity is destroyed. This is followed by the onResume() if there is some possibility of interaction with the user or it will be followed by the onDestroy() if the activity finished. (Burton, 2015)

f) onDestroy() :

This is method called when you activity is going to be destroyed. It is method which called when activity is going to be destroyed in the system and user clear all the application task. This is usually used when the activity usually finishing or the system is temporary destroying the activity to free some memory in the system. If any activity has thread which running in the background and try to download some data over the internet, it creates the thread in the onCreate() and try to destroy the thread here in the onDestroy(). (Burton, 2015)

g) onRestart() :

This is the method which is called when the activity is stopped and it will be going to restart. This method is not that much common to carry out the call back again in the system. This is the method which is followed by the onStart() method. (Burton, 2015)

6.2 Introducing Android Manifest file:

Manifest file in the android is the file which contains all the information of the android application. All the conditions as well as all the permissions are defined in android manifest file. Manifest file describes the all the fundamental characteristics about the application and each of its components. Manifest is nothing but a one type of interface which handles the communication between the android operating system and user application.

The main important thing is user needs to describe the each of the components of the application in the manifest file otherwise the operating system will not be considered it. Every project in the android has the manifest file which is usually stored in the root directory of that project hierarchy. Manifest file also explained more about the metadata of the applications as well as the components that application includes and the requirements of the application.

This file also includes the basic node for every activity, services, content providers and broadcast receivers which make the application and it also generally use the intent filters and some permission to make sure the way how they will communicate with each other internally and externally with the outside application. Manifest file also has the metadata which is used to define the application launcher, version of application, themes and all. Every application project can have only one single android manifest file. And by providing the permissions in the manifest file user can use location, sms, calls, data storage and other types of services inside the application. (App Manifest Overview, 2005)

6.3 Importance of Android Java file and Xml file:

1) Java file:

Android application mainly contains two files by which any application can be developed that is Java file and Xml file. Java file is usually contains the java source code for the application which is normally used to develop backend of the application. Backend of the application means the java source code usually helps the developer to make application functional by means application can be navigate from the one activity to another activity and this navigation can only defined inside the java source code.

All the functions or UI components everything should be defined in the java source code and by java source code developer make it functional in the application. Also a java files are mostly separated by the java package names by including the JUnit test

code inside the application. Every types of the activities which is used in the application that all activities class are stored inside the java folder.

For example the activity class called mainactivity.java is stored in the directory under the mainactivity package name in the application. In android every activity should inherit from the activity thus the mainactivity is also inherit from the AppCompatActivity which is already inherited by the FragmentActivity and the FragmentActivity is already inherited by the activity so after everything the activity is always inherited by the activity. So mainactivity is inherited by the activity. (Java File, 2005)

2) Xml file:

In android development xml is also most important type of file like java file. Xml is stands for Extensible Markup Language which is mostly used to describe the data. Because of the xml tags that are not defined by default user need to define the tags and this tags are easily readable by the users and machines.

Mainly xml files are used to prepare the user interface design of the application in the android where user can customize the UI design according to the requirements. Preparing a user interface design for any application is very easy inside the xml file where user can use lots of UI components, different layouts and many more functionality and features of the android platform which make application development very easy for the developers.

User can design the layouts according to the requirements and for the navigation or other pages in the application user can use more than one activity to finish the entire application with each working functionality. Also xml is light-weight language so every time xml is used for developing front end of the application because it doesn't make application layout heavy. (Layouts , 2005)

6.4 Layouts for the Android Applications:

The android layouts are most effective part of designing of an application in android. This is the main part which makes the look of the used application design like how it will look and how application will be put all the contents in the application. Generally the layouts are nothing but the subclasses of the view group which normally specify how the view will be arranged inside the different view group inside the application.

Layouts are used to states the visual structure of the user application interface. Usually there are two main ways to make the layout of the application in which user can directly define the UI components in XML file and second one is user can instantiate the layout components at runtime.

There are main four types of the layouts which are LinearLayout, RelativeLayout, FrameLayout and TableLayout. (Layouts , 2005)

1) LinearLayout:

Linear layout is the most common layout used inside the android application which is normally make the design elements in sequence order. This is the layout which arranges its elements in a single column or in the single row of the main application screen. Normally the direction of the elements are set by the `setOrientation()` function.

This is a view group which used to define the all the elements in the dingle direction like horizontally or vertically. This direction is specified by the `android:orientation` attribute in the application. In the linear layout user set the all the elements in the horizontal direction by setting up the orientation horizontal while user can define its all the elements in vertical order by setting up the orientation value vertical. (Goalkicker, 2017)

2) RelativeLayout:

Relative layout is also a common type of the layout which is used most inside the applications designing today. Relative layout usually based on the relation between the different elements in the application. This is the layout where the position of the each elements can be described only by the relation with the other elements or with the direct relation with the parent activity.

So mainly it is the layout which takes the position of element to place another element in the application or take the main parent container as a reference to place the elements in application. Sometimes this is the little hard layout to use because it is difficult to define the relation of elements with each other in the application and finally hard to get the desired layout of the application. Relative layout mostly managed by the main three type which is design according to about to container, design according to relative to other elements and design according to the alignment with the other elements. (Goalkicker, 2017)

3) FrameLayout:

Frame layout is nothing but the display of the frame inside the user application screen. Frame layout is originally designed to block out the particular area of the screen to display the single item. Any application can contain the multiple elements within the frame layout but the each elements must be positioned usually based on the top left of the screen in the application.

Inside the application screen the elements which are overlapped are usually displayed as an overlapping. Frame layout are mostly used for when the elements are hidden and displayed programmatically. User can use the attribute setting and make the layout visible by selecting the proper value. User can also call the `setVisibility` method to display the same results. (Goalkicker, 2017)

4) TableLayout:

Table layout is useful to create a table structure or usually it is more useful to put the elements in the rows and column format. Normally this layout is used to arrange the contents in the rows and columns. In the android application the table layout is work same as the table layout works in the html.

In the table layout the group of view containers do not display the line of border for each rows, columns and cells. Usually the table layout has the as many as columns with the rows and cells. In table layout it is possible that a table can able to leave the cells empty but the cell cannot spams the column as they usually do in the html. In table layout the height and width of the table is mostly carried out by the parent container.

There are number of operations that can be carried out with the columns in the android application which can helps user to design the screen as per requirements. Some of these operations are stretchcolumns, shrinkcolumns, collapsecolumns and some others operations are as well which make perfect user interface designing. (Goalkicker, 2017)

6.5 Multiple UI widgets for Android Applications:

Android UI widgets is also known as UI controls or controllers which usually helps the user to do some actions. Usually it is a kind of user interface element which user used to do some specific action or events by tapping on it.

Normally the input controllers are the interactive components which user use in application interface but the android provides the best version of the controllers which user can use in the application UI which in the form of buttons, text fields, edit texts, radio buttons, check boxes and many other advanced controllers.

User can easily add this kind of the control inside their application design. Using this android widgets user can easily make their user interface design very interactive and responsive and it runs faster than the expectations. Usually this UI controls are commonly used in every applications to do some tasks very easily and reached to the desired destinations.

There are many UI controls in android and some of them are briefly explained below.

- **Buttons:**

Buttons are the most common UI components used in the android. Buttons are the controllers which performs the action when it tapped by the users. Normally there are two types of the buttons in the android which is normal button with texts and the second one is button with images which is called as image buttons.

A standard button is represented by the android class `android.widget.Button`. Usually this button are used in UI by entering in the xml file or programming in the java file. There is also some click events are performed on the buttons which is usually used to transfer the data from one to another activity. (Androids-UI Controls, 2004)

- **Edittexts:**

Edittexts is generally a text fields which allows user to enter some inputs or some texts inside the user applications. In the edittexts user can enter the single line text or texts with multiple lines as per user requirements. Edittexts has a functionality which activate when user touch the text field and cursor is automatically placed on the text field and key board is automatically displayed.

Edittexts also provides the feature of selection of texts like cut, copy, paste and data check up by automatically performing the auto completion of texts. Edittexts also provides the facility of input type where user can choose the type of input what they want to insert in particular edittext. User can set the choice of their input by setting up the attribute called `android:inputType:` and choose whether they want to insert texts or numbers and so on. (Androids-UI Controls, 2004)

- **Image button:**

Image button is nothing but a one type of button with image on it. This is a button which has the image and it gives response when user tap on the image. Image button is represented by the android class called as `android.widget.Imagebutton`.

To use image button user must need to define or add the image in the drawable folder to make use of image button. Image button also has some type of the image click events which is the same as onclick event of button which responds with the user tap. (Androids-UI Controls, 2004)

- **Toggle button:**

Toggle button is nothing but the button with the multiple option. It is a button which provides the change of settings or options during the designing of applications means user can change the option or state in the application.

Toggle button is a normal button that user can swipe and change the mode of any state like user can make application on and off by using the toggle button when user need only a single swipe to change the state from the on to off.

- **Radio button:**

Radio button is button which provides the facility to select the any one type or button from the multiple choice of buttons. Radio button allows the user to select only one single value or option from the large set.

User can use radio button inside the application by defining radio button in the layout and for each option user can give to every radio button. All the radio buttons are usually grouped in to the group called radio group .By grouping all radio buttons together system always allows the user to select only one single button at the time. (Androids-UI Controls, 2004)

- **Checkbox:**

Checkbox is totally different from the radio buttons. Checkbox gives user the full freedom of selection of user choices like user can select more than one option from the large set of options. Usually in checkbox it needs to define each checkbox option inside the vertical layout or list where all the options are listed well.

For use of checkbox user can easily define the checkbox in layout and they can use click listener for each checkbox to get the every value selected by the user in checkbox.

- **Spinner:**

Spinner is also type of the button where user can select the one value from a large option list but it is little different from others. It provides the quick way to select or choose one option from the large set which has options in the drop down menu.

Spinner displays currently selected value when it is in the application because it is always shows the value which is selected by default. Spinner provides the drop down menu in which user can find the large set of option from where user can select only one option which will showed by spinner as a by default selected option.

- **Progress bar:**

Progress bar usually used to display the progress of the task. This is the most common used bar for showing the process or task of any application. Most common example of progress bar is when user downloading or uploading something with the internet user can see process of the task download or upload in the application.

Application also provides the latest update in progress bar when the process or task is move further in the system. Sometimes the progress bar is indeterminate by showing the cyclic view of process without giving the proper information about the process. (Androids-UI Controls, 2004)

- **Seek bar:**

Seek bar is generally extended version of the progress bar. Seek bar is a bit extension of progress bar which include the draggable object to the process. This draggable object can able to move left and right to set the current value of the process.

Seek bar is most common used bar in the android application because it is provide the selection of integer values using the natural user interface.

- **Rating bar:**

Rating bar is extended version of seek bar and progress bar. Rating bar has ability to show a rating in star which normally display the process. User can use the attribute `android.widget.Ratingbar` for displaying the components of rating bar in the star icon in the application.

User can set the number of stars by setting up the layout width as wrap contents. In rating bar the `getRating()` method of android ratingbar class is used to give the real value of rating number.

- **Date picker:**

Date picker is usually used to get dates in android applications. Android date picker used to allow user to select the date inside the application where they can select the day, month and year in the user interface. The `android.widget.DatePicker` is a subclass of `FrameLayout` class in the android. (Androids-UI Controls, 2004)

- **Time picker:**

Time picker is almost the same as the datepicker in the android application. Android time picker allow user to select the time of the day in different formats like 24 hour format or 12 hour AM/PM format.

Timepicker consists of the hour, minutes and the clock format in android application. The `android.widget.TimePicker` is a subclass of `FrameLayout` class in the android. (Androids-UI Controls, 2004)

- **Toast:**

Toast is kind of message usually used for display some message for few seconds. So normally toast is a one type of notification message that is pop up in the application for a specific duration of time and disappear soon. Toast only need the little space that user want to display the message.

It only takes the amount of the space need to display particular message and the current activity which is running inside the application is always stay visible for the user and in interactive condition. The `android.widget.Toast` class is subclass of `java.lang.Object` class.

- **AutoCompleteTextview:**

An `AutoCompleteTextview` is a one type of automated completion of text view which provides the tool to finish the text. This is one type of editable text view which provides the different type of related suggestions to the user when user try to typing something.

Mostly the suggestions are display through the drop down menu where the user find the lot of option related to the enter word inside the edit text and user can select the option from drop down menu and replace the content of edit text.

Android `AutoCompleteTextview` is the subclass of the `EditText` class in android. (Androids-UI Controls, 2004)

- **AlertDialog:**

Alert dialog is a tool which provides the application a platform when user can display more information or helps the user to display something. Usually the AlertDialog is used as a small windows which pop up inside the application to the user to make some kind of decisions or allow user to enter some more information about the application.

AlertDialog doesn't require lot of space inside the application and normally it is user to modelled the events that user must need to give some actions before going ahead inside the applications. (Android - Alert Dialog, 2004)

- **CheckedTextView:**

CheckedTextView is nothing but the combination of two different things used inside the user application interface. CheckedTextView is combination of text view and checkbox which means it is a textview with a checkbox.

Mostly the CheckedTextView is used when user want to include this inside the list view and select the setchoiceMode rather than choice_mode_none inside the application. Android CheckedTextView is subclass of TextView. (Androids-UI Controls, 2004)

- **TextSwitcher:**

TextSwitcher is mostly used to switch the text inside the applications. TextSwitcher is used to animate the label on the screen of the user applications. TextSwitcher usually animates the current text out and animate the new text inside the application when user called the method setText(CharSequence) inside the application.

Android TextSwitcher is a subclass of the Android ViewSwitcher class.

- **ImageSwitcher:**

ImageSwitcher is used to switch the images inside the applications in android.

ImageSwitcher is a function which mostly used by user inside the application when user want to switch smoothly between two images and help user to transform from one image to another by proper animations.

Android ImageSwitcher is a subclass of the Android ViewSwitcher class. (Android - Image Switcher, 2004)

- **AdapterViewFlipper:**

AdapterViewFlipper is mostly operates similar like the way Imageswitcher works inside the applications. AdapterViewFlipper usually used to switch or flip between the different views inside the applications.

AdapterViewFlipper is a way to switches between different views like Textview. Imageview and some other layout and give the smoothly transformation between the different views by the proper animations.

AdapterViewFlipper is subclass of ViewAnimator class in android. (Androids-UI Controls, 2004)

6.6 Menus in Android:

Menu is nothing but a simple collection of items or options for user in android application. Menu is generally used as efficient and friendly way to give the multiple options to user so user can choose proper option as per their requirements inside the applications. Menu is also considered as a simple user interface components which is mostly used in every applications.

Usually user needs this menu to put the multiple options inside the user interface. This menu are usually three types which are options menu, context menu and pop up menu. All the menus used inside the android applications are briefly explained below.

1) Options menu:

Options menu is the main first type of the menu used inside the applications. Options menu is nothing but the primary collections of options or menu items that is used inside the activity. This is the place where user mostly can placed an actions which has a large impact on the user applications means it can includes the email, settings and search and other options.

Mainly in the options menu the items are declared by the main activity subclass or by the fragment subclass. There are also some cases when the items of the options menu are declared by the main activity subclass and by the fragment subclass in this conditions they are mostly joined in the user interface.

In the option menu the items of the activities are always appear to user first which is always followed by the items of the fragment classes. (Burton, 2015)

2) Context menu:

Context menu is another type of menu which is almost different from the options menu. Context menu is also called as floating menu because this menu appears as a floating buttons means this menu appears generally when the user of the application press the item for a long time or press an element for some time.

This is type of menu which generally affects the contents of the applications and sometimes it is also little problematic for the context frame. Context menu can be also matched with the options or the way how right click behave in the windows or Linux.

Context menu is also considered as a graphical control elements by the technical perspective. (Burton, 2015)

3) PopUp menu:

Pop up menu is also different from above two menus because pop up menu shows the all the items in the pop up window. Android pop up menu usually display the menu under the anchor text in the application if application design has enough space for it or else it can be shows above the anchor text in the applications.

Normally the pop up menu display the total item list in a pop up window which is mostly anchored with the view inside the application.

The pop up menu usually disappears when user click other places than the pop up window in the application. Pop up menu also provides menu which is like overflow menu which represents the actions which are directly connected to the content inside the applications. (Burton, 2015)

6.7 Containers for Android Applications:

Containers is normally a tool which is used to group together different types of views in the applications. Normally the containers considered as a view which has the other view. There containers mostly has the different view which are list view, custom list view, expandable list view, grid view, web view, search view and others. All the different types of the views are explained below.

1) List view:

List view is most common used type of the view inside the applications. Mostly list view is used to displays the list of items which are scrollable in the user application. List view mostly grouped the different items together and display all the items in the vertical list. The items in the lists are inserted by source using the adapter class in the application.

Usually the source of the list items are anything like any type of arrays or it could be any databases queries or something like that. The adapter always works as an interface between list view and any type of source.

Adapter fetch the data from the source and allows the user to get access to the all the items in the list view. Adapter is also used to provides the view for each items used in the list view.

Thus, All the items in the lists are inserted automatically by the adapter and that adapter usually fetch the all the information by the general source as a string or database. (Android ListView, 2013)

2) Custom ListView:

Custom listview is used to make the extraa list view by customize the original layout file. In android usually array adapter always takes a single text view for the each row. It is not possible to display more than one view inside a single row.

Usually in the android default layout always has the list which used to displaying the items which always has the single textview for the each row. So mainly the custom listview is discovered to help the user to use more than one view inside the single row by customize the original layout file. (Android Custom ListVlew, 2013)

3) Expandable ListView:

Generally the listview used to display the large group of different items in the form of list. Expandable listview is an expansion of list view to make the user very efficient while grouping the items inside the different lists.

Expandable listview is usually used for grouping the items in the list and helps the user to display it in different categories. So basically it is view which helps user to group the data according to the different categories.

4) GridView:

GridView is almost different from the listview because both have different functionality. GridView is a type of view group which is used to display the items in the form of grid which is scrollable and give the two dimensional view of the items. In the android the grid items are also automatically inserted in the layout using the adapter.

In GridView user can easily select which type of the grid item they want inside the application. Usually the list adapter used to insert all the items inside the layout.

5) WebView:

Webview is different from all other views because this provides the functionality to insert the web in application. So it is basically related with the web and helps user to enable the web inside the application.

Webview is type of view group which is used to insert different types of web pages inside the user application and helps user to make it web application.

Basically the web pages used inside the application are loaded from the application itself or any type of URL which give the web pages. Mostly Webview is used for give some online contents in the user application and it use tool called webkit engine to display the web pages inside the user application. (Android WebView , 2013)

6) SearchView:

SearchView is normally a view group which allows the user to get access to the search options inside the application. Usually it is a view which give the user interface for the user to enter some types of the search query and allow user to send the search request to a search provider.

It is used to give the different types of query suggestions or some results to the user from which user can choose proper suggestions or results as per the requirements. When user wants to use searchview in action bar as action view user must need to define it as a set to iconified by default inside the android by the appropriate method. (Android SearchView, 2013)

7. Messaging Object and Display of Information in Mobile Application:

7.1 Intents in Android Applications:

Intent:

In android intent is used as one type of object or component which is usually transfer the message between the activities. An intent is a messaging object which is helps user to ask any kind of the action from another application components. Usually it is also used as an interface to transfer the user from one activity to other.

Intent provides the way to user to make connection between android application components to ask any kind of the actions between the different components. Intent is a message which most of the application in android listens and do the operation as per the requirements and do some specific action between the components. Usually intent use components from the same application or communicate with the other application.

Android also provides the chooser which provides the option from which user can choose the proper option to perform the operation because in some case there is a lot of possibilities that there are multiple application which can give the response to user messages so by chooser user decide the application by which user needs to do some particular operations.

There are multiple advantages that intent provides to the user that user can use intent to start any activity or user can use to start any services or user can deliver any broadcast by using the intent. There are mainly two different type of the intent which is implicit intent and explicit intent. Both the types of the intents are used to transfer the data and message between the activities and both are explained briefly below. (Burton, 2015)

a) Explicit intent:

Explicit intent is the first type of intent which is used to pass the message in the applications. Basically explicit intent is the type in which the targeted application or place or part is always known by user.

This is the intent type in which user always defined the target component name during the creation of any object intent.

When user specify the targeted components such as any class name or package name it is known as explicit intent. In explicit intent user must need to specify the navigate class or package to do intent in the application. (Burton, 2015)

b) Implicit intent:

Implicit intent is another type of intent which is totally different from the explicit intent. Like explicit intent in implicit intent it is not necessary or it doesn't require to specify the target components while creating the intent.

This is the type of intent which only need some information to perform the intent and they doesn't need to define any class or package name.

User can use implicit intent by giving the enough information to system so system can recognize which is the best part to do the particular action or intent in the application. User can also insert the URL and ask the system to find out the best part to run the given URL. (Burton, 2015)

7.2 Notifications for Android Applications:

Notification is the best feature that is provided by the android today in every devices. Notification is nothing but a type of message that gives some information to the user of the application about something that happens or just happened inside the device.

Generally notification is one type of message that display to the user outside of the any application's user interface. Normally notifications are used to offer some information about some events in applications. When the notifications are displayed for the first time they are displayed inside the notifications area with contain particular icon of the application.

User can later check the all the details of the application notifications by dragging down the notification drawer and view it's all the contents. User can also create the notification in the application by creating the NotificationCompat.Builder class and set the limits of the notification object like setSmallIcon, setContentTitle or or setContentText to make a good format of the notifications. User can also set the priority levels of the notifications to make it more visible to user of the application.

Mainly there are certain types of the notifications used in the applications like inbox style notifications, big text style notifications or big picture style notifications. All the types have totally different view to display the notifications to user which is briefly explained below. (Android - Notifications, 2004)

a) Inbox style notifications:

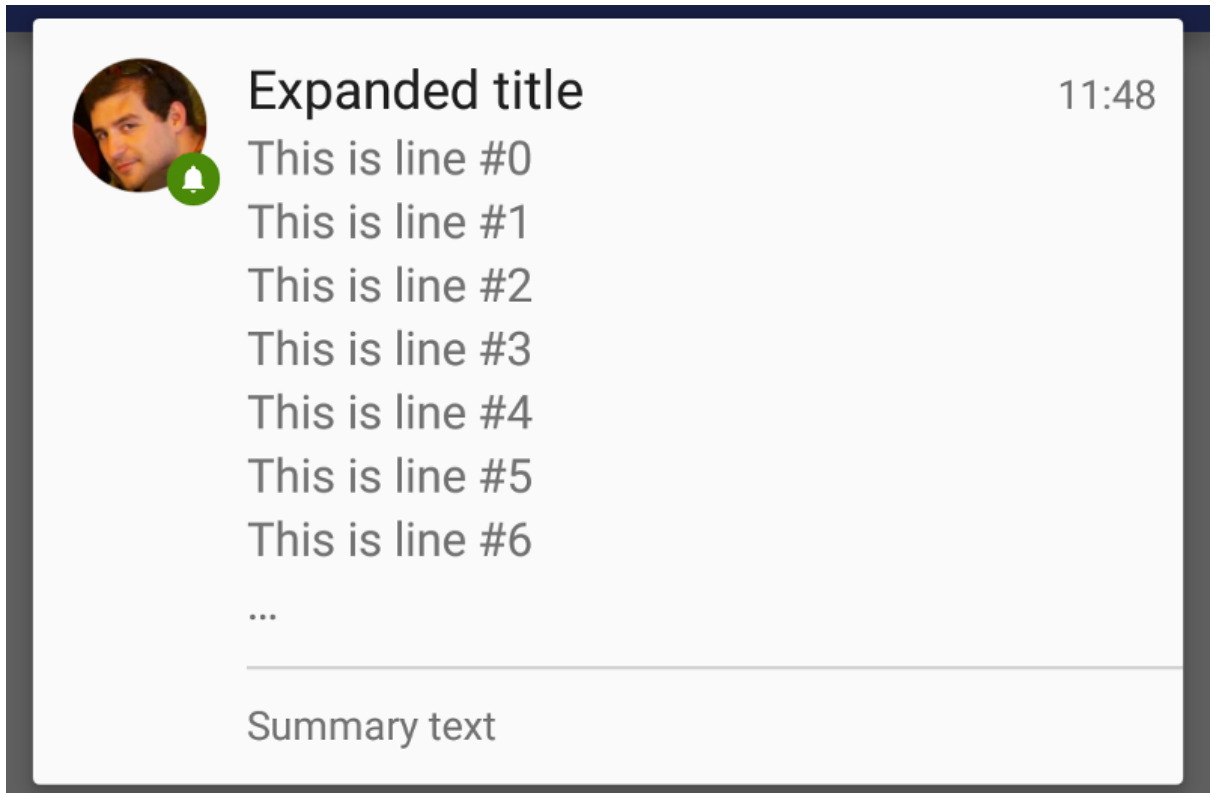


Figure-7.2.1

Inbox type notifications are most used types of the notifications in the applications. It is a type of notification which is used in the application when user want to display only some texts lines. Inbox type notifications allows user to make notifications which has some independent lines or some short text.

Mostly Inbox type notifications usually has the as many as seven lines and later on if the notifications would contains the more than seven lines the notification will ellipsize the further lines by displaying the " " symbol. In this type of notifications title, text and small icons are mandatory to display the notifications in the application. (Android - Notifications, 2004)

b) Big text style notifications:



Figure-7.2.2

This is also a most used type of the notifications used in the android application which has the large notifications. Big text style is used to display the large quantities of text inside the notifications which helps user to read all the notifications.

This is a notification type which maximum contains around 450 characters in the display which is a very large number of text. The remaining text of the notification is displayed as trimmed without the use of ellipsis. In this type of notifications title, text and small icons are mandatory to display the notifications in the application. (Android - Notifications, 2004)

c) **Big picture style notifications:**

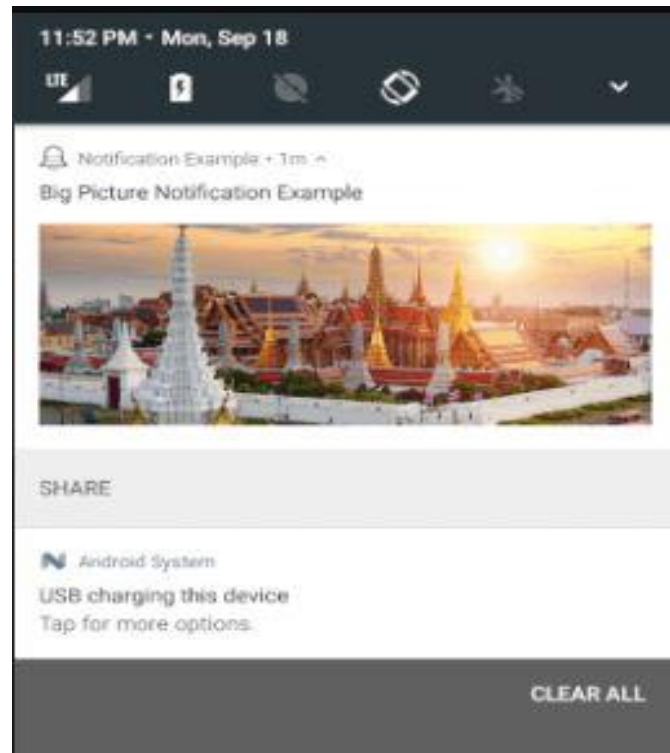


Figure-7.2.3

Big picture style notification is the last type of notification which is almost used in every application which can transfer the image files. Big picture style notification is mostly used to display the picture inside the notification. User can display the image content inside the notification by using this and user can only display the image according to the size of the screen.

Usually the limits of the displaying the image is depends on the screen of the mobile device and it can covered the image up to length by 256dp height. For the tablets the image length is according to the 2:1 ratio of the tablet screen.

In the notification the rest of the image is cropped automatically by the system. In this type of notifications title, text and small icons are mandatory to display the notifications in the application. (Android - Notifications, 2004)

8. Different types of Data storage For Android Mobile Applications:

Android data storage usually helps user to store or save the application data inside the user application. Android data storage provides the several options to the user to save the persistence data of application.

There are multiple options available for users to store data of their applications and the selection of the storage option where user want to store the data is absolutely depends on the requirements and the needs of the application. There are some options where user stores the data which is may be Shared preferences, internal storage, External storage, SQLite databases or cache storage.

All these types of the data storages are mainly used to store the user application data and serve the data back to the application of transfers the data when user need somewhere as an input or for some other reasons. These data storages are mostly behave different from each other and these all types of the storages are briefly explained below. (Burnette, 2010)

8.1 Shared Preferences in Android:

Shared preferences are the most common type data storage used for the application and it is the simplest method available in android to store the data. Usually shared preferences helps user to store the activity preferences and helps to store the private primitive types of data in key-value pairs.

Shared preferences always stores the data in form of key-value pair. There are different types of the data that can be stored like any primitive data like booleans, floats, ints, longs, strings etc. It is the most common type and when user want to retrieve the data from the application user needs to insert only key to the databases and then databases return corresponding value to the user from databases.

For example when user stored data like “Username” : ”Yogin” and “Password” : “12345” and when user wants to retrieve the data user only needs to insert the key to the data base like “Username” and then database gives the corresponding value “Yogin” from the database. (Android - Shared Preferences, 2004)

8.2 Different Storage options for Applications:

8.2.1 Internal storage:

Internal storage is another type of the storage option which used to store the private data on device memory. Android offers the feature that user can save the file directly on the any device internal memory.

Mostly the files which is stored in the internal memory usually they are private to the particular application and it doesn't allow other application to use that files.

The files which stored in internal storage are disappear when user removed the application. There are different types of the data that can be stored in the internal storage and they are normally text files, image files, video files, audio files and etc. (Android - Internal Storage, 2004)

8.2.2 External storage:

External storage is also a kind of storage used in android which helps user to store the public data on the shared external storage. In android like internal storage user also can able to save and read the data from the device external memory like such as from the sdcard of the mobile device.

Mostly the all files which is stored in the external storage is allow the user to read it globally by the any user. And user can also modify the files in the external storage when user enable mass storage to transfer the files on a computer.

External storage is mostly used in the case when user want to share the files between the different applications or user want to give access to files when using it inside the computer.

8.2.3 Cache storage:

Cache storage is nothing but a one type of the area with high-speed which is almost reserved section every time inside the main memory of the device or on storage device. Mostly the files that are stored by user in the internal storage is mostly deleted by the user otherwise it would always exists in the memory and occupy lot of spaces.

Usually the cache data which is stored in the application is not a permanent data stored in the application. Cache data usually removed by the system when system need some extra memory to store some other files.

Android always proves to cache some data rather than directly save it permanently. The cache data usually stored in the internal storage or external storage in android. (Data and file storage overview, 2005)

8.3 SQLite Database for Android Applications:

SQLite database is very simple way used to store the data or structured data in the private database. Generally all the data that is stored is in the form of the related tables. SQLite is nothing but a type of open source database structure which is generally used to store the data to the text file on a device.

Android always contains the in-built SQLite database which helps user to store the data of the application. SQLite provides the functionality which supports the all the feature of the relational databases. Usually SQLite is very light weight database which is always comes up with Android OS.

As SQLite provided by the android the user don't need to install or setup any kind of the additional databases. SQLite always supports some relational databases functions like inserting SQL syntax, transactions or prepared statements. SQLite also provides some operation on the databases like SQLite insert for inserting value in the database, SQLite read for reading value in the database, SQLite update for updating value in the database, SQLite delete for deleting value in the database. (Burnette, 2010)

8.4 SQL databases for Backend development:

Android applications mostly use sql databases for the backend development. Sql is main essential part while creating a backend of any android mobile applications and most common used backend language is php. Mostly the android applications call the php scripts to do several operations to make a connection with the Mysql databases.

In android platform generally all the admin panels or admin sides are developed by the Mysql databases which is mostly operated by the php scripts. This php scripts are used to creates the web services which is used to send the data between the cross-platform like mobile application and the admin panel. Usually the Mysql databases has some tables which is key object to make connection with the applications.

First required thing is to download and install the local web server called WAMP or WAMPP. Normally user need to create one database and some tables in the web server. User can create one file called conn.php to make a connection with the databases. User can also verify the connection by going into the local server's web browser and run the conn.php file.

After successful connection with databases user also can create the login file which is the main file which makes user to use the application by writing some php scripts and create the multiple files in the same way.

Mostly all the files which are created in the database are used to parse the data from the application means application can use that files to send some data in to the backend or admin side which is generally called as web services.

All this files are usually stored at local server while developing the application and every time user can check whether it is working correctly or not by running it in local web browser.

Once all the files are created and application are developed user can put the all the files on live server or in online server to make the application live and every user can send the data live on server.

By uploading the all the contents of the local server on online server user can send the data online on server means user can use that application to send the data live on server. Admin can view that data online that is inserted by the user of the applications.

9. Practical Example of Mobile Application Development on Android Platform:

9.1 Idea of an Example Application:

This part has a practical development of one example application which mostly used the above stated android application development information. Here is one example of android application which is normally developed to understand the functionality and features of the android platform.

The example application is named as Fire Safety System which is kind of application which used in the fire emergencies. This is application which is developed for helping the people in fire emergencies.

The main goal of this application is to provide help and emergency services to help the people in case of fire by providing them online fire vehicle availability and with the live navigation. This application provides user a key feature that they can book any fire vehicle from the nearby fire station by opening a map inside the application and they can book any vehicle for instant help and they can see the live navigation of the vehicle to them so they can easily estimate the time period for vehicle to arrive to that place.

This application is an android mobile application which is simple to use and which supports in android mobiles and tablets. This application needs common login with mobile number and password to use this application and after login they can easily use this application and click the fire button to book any fire vehicle in the main home page. The users which are the first time using this application they must need to register them to use this application and they can only register using the mobile number.

Once they will register user can fill their profile and go the main home page where they can see the home page as well as more about the application.

Also this application provides some other functionality to user like user can view and edit their profile, see the history, ask for some help in this application and some contact us pages and one more that they can give some advice and feedback to the application. There is more detailed information about this application and the more brief information about the functionality are explained below.

9.2 Problem definition:

Create a Fire safety system application as a help gateway for the users who need an instant help during fire emergency and the application that consist of basic functionality of fire safety and some additional features that make it worth using.

9.3 History of Technology and work-done:

✓ About Tools and Technology

- Tools: Android Studio.
- Technology: Java, XML.

✓ Brief history of Work Done

Till now I have created a fragment that consists of user login and registration for the first time application user. Any user can make use of application for fire emergency at any time by just entering the username and password for the application. Or user can make use of application for the first time by registering to the application. All the data stored in the application database. Admin can get access to all the data of user of the application. Application generate OTP for the user registration for the authentication purpose. Without that user do not have any access to the application.

Custom notification has been designed and implemented with all handling. As soon as user click a generate OTP for registration user will be notified in notification area. User can enter received OTP from there.

9.4 System requirements and constrains:

✓ User Characteristics

The user can be anyone who is using an android phone on API level 15 (android 4.0.3) and higher. User must have android phone and the app must be installed in it.

✓ **Hardware and Software Requirements**

Hardware Requirements: Android device with minimum 100MB space and 256MB RAM

Software requirements: API level 15 and higher i.e. android 4.0.3

✓ **Constraints**

- **Hardware Limitations**

Only android device or android tablet can use this application.

- **Interfaces to Other Applications**

As off now it do not interact with other applications.

- **Parallel Operations**

User can do any other work in parallel to the application but user cannot perform more than one operation of application at same time.

- **Higher Order Language Requirements**

Developer requires android to develop the application.

Users do not need any higher order language.

- **Reliability Requirements**

The system is reliable as it has been tested may times. If user finds any reliability issue than user can contact for help and support.

- **Criticality of the Application**

Many applications are available in market which offers different features but the application requires all required features at one place.

A clean and user-friendly UI. So to give basic required features it was required to create a new application.

- **Safety and Security Consideration**

As this application use user authentication via registered mobile number is required for the safety and security consideration.

9.5 System analysis:

a) Study of Current System

The current system is working prototype and fine but there are some missing features in every application that a user requires. Some application provides clean UI and basic functionalities. But user need more features. Some application provides many functionality but they forget to give clean UI and basic functionality.

Paid apps that contain all features but user do not want to spend money on those apps.

b) Problem and Weaknesses of Current System

The current system provides much more functionality that is useless to the user. It is too complex for a new user to manage in the application. Some missing basic features that are mandatory but not available. Paid apps offers every function but it is paid.

c) Requirements of New System

- User Requirements

Application with clean and user-friendly UI.

All basic required features those are mandatory in Fire safety system.

- System Requirements

Android device with API level 15 and more. Android version 4.0.3 and higher.

9.6 Activity/Processes in the application:

| Event | Description |
|-------------------|--|
| Open Activity | User open activity from android mobile phone and application installed in it. |
| User login | After opening application user has required to login for access the application. |
| User registration | New user required to register itself to going further in the application or for using the application it is necessary. |
| Post | User can post at the time of the emergency for help. |
| Map | User can see live navigation of the fire vehicle. |

9.7 UML diagrams and screenshots of the application UI:

Use case diagram:

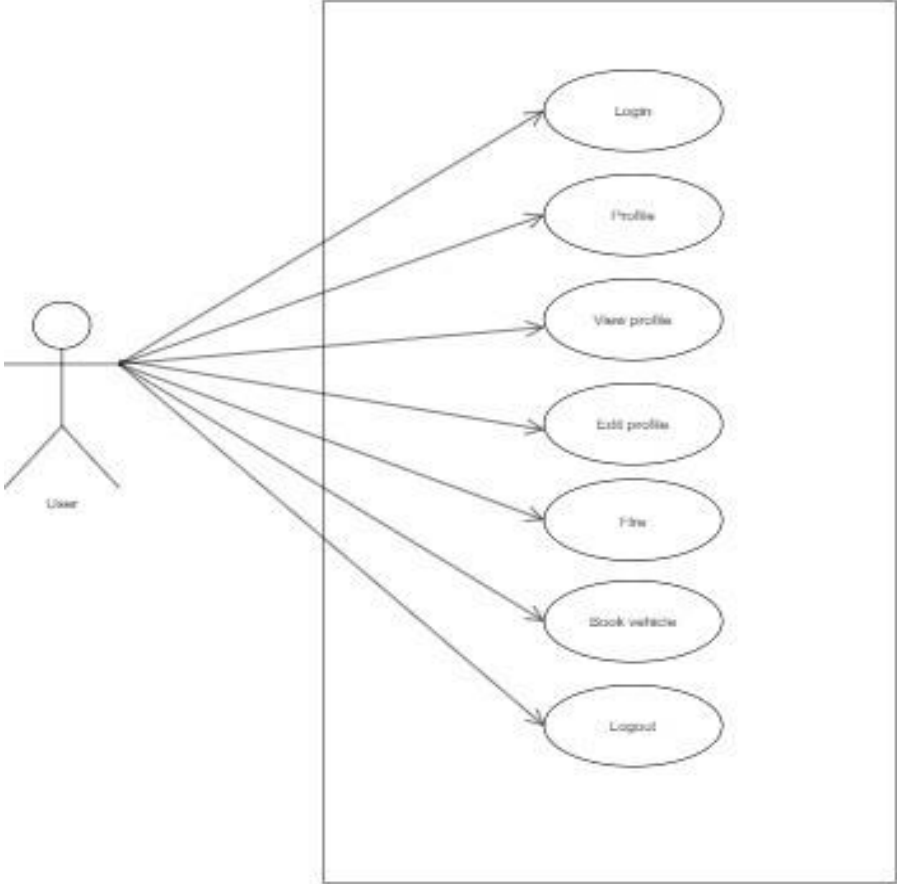


Figure-9.7.1

Class diagram:

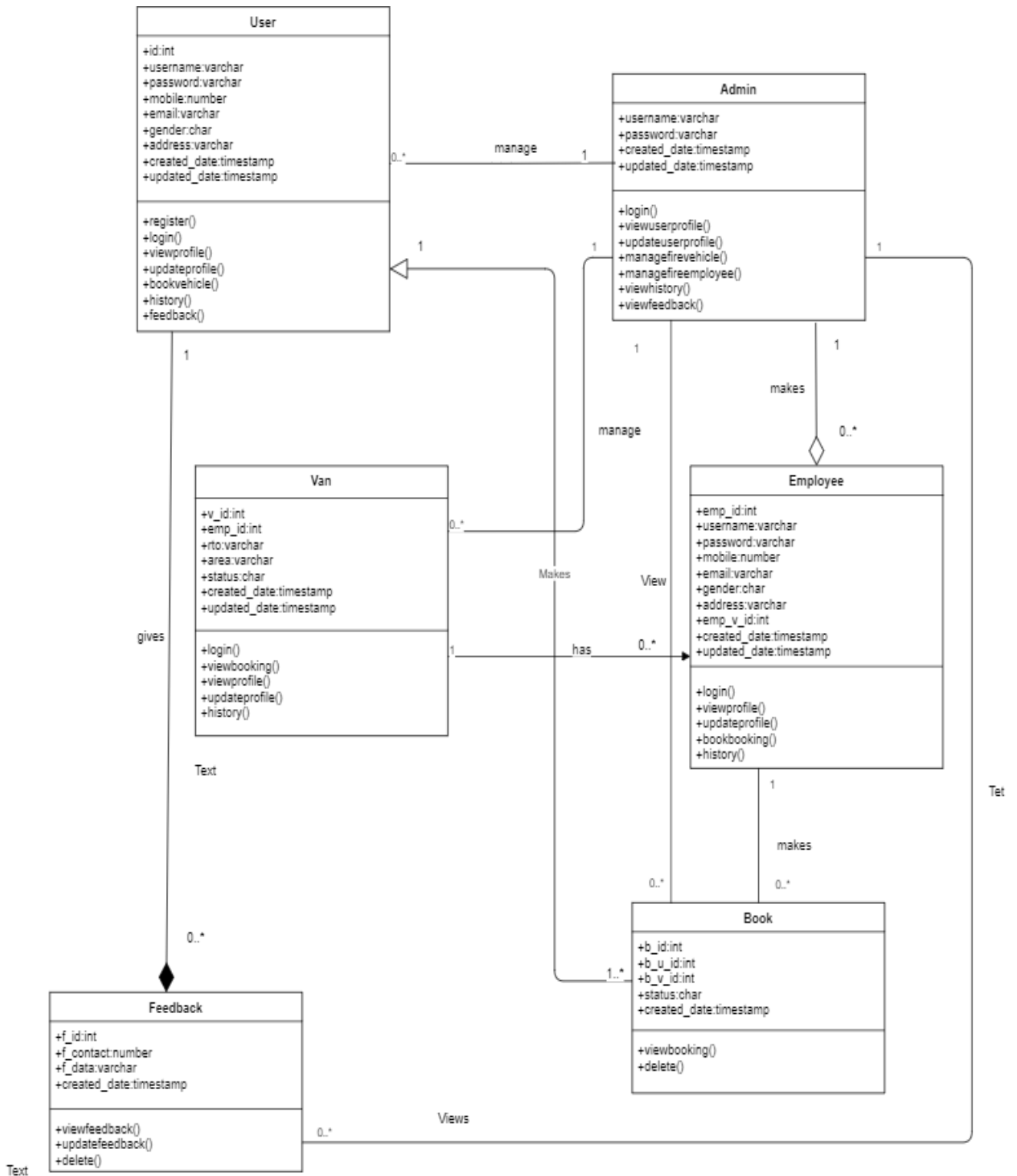


Figure-9.7.2

Screenshots of the UI designing (Application overview):

1) Starting screen:

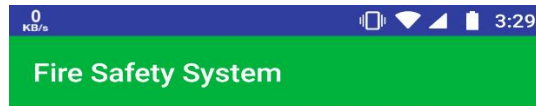


Figure-9.7.3

This is the main starting page of the example application which user can see when they open the application. This page is also called the front page or landing page of the application which is only appears for the 3 to 5 seconds and after that application automatically takes the user to the main login page of the application.

Every application has this page when its start and this page only appear for once and later it disappears till next time when user open the application for the second time after closing it.

This page is not connected with the other pages inside the applications and this activity doesn't has any other navigation link or way to navigate. This is the only activity which runs automatically when user starts activity and rest of the activities are connected internally.

2) Login screen:

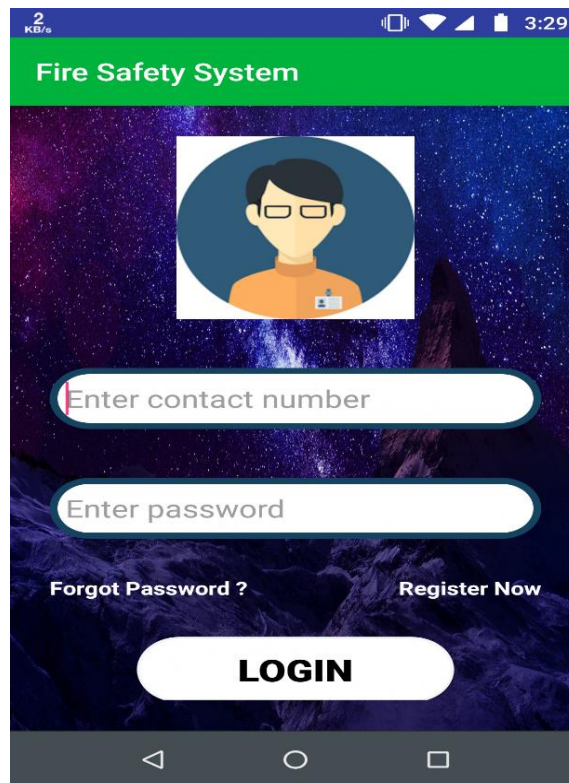


Figure-9.7.4

This is the login page which comes automatically after the starting page. This is the main page of application which allows user to use this application in the emergency purpose for instant help by login the application and directly select the fire option in the home screen. This is main navigate screen which allows user a multiple options from which user can select the correct option according to their needs. User can log inside the application by entering the mobile number and password of the application and press the login button to go to the home page.

If the user is a first time using this application then they must need to register them by click on the register now and user will be navigate to the register screen. Login screen also provides the functionality of forgot password by means if user forgot password of this application they might have chance to get the new password by clicking on forgot password which transfer the user to the forgot password page which send the password again by entering the mobile number of the registered user. This screen directly send the registered user to the home screen by login with mobile number and password and it will transfer the user in to register page who is using this application for the first time.

3) Register Screen:

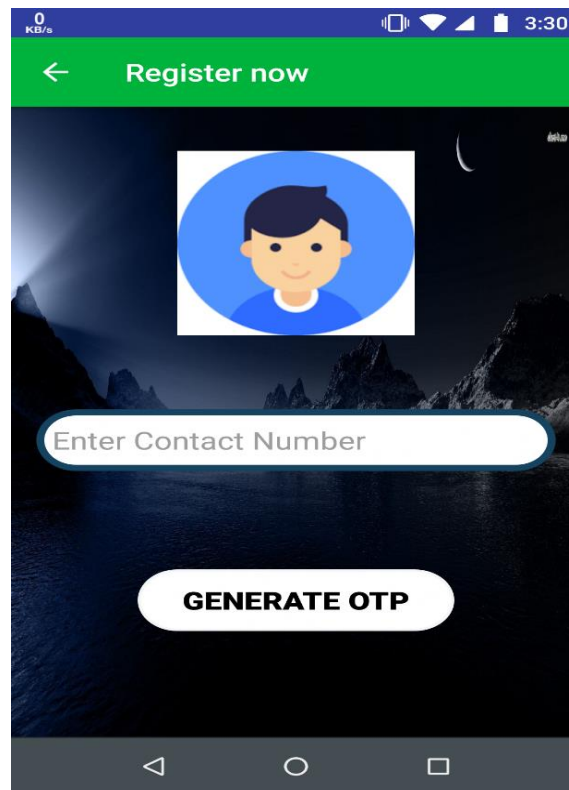


Figure-9.7.5

This is the register page which appears by clicking on the register now button in the login screen. This page is useful for the registering the new user means the user who is using this application for the first time they can register themselves here.

Registering process is very simple as they can enter their mobile number which they want to register with application and they currently using that number and after entering contact number user can simply press the generate OTP button which generates the unique one time password – OTP for the user and send that OTP on this mobile number which is registered with the application.

Here this application is using the one time password – OTP for the authentication of application that means the user authenticate and user is only person who is registering their mobile number in the application and by entering the OTP user want to confirm the authentication so there is no chance that somebody can misuse of this application.

Once user press generate OTP they can wait for sometimes and in few seconds user will received auto generated password from the application to their entered mobile number to go ahead in the registering process.

4) Verify OTP Screen:

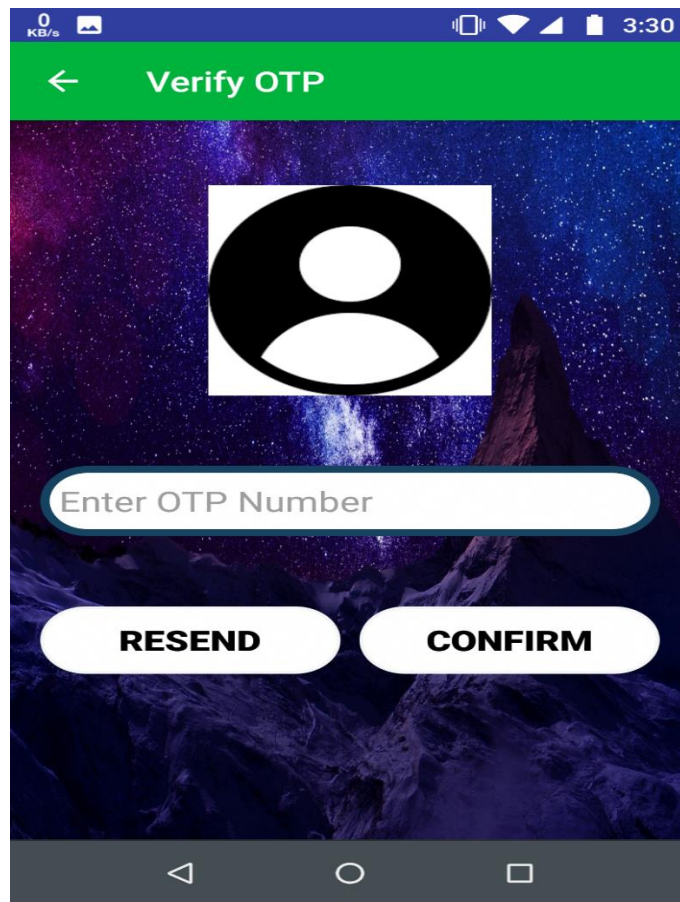


Figure-9.7.6

This is the screen which is called verify OTP screen where user will be navigate after pressing the generate OTP button in the register screen. This screen is navigated by the register screen and this screen providing functionality of confirming the OTP and resending the OTP for the registering process.

When user get the OTP they can enter that OTP in the box and confirm it so user will be moved to next and final stage of the registering process. If user will not get OTP for some reasons application provides a chance to user to get a OTP again from the application by the resend OTP button and user will get new OTP again for the registration and user don't need to enter the OTP again in the application.

Once the OTP will confirm by the application it will navigate the user to the make your profile screen which is the last step to complete the registration for the application and then user will allow to see the main home page of the application.

5) Make profile Screen:

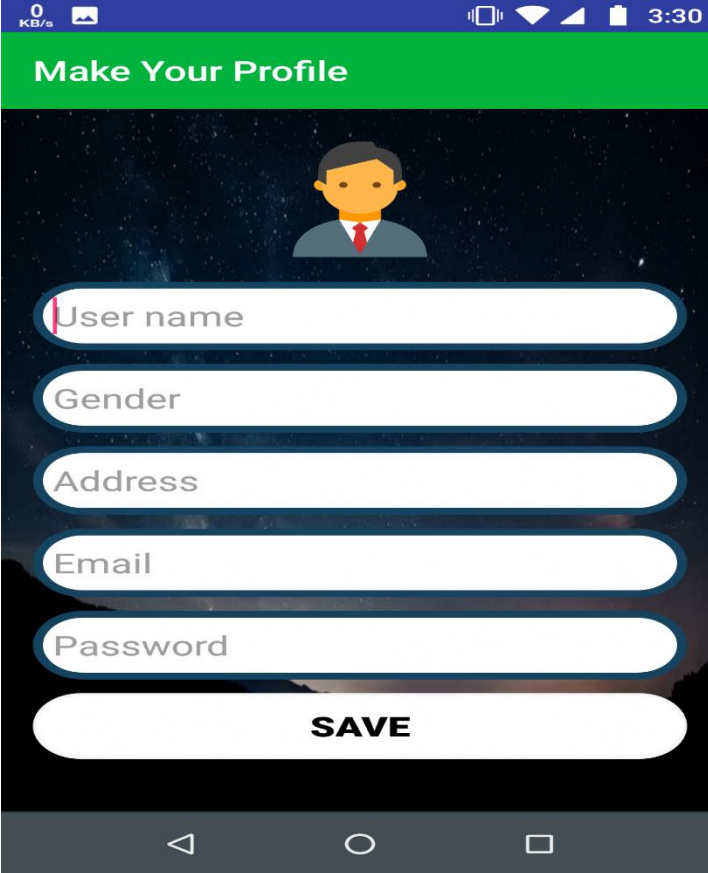


Figure-9.7.7

This is the screen which is called as make your profile screen and this is the last step for the completing the registration process. After confirming OTP user can add some personal information themselves to make their profile in the application. Application needs user to fill some information like username that user wants to name, gender of the user, address of the user and email address of the user so application can easily contact the user in some case of emergency.

User can also set their password here which is going to use again when user will going to login again in the application next time. Once user will save all the data that user insert it will be saved in the database so it will be no longer available for the other user.

User can also view this details in the home page which will come after this screen where user can see all this information in the profile menu and user can also edit some of this information.

6) Home Screen:

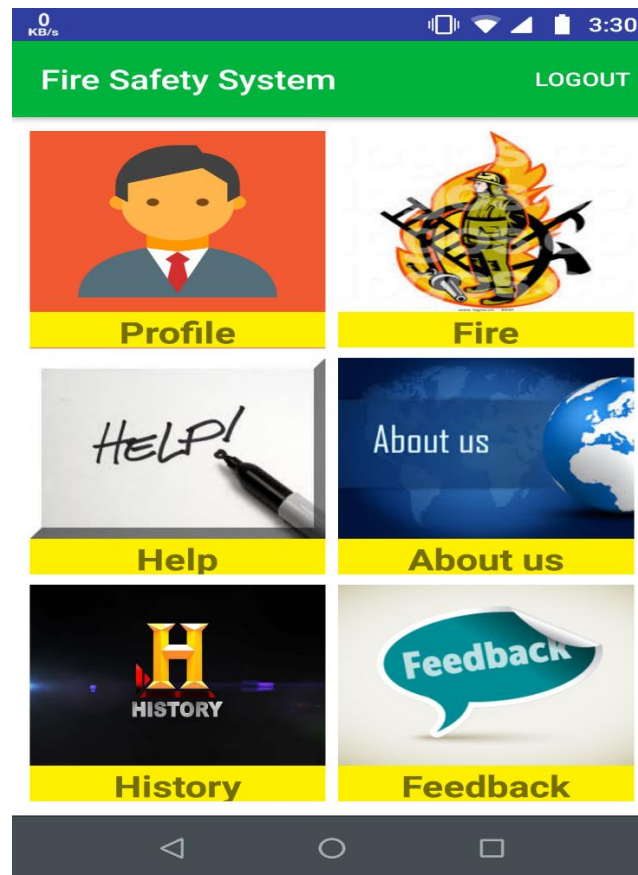


Figure-9.7.8

This is the main home screen of the application which provides user a number of different functionality. This screen is navigated by two different places like first user who is registering for the first time they can fill their profile details and when they submit the details user navigated to this home screen and second is by main login screen user can insert mobile number and password for login and after login they can directly see this home screen. The application home screen provides multiple options to user from which user can choose any suitable option according to the requirements. Home screen has profile option to look into user profile and make some changes if user want, it has help option where they can find some contact details for any kind of help in application, about us option gives some information about the application to user, user can view their history in history option, for some comments and suggestion user can user feedback option to send some feedback after using application and the main fire option provides emergency help where user can see the map and available vehicle which they can book for help. Home screen also has logout functionality for user which user can use to logout from the application after using it.

7) View Profile Screen:

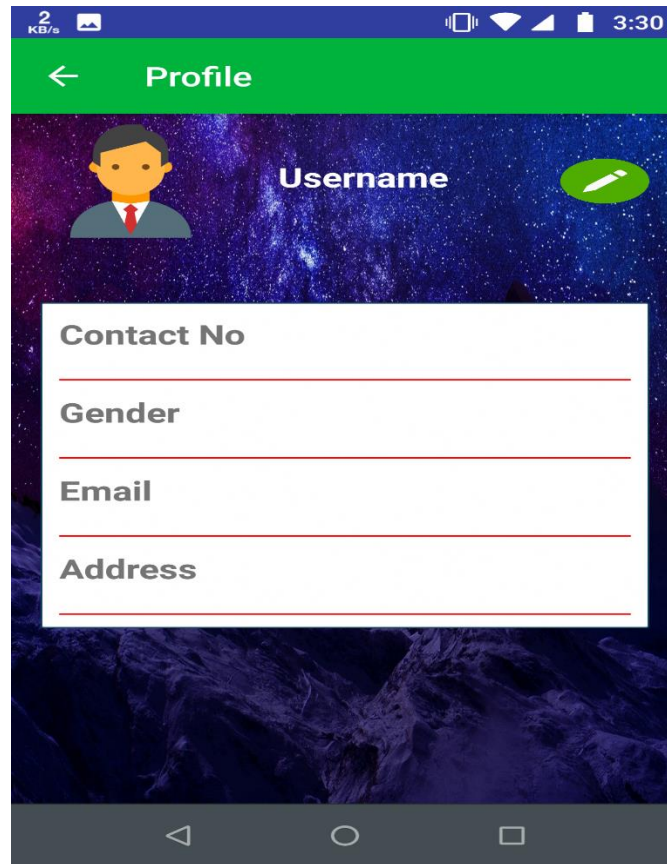


Figure-9.7.9

This is the profile screen where user can see their personal information that user entered in the registration process. User can find the username, entered contact number, gender, email address, and user home address in this view profile screen. But due to lack of live server connectivity and backend also because it is working prototype application unable to view the user information here.

This screen has also one functionality which helps user to edit the information or change some information that user entered in the application during the registration. User can find the pencil icon just right beside the username which helps user to change some the information by clicking on it and it will transfer user to a new screen which has all previous details from which user can change what user want to edit.

The navigated screen is edit profile screen where user make some change and submit it and that screen will be followed by this screen when user click the pencil icon.

8) Edit Profile Screen:

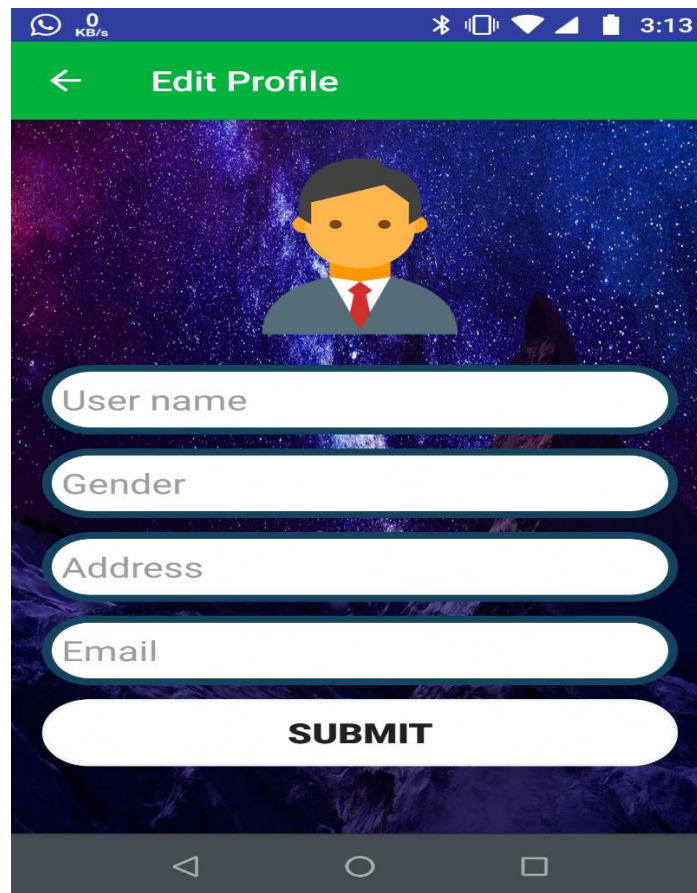


Figure-9.7.10

This screen is edit profile screen which is navigated by the view profile screen when user click the pencil icon and it has all the information which user can see in the view profile screen. This screen carries all the information to make it easy for user to edit some of the information that user want to change in the profile.

In this screen user can see the username, gender, home address and email address from this user can edit the required information. But due to lack of live server connectivity and backend also because it is working prototype application unable to view the user information here. User can see mobile number in the view profile screen but user can not able to see in this edit screen because that is the only way for authentication of user so application doesn't allow user to edit the contact number once user registered in the application with that number. After editing all the required information user can click submit button which save all the information that entered by the user now and store it in the database. After clicking submit button user will automatically navigate to the main home screen of the application.

9) Fire Screen:



Figure-9.7.11

This is a screen which user can use in the case of fire emergency by clicking on fire button in the home screen and user will be navigated to this screen which is a Google map. The main functionality of this screen is that it will take a latitude and longitude of the user position and it will show the map according the coordinates of user. According to the user location map will display the nearby fire station and vehicle which is nearly available to user location. User can click on any available fire vehicle which is very near and the vehicle will be booked for the user and user will see it in the history options. Once user booked any fire vehicle user can also see its live navigation to user location so it will help user to estimates the arrival time of the fire vehicle to location and help user to resolve the fire emergency accordingly. This screen has very simple functionality to use for every user and they can get the instant help by using this application. But due to lack of live server connectivity and backend also because it is working prototype application unable to show the nearby fire vehicles in the Google map.

10) Help Screen:

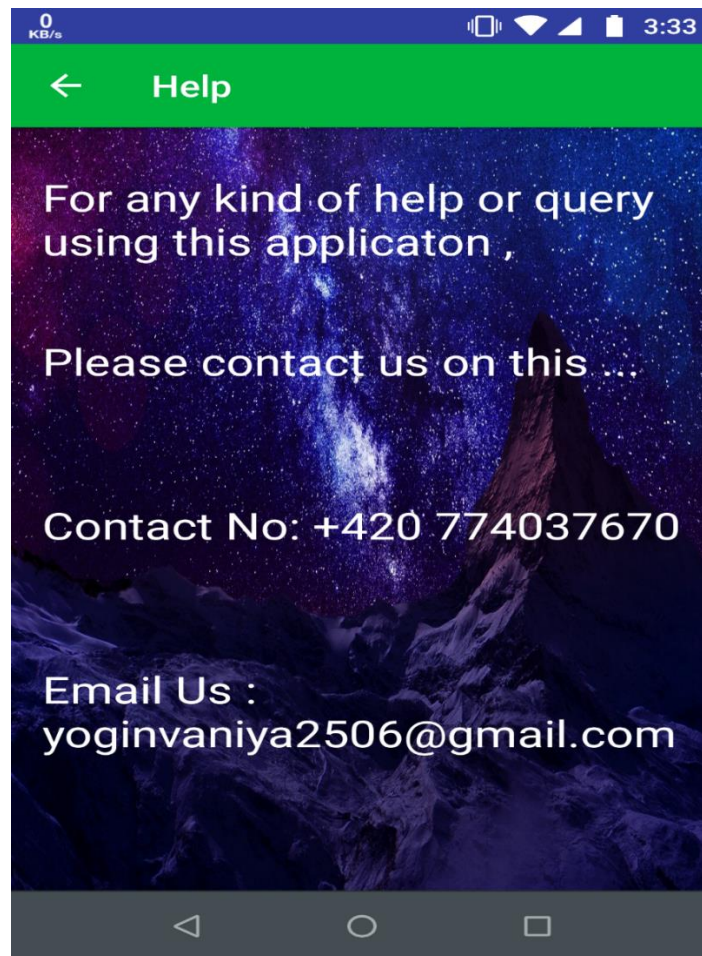


Figure-9.7.12

This screen is known as help screen which is also available in the main home screen menu. This screen provides the help function to the user of this application which allows user to contact application for any kind of help related to using of this application.

This screen has important contact number and email address which user can use to contact or ask for any kind of help. Available contact number is a help line contact number for this application where user can call and ask for urgent help.

Application provides also an email address which is helpdesk email where user can write certain queries which will be solved by the system and user will get the response about the queries from the application as per early convenience from the support staff.

11) About us Screen:

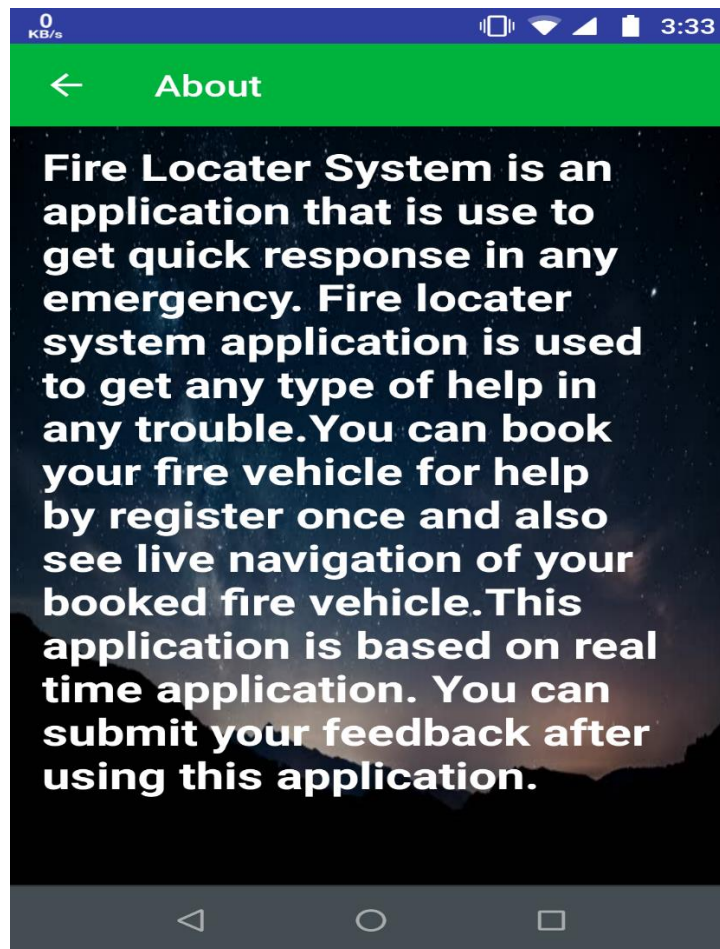


Figure-9.7.13

This screen is about us screen for the application which is available in the main home screen of the application. This screen is used to provide some information about the application and the feature of the application to the user.

This is the option where user can get more information about the application and also user can get some tips like how to use the application or how user can book fire vehicle in urgent help or some other information.

This screen also provides information about different features of the application which user can use for different purpose in the application.

12) History Screen:

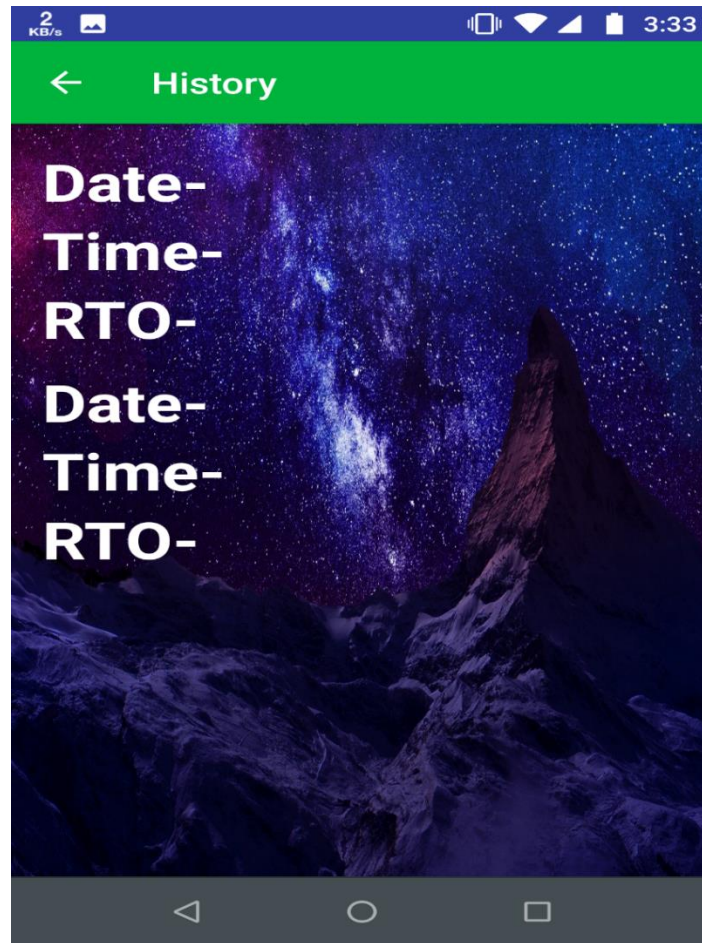


Figure-9.7.14

This is the history screen which is also a part of home screen menu options which usually provides the history of using this applications. This screens show some information about the how many time user booked some fire vehicles before means how many time user took advantages of this application in case of fire emergency.

This screen shows the information about the date, time and RTO means the date and time of day when user last time booked any fire vehicle and it is also provides the details about the vehicle with RTO number of the vehicle that was used last time.

This is the basically a nice feature that application provides to keep track of the user who already used this application before in some case of fire emergency. But due to lack of live server connectivity and backend also because of it is demo version application unable to view the vehicle and last time used information here.

13) Feedback Screen:

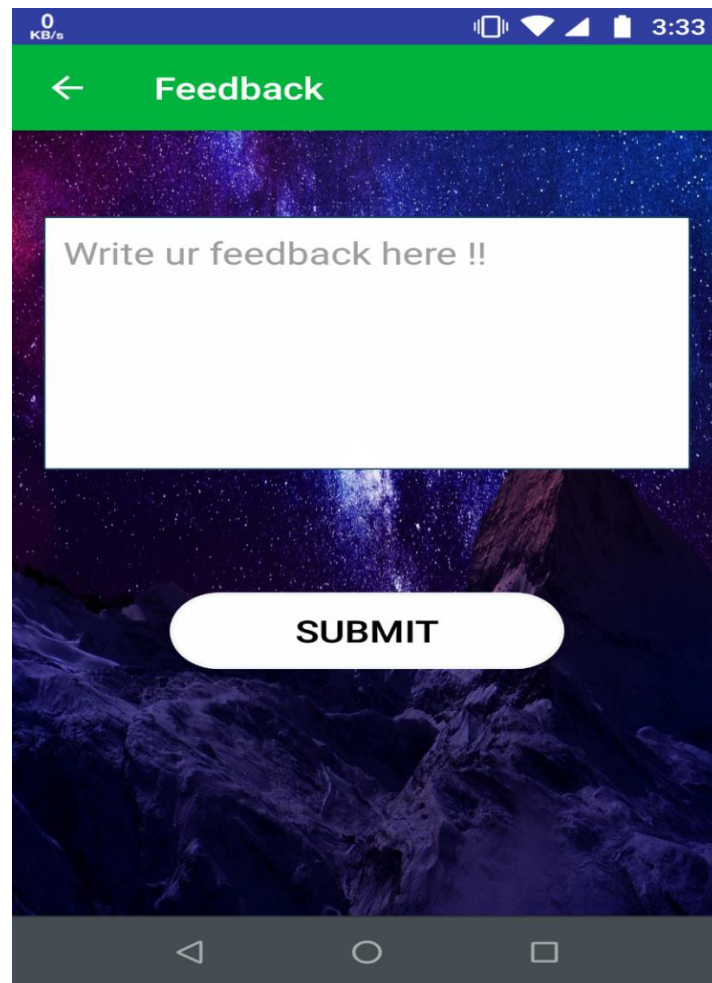


Figure-9.7.15

This is the last screen or functionality of the main home screen of the application. This is feedback screen which allows user to give some comments about the application or some special comments that user want to give about the application.

By feedback screen user can send some feedback to application which is available for support in the backend and they can easily see the feedback given by the user. In the feedback box user can type some sentences about some problem and some advice which will be useful for the better user experience for the application and it helps application to improve more without any drawbacks.

By submitting the feedback when user click the submit button feedback will be sent to admin and user will be navigated to the main home screen again.

14) Forgot password Screen:

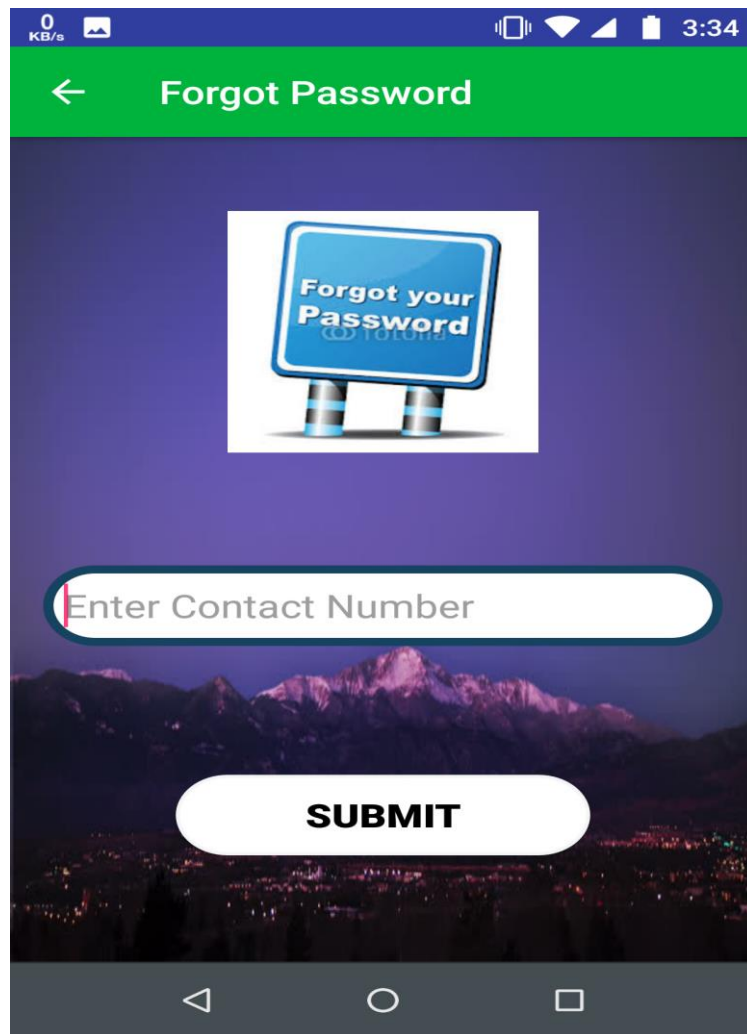


Figure-9.7.16

This is forgot password screen which is functionality provided by the main login screen when user open the application. User can see this functionality when user open the application and try to login the application this screen is used when user forget the password for the application.

When user click on forgot password button the application transfer user in to new forgot password screen which allow user to enter the mobile number again in the application. Once user enters the mobile number user will click the submit button so the application will send some new password for the user for the application. User can use that new password to login in the application and later user can change the password in the application after login. Application use very simple technique to change the password for the users in the application by sending them password on mobile numbers.

10. Testing of an Android Mobile Application:

Here in this part we have mention the testing part of the above example application. This thesis part shows how to do mobile application testing according to the thesis methodology. Here I will show you the sample of the testing document like test case, bug report and test summary report. This is just a sample to get idea about how testing task works in android mobile application.

Application requirements

Test subject: Fire Safety System

UI Description:

1. Application login page should contents two main field where first field is for to enter contact number and second field is for enter password. First field will allow only numbers and second will allow numbers and characters only. First field has input length till 9 digits only and second filed will allow length of password is the inputs only. There should be a “login” button to click and enter in application. There is should be option is given for recovery of password and for new user registration.

If we click on forgot password it should be navigate us to the another page called as forgot password and there should be a field to enter contact number and to submit the information there should be a “submit” button.

If we will click on “register now” it should be navigate us to the another page called as register now page. There should be a filed for enter contact number and button to generate OTP and which has several pages to finish the user registration process.

2. If we will enter valid contact number and password and click on log in button application should navigate us to the home page. The home page should has 6 options for user which are Profile, Fire, Help, About us, History and Feedback also home page has log out button.

Profile option should show users to their information like “Username”, “Contact no”, “Gender”, “Email”, “Address”. There should be option or pencil button beside that to edit “Username”, “Contact no”, “Gender”, “Email”, “Address” and one back arrow to go back on home page. This page will show user the information which user used while registration process.

Fire option will show user a Google map. It will show user their current locations by taking the lat long of user and shows them available fire vehicle nearby user’s location. Once user will book any fire vehicle by click on it they will get the notification and the fire vehicles will live to reach the user location and user can see the live navigation of fire vehicle to the user location which helps user to estimate the time of arrivals of fire vehicles for help.

Help option should contain the information about the responsible person of the application.

It should show information about the contact number and email address to contact if user have any query. Also a back arrow to go back to home page.

About us option should contain the necessary information about application and on top back arrow to navigate back on the home page.

History option should contain the information about previous help requests. It should contains the information about “date”, “time”, and “rto” which shows the information about date and time when user used this application for help last time with the vehicle information as rto number of the vehicle used for help. This page contains back arrow on the top to navigate on home page.

Feedback option should contain one field where user can write about experience after using the fire safety application as well as some suggestions and there should be a submit button to submit all the input information for application back office. As all option this should also have back arrow on the top to navigate on the home page.

Log out option should be on the top in right corner on home page. After using the application if user want to log out from their account user should click on that logout button and it will navigate you to the mail log in page.

Now here we have some sample example of the mobile application testing. This is not accurate and professional work. It is just for understand of the methodology and goal of the thesis.

Test case ID: 01

Test title: Valid inputs for the main log in page in filed 1 and field 2.

Pre-condition: application should installed and launched in android mobile phone

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|------------------------------|-----------|--|---------------|--------------------|-------|
| 1 | Enter date in field 1 | 1234125 | Input data should be displayed in field 1 | | | |
| 2 | Enter data in field 2 | ABGTY1234 | Input data should be display in field 2 | | | |
| 3 | Check the log in button | | Login button should be highlighted | | | |
| 4 | Click action on login button | | Page should be navigated from log in page to home page | | | |

Test case ID: 02

Test title: invalid inputs for the main log in page in filed 1 and field 2.

Pre-condition: application should installed and launched in android mobile phone

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|------------------------------|------------|---|---------------|--------------------|-------|
| 1 | Enter date in field 1 | 1365!@# | Input data should not display in field 1 | | | |
| 2 | Enter data in field 2 | ABGT!@#234 | Input data should not display in field 2 | | | |
| 3 | Check the log in button | | Login button should not be highlighted | | | |
| 4 | Click action on login button | | User will stay on the same main log in page | | | |

Test case ID: 03

Test title: Verification of “forgot password” button.

Pre-condition: application should installed and launched in android mobile phone

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|-----------------------------------|-----------|--|---------------|--------------------|-------|
| 1 | Enter date in field 1 | 1234#\$ | Input data should not display in field 1 | | | |
| 2 | Enter data in field 2 | ABGTY1234 | Input data should be display in field 2 | | | |
| 3 | Check the log in button | | Login button should not be highlighted | | | |
| 4 | Click action on “forgot password” | | Page should be navigated from log in page to forgot password page. | | | |

| | | | | | | |
|---|--|--|--|--|--|--|
| | | | | | | |
| | | | There should be field for enter password | | | |
| 5 | Enter the contact number in that field | | Contact number should display on field | | | |
| 6 | Check the submit button | | Submit button should be highlighted. | | | |
| 7 | Click action on “submit button” | | Contact number should be sent to back office | | | |

Test case ID: 04

Test title: verification of “profile” option.

Pre-condition: application should installed and launched in android mobile phone. Login process should be done and already navigated to the home page.

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|-------------------------------|-----------|--|---------------|--------------------|-------|
| 1 | Click action on the “Profile” | | Page should navigate to the viewprofile page | | | |
| 2 | Check the information shown. | | Username Contact no Gender Email Address This details should show on viewprofile page | | | |

Test case ID: 05

Test title: verification of “Fire” option.

Pre-condition: application should installed and launched in android mobile phone. Login process should be done and already navigated to the home page

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|----------------------------|-----------|---|---------------|--------------------|-------|
| 1 | Click action on the “fire” | | Page should navigate to the fire page. | | | |
| 2 | | | Google maps should be loaded on screen. | | | |

Test case ID: 06

Test title: verification of “Help” option.

Pre-condition: application should installed and launched in android mobile phone. Login process should be done and already navigated to the home page

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|----------------------------|-----------|---|---------------|--------------------|-------|
| 1 | Click action on the “Help” | | Page should navigate to the Help option | | | |
| 2 | Check the display | | There should be displayed contact number and emails address for the help about application. | | | |

Test case ID: 07

Test title: verification of “About us” option.

Pre-condition: application should installed and launched in android mobile phone. Login process should be done and already navigated to the home page.

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|---------------------------------|-----------|---|---------------|--------------------|-------|
| 1 | Click action on the “About us ” | | Page should navigate to the about us option | | | |
| 2 | Check the display screen | | There should be displayed all necessary information about the application | | | |

Test case ID: 08

Test title: verification of “history” option.

Pre-condition: application should installed and launched in android mobile phone. Login process should be done and already navigated to the home page.

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|-------------------------------|-----------|---|---------------|--------------------|-------|
| 1 | Click action on the “history” | | Page should navigate to the history option | | | |
| 2 | Check the display | | There should be displayed Date, time, RTO information about user last requests. | | | |

Test case ID: 09

Test title: verification of “feedback” option.

Pre-condition: application should installed and launched in android mobile phone. Login process should be done and already navigated to the home page

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|------------------------------------|-----------|--|---------------|--------------------|-------|
| 1 | Click action on the “feedback” | | Page should navigate to the feedback option | | | |
| 2 | | | There should be displayed text box field | | | |
| 3 | Enter the data into text box field | | Write feedback in text box field | | | |
| 4 | Check the submit button | | Submit button should be highlighted. | | | |
| 5 | Click action on submit button | | Feedback should be submitted to the back office. | | | |

Test case ID: 10

Test title: Verification of “logout button”

Pre-condition: application should installed and launched in android mobile phone

Dependencies: No dependencies.

Test case Designed by: Yogin Vaniya

Test case Designed date: 10.01.2020

Test case assigned to:

Test case assigned on: 10.01.2020

| step | Test steps | Test data | Expected Result | Actual Result | Status (Pass/Fail) | Notes |
|------|-----------------------------------|-----------|---|---------------|--------------------|-------|
| 1 | Enter date in field 1 | 1234125 | Input data should be displayed in field 1 | | | |
| 2 | Enter data in field 2 | ABGTY1234 | Input data should be displayed in field 2 | | | |
| 3 | Check the log in button | | Login button should be highlighted | | | |
| 4 | Click action on login button | | Page should navigate from log in page to home page | | | |
| 5 | | | Log out button should be displayed on right side on top. | | | |
| 6 | Click the action on logout button | | There screen should navigate from home page to main log in page | | | |

Sample Bug report for website “Fire Safety Application”

Bug Name: The login button is not highlighted

Bug ID: test case 01

Area Path:

Build Number: 01

Severity: high

Priority: high

Assigned to: Mariana

Reported By: Yogin Vaniya

Reported On: 01.01.2020

Reason: defect

Status: new

Environment: Any android mobile phone

Description: Unable to highlight the login button after valid input in filed one and two.

Steps To Reproduce:

- 1) Launch the application “fire safety ”
- 2) Enter valid data in field 1 and 2.
- 3) Check the login button is not highlighted

Expected result: button should be highlighted and user can login successfully to the next page.

Actual result: button is not highlighted after valid inputs.

11. Conclusion:

The goal of the thesis was to provide the information about android mobile application development with a practical example which makes easy for user to imagine the real view of an application. This thesis have covered almost every concepts that involved in the android mobile application development. In this thesis, there was brief information about the every essential components used in the application development which helps user to understand the application development from the basic level and helps user to easily start developing mobile application on android platform. The practical example application was working prototype and provide some basic functionality that any application needs and the application was trying to satisfy the idea of developing an application about the fire safety.

12. Bibliography:

Activities. (2015). Retrieved from Codelabs: <https://codelabs.developers.google.com/codelabs>

Android - Alert Dialog. (2004). Retrieved from tutorialspoint:
https://www.tutorialspoint.com/android/android_alert_dialoges.htm

Android - Image Switcher. (2004). Retrieved from tutorialspoint:
https://www.tutorialspoint.com/android/android_imageswitcher.htm

Android - Internal Storage. (2004). Retrieved from tutorialspoint:
https://www.tutorialspoint.com/android/android_internal_storage.htm

Android - Notifications. (2004). Retrieved from tutorialspoint:
https://www.tutorialspoint.com/android/android_notifications.htm

Android - Shared Preferences. (2004). Retrieved from tutorialspoint:
https://www.tutorialspoint.com/android/android_shared_preferences.htm

Android Custom ListView. (2013). Retrieved from Javatpoint:
<https://www.javatpoint.com/android-custom-listview>

Android ListView. (2013). Retrieved from Javatpoint: <https://www.javatpoint.com/android-listview-example>

Android overview. (2004). Retrieved from tutorialspoint:
https://www.tutorialspoint.com/android/android_overview.htm

Android SDK. (2005). Retrieved from techopedia:
<https://www.techopedia.com/definition/4220/android-sdk>

Android SearchView. (2013). Retrieved from Javatpoint:
<https://www.javatpoint.com/android-searchview>

Android software development. (2006). Retrieved from wikipedia:
https://en.wikipedia.org/wiki/Android_software_development

Android WebView . (2013). Retrieved from Javatpoint: <https://www.javatpoint.com/android-webview-example>

Androids-UI Controls. (2004). Retrieved from tutorialspoint:
https://www.tutorialspoint.com/android/android_user_interface_controls.htm

App Manifest Overview. (2005). Retrieved from developer.android.com:
<https://developer.android.com/guide/topics/manifest/manifest-intro>

Burnette, E. (2010). *Hello, Android: Introducing Google's Mobile Development Platform* by Ed Burnette. Dallas, Texas: Pragmatic Programmers, LLC.

Burton, M. (2015). *Android App Development for Dummies*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Codes learning. <https://www.udemy.com/>

[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

<https://www.tellforceblog.com/wp-content/uploads/2019/10/Android-evolution.jpg>

https://developer.android.com/guide/platform/images/android-stack_2x.png

https://www.tutorialspoint.com/android/images/google_maps.jpg

<https://static.javatpoint.com/images/androidimages/Android-Activity-Lifecycle.png>

https://materialdoc.cn/images/inbox_style_expanded-1.png

https://miro.medium.com/max/1052/1*iMxdeH1MPLrAWRmVLw-Pdg.png

https://www.tutlane.com/images/android/android_bigpicture_notification_example_result.png

Data and file storage overview. (2005). Retrieved from developer.android.com:
<https://developer.android.com/training/data-storage>

Goalkicker. (2017). *Android Notes For Professionals*. Goalkicker.com.

History of Android. (2013). Retrieved from javatpoint: <https://www.javatpoint.com/android-history-and-versions>

Java File. (2005). Retrieved from developer.android.com:
<https://developer.android.com/reference/java/io/File>

Layouts . (2005). Retrieved from developer.android.com:
<https://developer.android.com/guide/topics/ui/declaring-layout>

Types of Mobile Apps by Technology. (2017). Retrieved from CleverTap:
<https://clevertap.com/blog/types-of-mobile-apps/>

13. Appendix

| XML File | JAVA File | Description |
|----------------------|-------------------|--|
| Activity_check0 | Check_0.java | These files are used to give a landing page to the application. These files contain the source codes which helps application to start. |
| Activity_login | Login.java | These files allow user to enter credentials for login inside the application. |
| Activity_login2 | Login2.java | These files provides the way for user to create a new password if user forgot the password. |
| Activity_register | Register.java | These files provides facility to new users who wants to register in the application. |
| Activity_verify | Verify.java | These files used to verify the user who enter the received OPT by the application for the authentication purpose. |
| Activity_profile | Profile.java | These files allow users to create their profile for the application by entering the required personal information. |
| Activity_home | Home.java | This file is the main source files which has all the information about applications as it is the main home screen of the application where all the functionalities are placed. |
| Activity_viewprofile | Viewprofile.java | These files are created to display the details of the user who just registered in the applications and entered the details in the profile page. |
| Activity_editprofile | Editprofile.java | These files allow user to edit the details that user entered during the registration process. |
| Activity_maps | Mapsactivity.java | These files give a google map which shows the current location of the user in the map and available nearby |

| | | |
|-------------------|---------------|---|
| | | fire vehicles which user can book and see the live navigation to the user location. |
| Activity_help | Help.java | These files provides the information about the help like how user can get the any kind of help from the application. |
| Activity_about | About.java | These files provides the information about the application like what is this application and some further information about the application. |
| Activity_history | History.java | These files user to store the details about the use of application like it keep tracks of history about how many times user use this application to get help in any fire emergency. |
| Activity_feedback | Feedback.java | These files are used to store the data of user feedback which store the long text formatted feedback by user. |

| File name | XML code | JAVA code |
|-----------------|--|--|
| Activity_check0 | <pre><TextView android:layout_width="match_parent" android:layout_height="0dp" android:layout_weight="1"/> <ImageView android:layout_width="300dp" android:layout_height="300dp" android:layout_gravity="center" android:background="@drawable/fire1"/ > <TextView android:layout_width="match_parent"</pre> | <pre>@Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _check_0); Thread thread = new Thread() { @Override</pre> |

| | | |
|-----------------|--|---|
| | <pre> android:layout_height="0dp" android:layout_weight="1"/> </pre> | <pre> public void run() { super.run(); try { sleep(2000); } catch (InterruptedExce tion e) { e.printStackTrace(); } finally { Intent intent = new Intent(Check_0.this, Login.class); startActivity(intent); } } </pre> |
| Activity_login | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="20dp" android:orientation="vertical"> <Button android:layout_width="200dp" android:layout_height="wrap_content" android:textSize="25dp" android:text="Login" android:textStyle="bold" android:id="@+id/login" android:textColor="#000000" android:background="@drawable/edittex t5" android:layout_gravity="center" android:layout_marginTop="10dp" /> </LinearLayout> </LinearLayout> </pre> | <pre> Button login = (Button)findViewById(R.id.login) ; login.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent i1 = new Intent(Login.this,Home.class); startActivity(i1); } }); </pre> |
| Activity_login2 | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" </pre> | <pre> public class Login2 extends AppCompatActivity { EditText ed1; Button sbm; </pre> |

| | | |
|--|---|--|
| | <pre> android:layout_gravity="center" android:orientation="vertical"> <EditText android:layout_width="match_parent" android:layout_height="50dp" android:hint="Enter Contact Number" android:textSize="20dp" android:inputType="number" android:id="@+id/ed1" android:layout_gravity="center" android:layout_marginTop="70dp" android:background="@drawable/edittext2"/> </LinearLayout> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="50dp" android:orientation="vertical"> <Button android:layout_width="200dp" android:layout_height="50dp" android:text="Submit" android:textSize="20dp" android:layout_gravity="center" android:textAlignment="center" android:id="@+id/sbm" android:layout_marginTop="30dp" android:textStyle="bold" </pre> | <pre> @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _login2); ed1 = (EditText) findViewById(R.id.ed1); sbm = (Button) findViewById(R.id.sbm); sbm.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent i1 = new Intent(Login2.this,Login.class); startActivity(i1); } }); } } </pre> |
|--|---|--|

| | | |
|-------------------|--|--|
| | <pre> android:textColor="#000000" android:background="@drawable/edittext5"/> </LinearLayout> </pre> | |
| Activity_register | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="50dp" android:orientation="vertical"> <EditText android:layout_width="match_parent" android:layout_height="50dp" android:hint="Enter Contact Number" android:textSize="20dp" android:inputType="number" android:id="@+id/gcontact" android:layout_gravity="center" android:layout_marginTop="10dp" android:background="@drawable/edittext2"/> </LinearLayout> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="50dp" android:orientation="vertical"> <Button android:layout_width="200dp" android:layout_height="50dp" </pre> | <pre> public class Register extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _register); final EditText gcontact = (EditText)findViewById(R.id.gco ntact); Button motp = (Button)findViewById(R.id.otp); motp.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent i1 = new Intent(Register.this,Verify.class); startActivity(i1); } }); } } </pre> |

| | | |
|-----------------|--|---|
| | <pre> android:text="Generate OTP" android:textSize="20dp" android:textStyle="bold" android:textColor="#000000" android:layout_gravity="center" android:id="@+id/otp" android:layout_marginTop="30dp" android:background="@drawable/edittext5"/> </LinearLayout> </pre> | |
| Activity_verify | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:orientation="vertical"> <EditText android:layout_width="match_parent" android:layout_height="match_parent" android:hint="Enter OTP Number" android:textSize="20dp" android:id="@+id/otptext" android:inputType="number" android:background="@drawable/edittext2" android:layout_gravity="center" android:layout_marginTop="50dp"/> </LinearLayout> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" </pre> | <pre> public class Verify extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _verify); Button mresend = (Button)findViewById(R.id.resend); Button mconfirm = (Button)findViewById(R.id.confirm); final EditText otptext = (EditText)findViewById(R.id.otptext); mconfirm.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent i1 = new Intent(Verify.this,Profile.class); startActivity(i1); </pre> |

| | | |
|------------------|---|---|
| | <pre> android:layout_marginTop="50dp" android:orientation="horizontal"> <Button android:layout_width="0dp" android:layout_height="match_parent" android:background="@drawable/edittext5" android:text="resend" android:id="@+id/resend" android:textStyle="bold" android:textColor="#000000" android:textSize="20dp" android:layout_weight="1"/> <TextView android:layout_width="10dp" android:layout_height="match_parent" /> <Button android:layout_width="0dp" android:layout_height="match_parent" android:text="confirm" android:id="@+id/confirm" android:textStyle="bold" android:textColor="#000000" android:textSize="20dp" android:background="@drawable/edittext5" android:layout_weight="1"/> </LinearLayout> </pre> | <pre> } }); } } </pre> |
| Activity_profile | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:orientation="vertical"> </pre> | <pre> public class Profile extends AppCompatActivity { EditText ed1,ed2,ed3,ed4,ed5; Button sv; </pre> |

| | | |
|---------------|--|---|
| | <pre> <EditText android:layout_width="match_parent" android:layout_height="match_parent" android:layout_gravity="center" android:hint="Password" android:textSize="20dp" android:layout_marginTop="10dp" android:id="@+id/ed5" android:background="@drawable/edittext2"/> </LinearLayout> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:orientation="vertical"> <Button android:layout_width="match_parent" android:layout_height="match_parent" android:text="Save" android:layout_gravity="center" android:id="@+id/sv" android:layout_marginTop="10dp" android:background="@drawable/edittext5" android:textSize="20dp" android:textStyle="bold" android:textColor="#000000"/> </LinearLayout> </pre> | <pre> @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_profile); ed1=(EditText)findViewById(R.id.ed1); ed2=(EditText)findViewById(R.id.ed2); ed3=(EditText)findViewById(R.id.ed3); ed4=(EditText)findViewById(R.id.ed4); ed5=(EditText)findViewById(R.id.ed5); sv=(Button)findViewById(R.id.sv); sv.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent i1 = new Intent(Profile.this,Home.class); startActivity(i1); } }); } } </pre> |
| Activity_home | <pre> <LinearLayout android:layout_width="0dp" android:layout_height="match_parent" </pre> | <pre> @Override public boolean onOptionsItemSelected(MenuItem item) { </pre> |

| | | |
|--|--|--|
| | <pre> android:orientation="vertical" android:layout_weight="1"> <LinearLayout android:layout_width="match_parent" android:layout_height="0dp" android:orientation="vertical" android:layout_weight="1"> <Button android:layout_width="match_parent" android:layout_height="match_parent" android:id="@+id/fire" android:background="@drawable/images"/> </LinearLayout> <LinearLayout android:layout_width="match_parent" android:layout_height="0dp" android:background="#fff000" android:orientation="vertical" android:layout_weight="0.2"> <TextView android:layout_width="wrap_content" android:layout_gravity="center" android:layout_height="match_parent" android:text="Fire" android:textSize="22dp" android:textStyle="bold"/> </LinearLayout> </LinearLayout> </pre> | <pre> int id = item.getItemId(); //noinspection SimplifiableIfStatement if (id == R.id.action_settings) { AlertDialog.Builder builder = new AlertDialog.Builder(Home.this); builder.setCancelable(false) .setTitle("Logout") .setPositiveButton("Yes", new DialogInterface.OnClickListener() { @Override public void onClick(DialogInterface dialogInterface, int i) { Intent i1 = new Intent(Home.this,Login.class); startActivity(i1); } }) .setNegativeButton("No", new DialogInterface.OnClickListener() { @Override public void onClick(DialogInterface dialogInterface, int i) { dialogInterface.cancel(); } }); AlertDialog alert = builder.create(); alert.setMessage("Do you want to logout?"); alert.show(); } </pre> |
|--|--|--|

| | | |
|----------------------|---|--|
| | | <pre> return super.onOptionsItemSelected(ite m); } } </pre> |
| Activity_viewprofile | <pre> <LinearLayout android:layout_width="0dp" android:layout_height="match_parent" android:orientation="vertical" android:layout_weight="2"> <TextView android:layout_width="match_parent" android:layout_height="0dp" android:layout_weight="1"/> <TextView android:layout_width="wrap_content" android:layout_height="0dp" android:text="Username" android:textSize="20dp" android:id="@+id/tv1" android:textStyle="bold" android:textColor="#ffffff" android:layout_gravity="center" android:layout_weight="1"/> <TextView android:layout_width="match_parent" android:layout_height="0dp" android:layout_weight="1"/> </LinearLayout> </pre> | <pre> public class Viewprofile extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentview(R.layout.activity _viewprofile); Intent pr = getIntent(); ImageView p = (ImageView)findViewById(R.id. p); p.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent i1 = new Intent(Viewprofile.this,Editprofile .class); startActivity(i1); } }); } } </pre> |
| Activity_editprofile | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="0dp" </pre> | <pre> public class Editprofile extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { </pre> |

| | | |
|---------------|--|--|
| | <pre> android:orientation="vertical"> <ImageView android:layout_width="150dp" android:layout_height="150dp" android:id="@+id/im1" android:layout_gravity="center" android:background="@drawable/e"/> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:orientation="vertical"> <EditText android:layout_width="match_parent" android:layout_height="match_parent" android:hint="User name" android:layout_gravity="center" android:textSize="20dp" android:layout_marginTop="10dp" android:background="@drawable/edittext2" android:id="@+id/ed1"/> </LinearLayout> </pre> | <pre> super.onCreate(savedInstanceState); setContentView(R.layout.activity_editprofile); Intent i1 = getIntent(); Button sb =(Button)findViewById(R.id.sb); sb.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent i1 = new Intent(Editprofile.this,Home.class); startActivity(i1); } }); } } </pre> |
| Activity_maps | <pre> <fragment xmlns:android="http://schemas.android.com/apk/res/android" xmlns:map="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:id="@+id/map" </pre> | <pre> public class MapsActivity extends FragmentActivity implements OnMapReadyCallback { private GoogleMap mMap; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); </pre> |

| | | |
|---------------|---|--|
| | <pre> android:name="com.google.android.gms. maps. SupportMapFragment" android:layout_width="match_parent" android:layout_height="match_parent" tools:context="com.example.user.firesafe tysystem. MapsActivity" /> </pre> | <pre> setContentView(R.layout.activity _maps); SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager() .findFragmentById(R.id.map); mapFragment.getMapAsync(this); } @Override public void onMapReady(GoogleMap googleMap) { mMap = googleMap; LatLng prague = new LatLng(50.1006,14.3954); mMap.addMarker(new MarkerOptions().position(prague) .title("Marker in Prague")); mMap.moveCamera(CameraUpda teFactory.newLatLng(prague)); } } </pre> |
| Activity_help | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="30dp" android:orientation="vertical"> <TextView android:layout_width="match_parent" android:layout_height="70dp" android:text="Contact No: +420 774037670 " android:textSize="25dp" android:textColor="#ffffff" android:layout_gravity="center" </pre> | <pre> public class Help extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _help); Intent hp = getIntent(); } } </pre> |

| | | |
|----------------|---|---|
| | <pre> android:layout_marginTop="10dp"/> </LinearLayout> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="30dp" android:orientation="vertical"> <TextView android:layout_width="match_parent" android:layout_height="70dp" android:text="Email Us : yoginvaniya2506@gmail.com" android:textSize="25dp" android:textColor="#ffffff" android:layout_gravity="center" android:layout_marginTop="10dp"/> </LinearLayout> </pre> | |
| Activity_about | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="10dp" android:orientation="vertical"> <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Fire Locator System is an application that is use to get quick response in any emergency. Fire </pre> | <pre> public class About extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _about); Intent ab = getIntent(); } } </pre> |

| | | |
|------------------|---|---|
| | <p>locator system application is used to get any type of help in any trouble. You can book your fire vehicle for help by register once and also see live navigation of your booked fire vehicle. This application is based on real time application. You can submit your feedback after using this application."</p> <pre> android:textStyle="bold" android:textColor="#ffffff" android:textSize="25dp"/> </LinearLayout> </pre> | |
| Activity_history | <pre> <LinearLayout android:layout_width="wrap_content" android:layout_height="wrap_content" android:orientation="vertical" android:layout_marginTop="10dp"> <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Date-" android:textSize="35dp" android:textColor="#ffffff" android:textStyle="bold"/> <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Time-" android:textSize="35dp" android:textColor="#ffffff" android:textStyle="bold"/> <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="RTO-" android:textSize="35dp" </pre> | <pre> public class History extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _history); Intent hs = getIntent(); } } </pre> |

| | | |
|-------------------|---|--|
| | <pre> android:textColor="#ffffff" android:textStyle="bold" android:id="@+id/textView" /> </LinearLayout> </pre> | |
| Activity_feedback | <pre> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="40dp" android:orientation="vertical"> <EditText android:layout_width="match_parent" android:layout_height="wrap_content" android:id="@+id/ed1" android:hint="Write ur feedback here !!" android:inputType="textMultiLine" android:lines="8" android:minLines="6" android:gravity="top left" android:textSize="20dp" android:maxLines="10" android:scrollbars="vertical" android:background="@drawable/edittext3"/> </LinearLayout> <LinearLayout android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_marginTop="50dp" android:orientation="vertical"> <Button android:layout_width="200dp" android:layout_height="50dp" android:text="Submit" </pre> | <pre> public class Feedback extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity _feedback); Intent fb = getIntent(); } }. </pre> |

| | | |
|--|---|--|
| | <pre> android:textSize="20dp" android:id="@+id/btn" android:layout_gravity="center" android:layout_marginTop="30dp" android:textColor="#000000" android:background="@drawable/edittext5"/> </LinearLayout></pre> | |
|--|---|--|