**Chapter-1**

**Introduction**

To respond to the rising public transport needs and mitigate escalating vehicular traffic in the twin-cities of [Hyderabad](https://en.wikipedia.org/wiki/Hyderabad) and [Secunderabad](https://en.wikipedia.org/wiki/Secunderabad), the erstwhile [Telangana Government](https://en.wikipedia.org/wiki/Andhra_Pradesh_Government) and the [South Central Railway zone](https://en.wikipedia.org/wiki/South_Central_Railway) jointly launched the [Multi Modal Transport System](https://en.wikipedia.org/wiki/Multi-Modal_Transport_System_(Hyderabad)) (MMTS) in August 2003. This Rs. 167 crores worth transport system, 50 percent of which was financed by the State Government and the remaining by the Central Government, used the already existing rail lines of the [Indian Railways](https://en.wikipedia.org/wiki/Indian_Railways) in and around Hyderabad making it a highly cost effective one. However, within months of starting of its operation, it became apparent that it has been met with poor response. The reasons for this dismal response include poor punctuality of trains, lack of dedicated tracks for MMTS services, and absence of feeder bus services limiting the network’s reach.

Though the ridership of the MMTS has increased over the years, the failure in augmentation of the services in line with the increasing ridership has led to its unpopularity among 150,000 passengers who avail the services daily. This notwithstanding, the [Union Ministry of Urban Development](https://en.wikipedia.org/wiki/Ministry_of_Urban_Development), in October 2003, gave its nod for the Hyderabad Metro Rail ,this project was seen necessary in view that the population of Hyderabad was forecasted to reach 13.6 million by the year 2021. According to the initial plan, the metro was to be connected with the already existing MMTS to provide commuters with alternate modes of transport. Simultaneously, the proposals for taking up the construction of [MMTS Phase-II](https://en.wikipedia.org/wiki/Multi-Modal_Transport_System_(Hyderabad)#Phase_II) were also taken forward. Hyderabad's metro project is ambitious in scope and swift in execution.

**1.1 Present System**

The existing system has some ways of booking tickets for a MMTS. one is to book tickets at the ticket counter of respective railway station .Former is one of the hectic processes where one should stand in long queues for hours This system is basically concerned with the reservation, but not cancellation of railway tickets to the passengers,

* Passenger cannot be refunded.
* Standing in long queues for hours.
* Sometimes train must be late so that passenger looses his journey.
* Due to negligence of passenger ticket can be miss.

**1.2 What we are proposing?**

The proposed system is an Android application where one can book tickets with just one click go. A smart phone user can book tickets at any time of day or night. He will be guided with all the necessary steps to book tickets and scan ticket at the entrance of metro station by the app.

Also in the proposed system, customers can cancel seats at a suitable time (20min before). If the customer wishes to cancel his tickets he will be given a confirmation details regarding his cancellations. As the customer buys tickets online through his/her R-wallet , on cancellation of tickets the refund amount (20% of the amount will be charged for service charges) will be added back to his/her R-wallet account. To enhance the refund function, all the customers have to register and become a member before buying tickets so that he faces no problem while accessing the Application.

* It reserves and cancels seats for the passenger.
* It contains information about the trains.
* It contains information about the passenger.
* It contains the details of reservation fees, any concessions etc.
* It makes entries for reservation, waiting, cancelled tickets.
* It will update for uptime and downtime trains
* To avoid long queues QR-code is used.
* Money will redeemed from R-wallet.

**Chapter-2**

**Literature Survey**

A Literature review is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews are secondary sources and do not report new or original experimental work.

2.1 **Manual Booking and E-Booking**

It is analyzed from the people that the current booking systems i.e, The existing system has some ways of booking tickets for a MMTS. One is to book tickets at the ticket counter of respective railway station. Former is one of the hectic processes where one should stand in long queues for hours This system is basically concerned with the reservation, but not cancellation of railway tickets to the passengers. And also we have another type which is E-booking, this type of process consider only through computer based systems , again tickets are collect from counter.

**2.2 M-Booking**

In order to avoid these drawbacks we have introduced M-booking which is Android based application that a customer can book a ticket at anywhere and anytime.He will be guided with all the necessary steps to book tickets and scan ticket at the entrance of metro station by the app.

Also in the proposed system, customers can cancel seats at a suitable time (20min before). If the customer wishes to cancel his tickets he will be given a confirmation details regarding his cancellations. As the customer buys tickets online through his/her R-wallet , on cancellation of tickets the refund amount (20% of the amount will be charged for service charges) will be added back to his/her R-wallet account. To enhance the refund function, all the customers have to register and become a member before buying tickets so that he faces no problem while accessing the Application.

**Chapter-3**

**SYSTEM ANALYSIS**

To provide flexibility to the users, the interfaces have been developed that are accessible through smartphones. The GUI’s at the top level have been categorized as.

1. Administrative user interface

2. The operational or generic user interface

The ‘administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

The ‘operational or generic user interface’ helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities

**3.1 Functional Activities**

**Modules:**

* **Usage of application** With advent of smart phones the passengers can easily book a ticket anywhere anytime just one click go , so that time will be reduces and passenger arrive within a time.
* **Login** user need to login with unique Id and Password.
* **Ticket** when passenger book a ticket, QR-code will be generated on the user’s application. And that QR-code must be scanned at entrance of metro station.
* **R-Wallet** once ticket is generated money will be redeemed from user’s R-wallet. If cancellation acts again money will refunded to R-wallet. Money can added through net banking as well as visa.
* **User friendly** The system is easy to learn and understand. A native user can also use the system effectively , without any difficulties.
* **Database** To obtain statistic information from the booking record.

**3.2 Non-Functional Activities:**

**Response Time:**

The response of all the operation is good. This has been made possible by careful programming.

**Portable:**

The application is used in any android platforms, even android version enhanced.

**Security:**

The system use user id and password in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity. If user forget his password a OTP sent to his/her register mobile number. The system’s back-end servers shall only be accessible to authenticated management.

**Reliability:**

All the information updated time to time, whenever needs to book a ticket all the updates shown on the application. Also the system will be functioning inside a container. Thus the overall stability of the system depends on the stability of container and its underlying operating system.

**Availability:**

The system should be available at all times, meaning the user can access it using a mobile app, only restricted by the down time of the server on which the system runs. A customer friendly system which is in access of people around the world should work 24 hours. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the Organizer. Then the service will be restarted. It means 24 x 7 availability.

**Maintainability:**

A application database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the project will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.

**Supportability:**

The code supports all android application and supporting modules of the system will be well documented and easy to understand.

# 3.3 System Requirement Specification:-

**Software Requirements:**

* Operating System : Windows 8.1/10 64/32 bits
* Technology : Android Studio 2.0
* Database : SQL lite

**Hardware Requirements:**

* Hardware : Intel core i3
* Speed : 2.1 Ghz
* RAM : 4GB
* External Device : Key Board,Mouse,Monitor/Laptop

**Chapter-4**

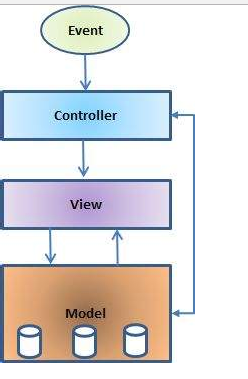
**Design**

**4.1 MVC**

Basic MVC ARCHITECTURE advertisements. Model View Controller or MVC as it popularly called, is a software design pattern for developing web applications. A model view controller pattern is made up of the following three parts:

* Model – The lowest level of the patter which is responsible for maintaining data.
* View- This is responsible for displaying all or a portion of the user.
* Controller – Software code that controls the interactions between the model and view.

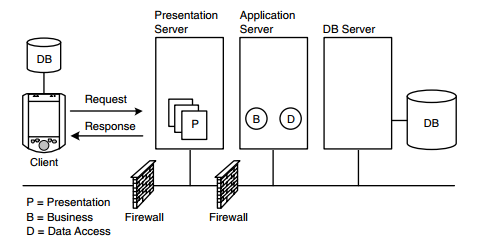
MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns . Here the controller receives all request from the application then works with the model to prepare any data needed by the view. The view then uses the data prepared by the controller to generate the final presentable response. The MVC abstraction can be graphically rpresented.



**Fig 4.1 MVC**

**4.2 3 Tier- Architecture**

we discuss mobile application architectures. We start by describing some of the general concepts and terms behind client-server architectures and follow this by describing clients and servers and the connectivity between them. We then present several interesting architectural patterns and describe why they are useful as general mobile application architecture solutions. Finally, we discuss some of the tenets behind good architectural design and the considerations you need to be aware of when designing mobile applications.



**Fig 4.2 3 Tier Architecture**

**4.3 UML**

**Unified Modeling Language**

**Overview:**

Unified Modeling Language, UML for short, is the international standard notation for Object-Oriented Analysis and Design (OOAD). It is a standardized specialization language that can be used for Object Modeling. It has been defined by the Object Management Group (OMG) and has already become the de-facto standard for designing Object-Oriented Software Applications. In the field of software engineering, the UML is a standardized specification language for object modeling. UML is a general-purpose modeling language that includes a graphical notation used to create an abstract model of a system, referred to as a *UML Model*.

**Introduction to UML:**

According to the OMG specification, "The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. UML offers a standard way to write a system's blueprints, including conceptual things, such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components."

The UML is purely process independent, i.e., it is not tied up with a SDLC (Software development life cycle) process. The basic purpose behind UML modeling is visualizing, constructing, specifying and documenting a system. It should be noted that UML is a standard; it is not a methodology, process or a procedure. Rather, we use UML as a standard that uses some predefined standard notations with a view to modeling and defining a software system, to document it and define the artifacts involved.

UML is not restricted to modeling software. UML is also used forbusiness process modeling, systems engineering modeling and representing organizational structures. UML has been a catalyst for the evolution of model-driven technologies, which include Model Driven Development (MDD), Model Driven Engineering (MDE), and Model Driven Architecture (MDA).

**Goals of UML:**

The UML was invented primarily to address the challenges faced in the design and architecture of complex systems. The basic objectives or goals behind UML modeling are:

* Define an easy to use and visual modeling language for modeling a system's structure
* Provide extensibility
* Be language and platform independent so that it can be used for modeling a system irrespective of the language and platform in which the system is designed and implemented
* Incorporate the best possible practices at par with the industry standards
* Provide support for Object Orientation, design and apply frameworks and patterns

**4.4 UML Diagrams:**

In UML there are various models that define the notation and semantics for a wide variety of domains. The UML models depict how the classes and objects in a system interact with one another. UML diagrams are used to provide a graphical representation of the system being modeled. UML 2.0 defines thirteen diagrams that are broadly classified into three categories with each category containing one or more diagrams that fall under that category. These categories are:

**4.4.1 The Structural Diagrams**

The Structural Diagrams relate to the static structure of a system, i.e., they represent elements that are static in nature. These diagrams are fundamental to the UML modeling of a system and portray the static structure of the system as a whole. The Structural Diagrams are comprised of the following:

It includes the following diagrams:

* The Class diagram
* The Component diagram
* The Composite Structure diagram
* The Deployment diagram
* The Object diagram
* The Package diagram
  + 1. **The Behavioral Diagrams**

The Behavioral Diagrams model how the system functions. These comprise of:

* Use Case Diagram
* Activity Diagram
* State Machine Diagram

**4.4.3 The Interaction Diagrams**

These diagrams represent how flow of data and control takes place in the system that is being modeled. They are a subset of the Behavioral Diagrams. These include:

* Sequence Diagram
* Collaboration

**Usecase Diagram**

A Use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system.

### SEQUENCE DIAGRAM:

A Sequence diagram is a virtual representation of a scenario. A sequence diagram shows the various actors in the scenario, and the way they interact with all the subsystems. A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.

### STATE CHART DIAGRAM:

A state chart diagram shows how an object dynamically changes its lifetime. A State is a condition or situation in which the object satisfies some condition, does some Task, or waits for an event to trigger. A state chart diagram address the dynamic view of the system.

### Activity Diagram

Activity diagrams represent workflows in an graphical way. They can be used to describe business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams. Check out this wiki article to learn about symbols and usage of activity diagrams.

### CLASS DIAGRAM:

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Class diagram address the static design view of a system. Class diagrams that include Active classes address the static process view of a system.

A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements on or more interfaces.

### Component Diagram

A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that has many components. Components communicate with each other using [interfaces](http://en.wikipedia.org/wiki/Interface_(object-oriented_programming)). The interfaces are linked using connectors

### Deployment Diagram

A deployment diagrams shows the hardware of your system and the software in those hardware. Deployment diagrams are useful when your software solution configuration is deployed across multiple machines with each having a unique.

.



**Fig 4.3 Usecase diagram**



**Fig 4.4 Sequence Diagram for Booking Ticket**



**Fig 4.5 sequence Diagram for Ticket cancellation**



**Fig 4.6 Class Diagram**



**Fig 4.7 State Diagram for Booking Ticket**



**Fig 4.8 State Diagram for Ticket Cancellation**

**4.5 Data Dictionary**

**Table 4.1 Passenger:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Passenger** | **Type** | **Primary key** | **Foreign key** | **Null value** | **Unique key** |
| Name | Char |  |  |  |  |
| User id | Varchar |  | 1 | Not null | Yes |
| Password | Varchar |  |  | Not null |  |
| Adhaar no | Number |  |  |  |  |
| Contact no | Number |  |  |  |  |
| Age | Number |  |  |  |  |
| Gender | Char |  |  |  |  |
| Address | Varchar |  |  |  |  |

**Table 4.2 Login:**

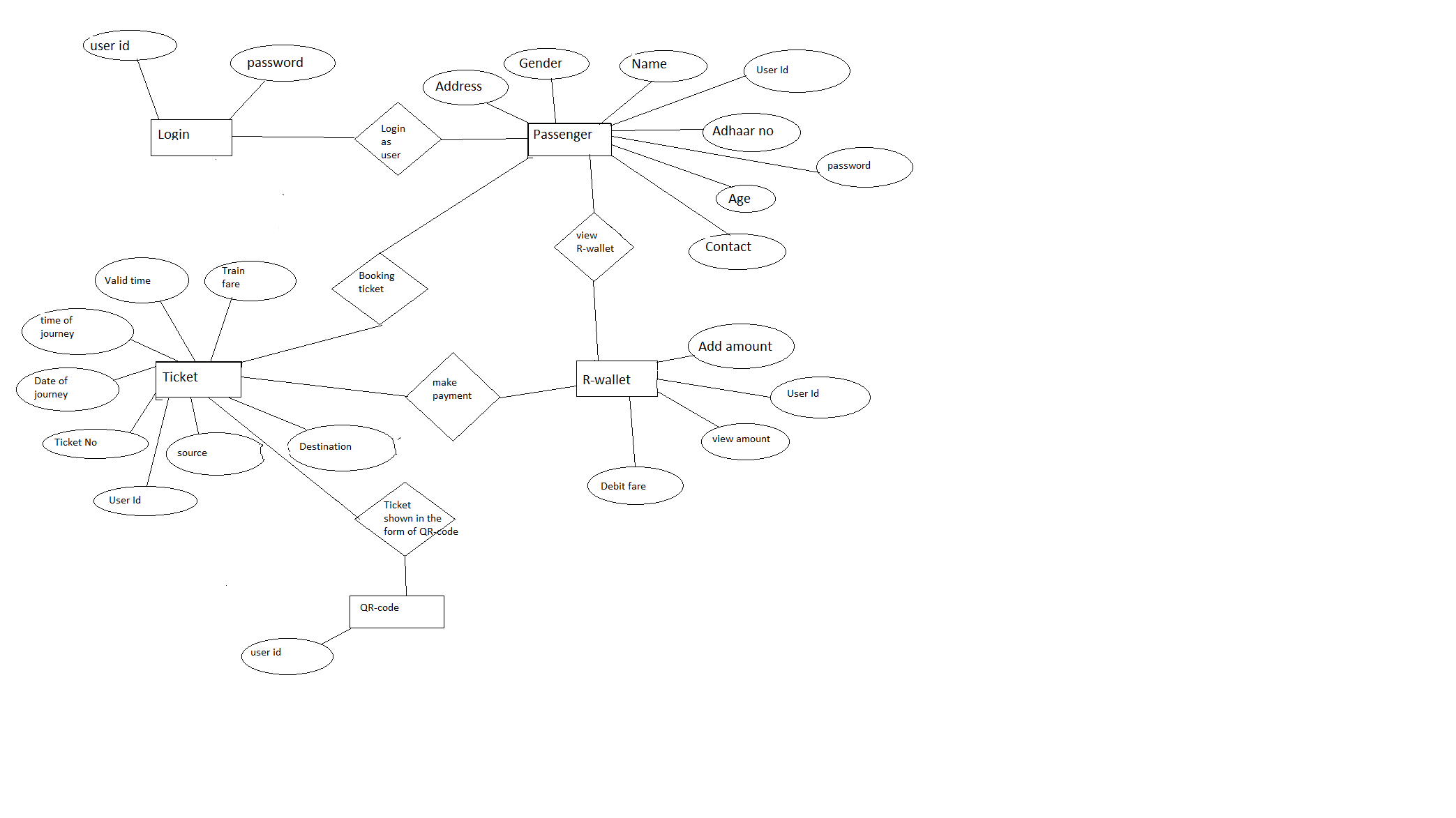
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Login** | **Type** | **Primary key** | **Foreign key** | **Null value** | **Unique key** |
| User id | Varchar | 1 |  | Not null | Yes |
| Password | Varchar |  |  | Not null |  |

**Table 4.3 Ticket**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ticket** | **Type** | **Primary key** | **Foreign key** | **Not null** | **Unique key** |
| Ticket no | Number |  |  |  |  |
| Train fair | Char |  |  |  |  |
| Valid time | Time |  |  |  |  |
| User id | Varchar | 1 |  | Not null | yes |
| Date of journey | Date |  |  |  |  |
| Time of journey | Time |  |  |  |  |
| Source | Char |  |  |  |  |
| Destination | Char |  |  |  |  |

**Table 4.4 R-wallet:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **R-wallet** | **Type** | **Primary key** | **Foreign key** | **Null value** | **Unique key** |
| View amount | Number |  |  |  |  |
| Add amount | Number |  |  |  |  |
| User id | Varchar | 1 |  | Not null | Yes |
| Debit fair | Number |  |  |  |  |



**Fig 4.9 Entity Relationship Diagram**

**Chapter-5**

**Implementation**

**5.1 Installation of Android Studio**

### Download Android Studio

Google provides Android Studio for the Windows, Mac OS X, and Linux platforms. You can [download this software](https://developer.android.com/studio/index.html) from the Android Studio homepage. (You'll also find the traditional SDKs, with Android Studio's command-line tools, available from the Downloads page.) Before downloading Android Studio, make sure your platform meets one of the following requirements:

**Windows OS**

* Microsoft Windows 7/8/10 (32-bit or 64-bit)
* 2 GB RAM minimum, 8 GB RAM recommended
* 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
* 1280 x 800 minimum screen resolution
* JDK 8
* For accelerated emulator: 64-bit operating system and Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality

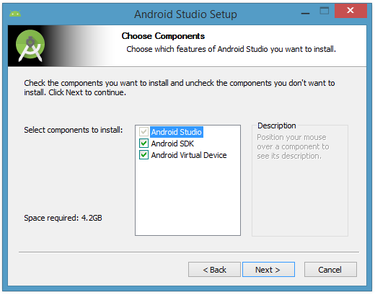
### Installing Android Studio on 64-bit Windows 8.1

I launched android-studio-bundle-143.2821654-windows.exe to start the installation process. The installer responded by presenting the Android Studio Setup dialog box s



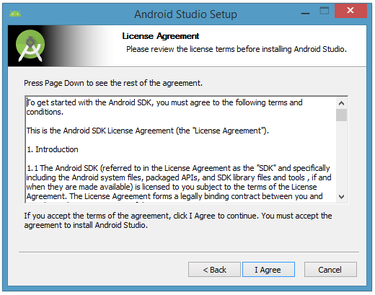
**Fig5.1 Set up Android Studio**

Clicking Next took me to the following dialog box, which gives you the option to decline installing the Android SDK (included with the installer) and an Android Virtual Device (AVD).



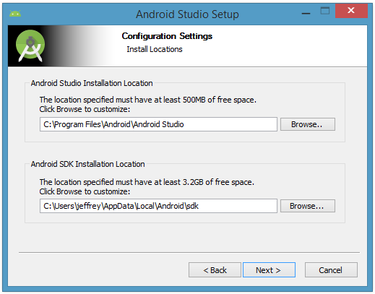
**Fig5.2 Do you want to install the Android SDK and AVD?**

I chose to keep the default settings. After clicking Next, you'll be taken to the license agreement dialog box. Accept the license to continue the installation.



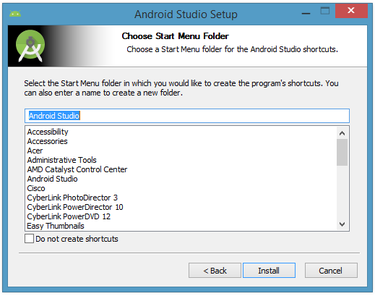
**Fig5.3 Accept the license agreement to continue installation**

The next dialog box invites you to change the installation locations for Android Studio and the Android SDK.



**Fig5.4 Android setup**

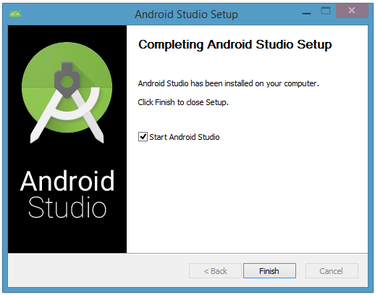
Change the location or accept the default locations and click Next.The installer defaults to creating a shortcut for launching this program, or you can choose to decline. I recommend that you create the shortcut, then click the Install button to begin installation.



**Figure5.5 Create a new shortcut for Android Studio**

The resulting dialog box shows the progress of installing Android Studio and the Android SDK. Clicking the Show Details button will let you view detailed information about the installation progress.

The dialog box will inform you when installation has finished. When you click Next, you should see the following:



**Figure 5.6 Leave the Start Android Studio check box checked to run this software**

To complete your installation, leave the Start Android Studio box checked and click Finish.

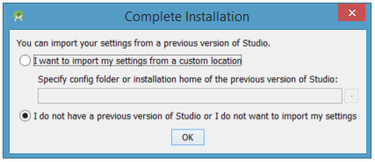
## Running Android Studio

Android Studio presents a splash screen when it starts running:



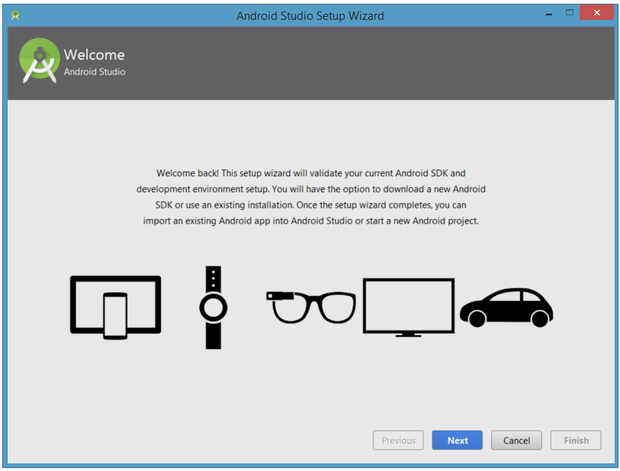
**Fig5.7 Android Studio's start screen**

On your first run, you'll be asked to respond to several configuration-oriented dialog boxes. The first dialog box focuses on importing settings from any previously installed version of Android Studio.



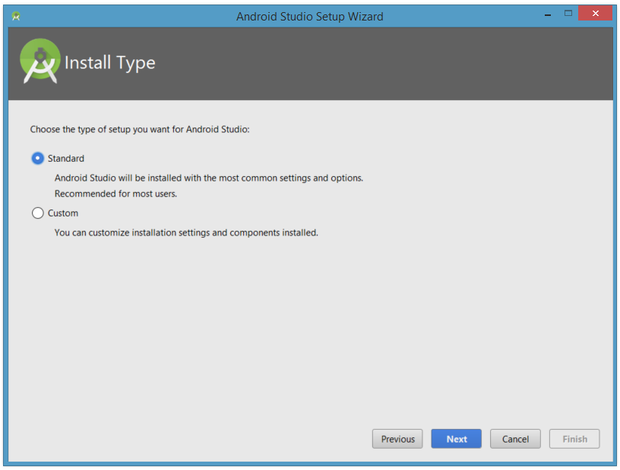
**Fig5.8 Import settings**

If you're like me, and don't have a previously installed version, you can just keep the default setting and click OK. Android Studio will respond with a slightly enhanced version of the splash screen, followed by the Android Studio Setup Wizard dialog box:

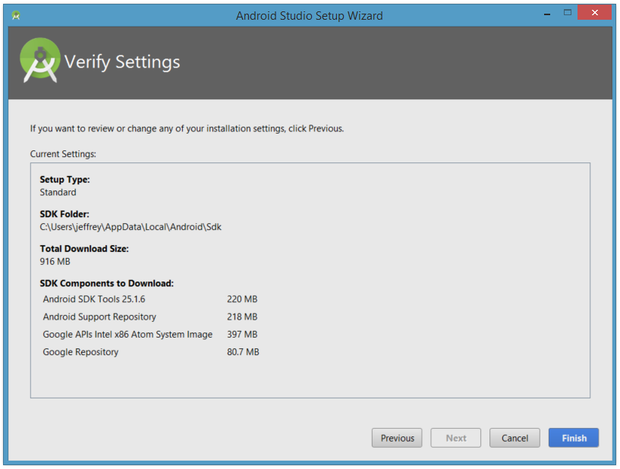


**Fig5.9 Validate your Android SDK and development environment setup**

When you click Next, the setup wizard invites you to select an installation type for your SDK components. For now I recommend you keep the default standard setting.

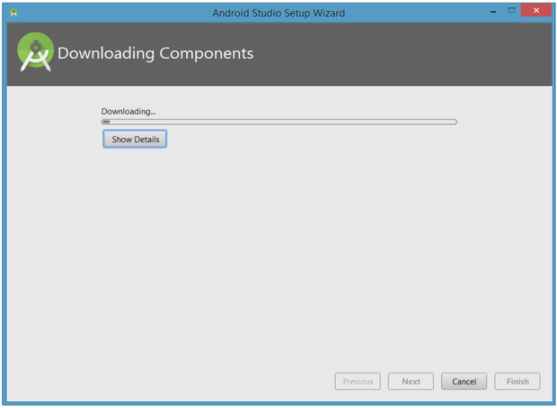


**Fig5.10 Choose an installation type**

Click Next and verify your settings, then click Finish to continue. 

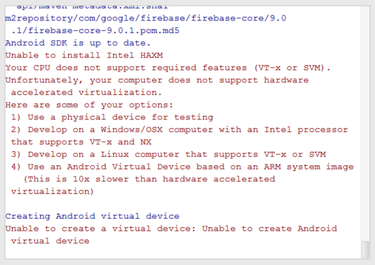
**Fig5.11 Review settings**

The wizard will download and unzip various components. Click Show Details if you want to see more information about the archives being downloaded and their contents.



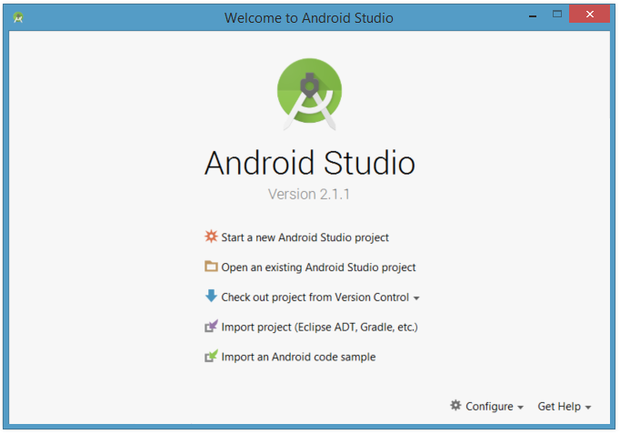
**Fig5.12 The wizard downloads and unzips Android Studio components**

If your computer isn't Intel based, you might get an unpleasant surprise after the components have completely downloaded and unzipped:



**Fig5.13 Intel-based hardware acceleration is unavailable**

Your options are to either put up with the slow emulator or use an Android device to speed up development. I'll discuss the latter option later in the tutorial. Finally, click Finish to complete the wizard. You should see the Welcome to Android Studio dialog box:



**Fig5.14 Welcome to Android Studio**

You'll use this dialog to start up a new Android Studio project, work with an existing project, and more. You can access it anytime by double-clicking the Android Studio shortcut on your desktop.

**Outputs and screeshots**

**1.activity-home-page.xml**

Source code about home page layout ,once user open the application directly shown home page where user need to register with certain specifications.

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.example.fashkl.project.HomePage">

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/back3"

android:scaleType="fitXY" />

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="250dp"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_centerVertical="true"

android:layout\_marginLeft="5dp"

android:layout\_marginRight="5dp"

android:background="#99FFFFFF">

<Button

android:id="@+id/credits\_Button"

android:layout\_width="110dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_marginRight="50dp"

android:layout\_marginTop="110dp"

android:background="@drawable/mybtn3"

android:text="register"

android:textColor="#FFFFFF"

android:textSize="20dp" />

<Button

android:id="@+id/login\_Button"

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_marginLeft="50dp"

android:layout\_marginTop="110dp"

android:background="@drawable/mybtn3"

android:text="Login"

android:textColor="#FFFFFF"

android:textSize="20dp" />

<TextView

android:id="@+id/detailHeader"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="#6F839A"

android:gravity="center\_horizontal"

android:padding="10dp"

android:text="Local Train Ticketing System"

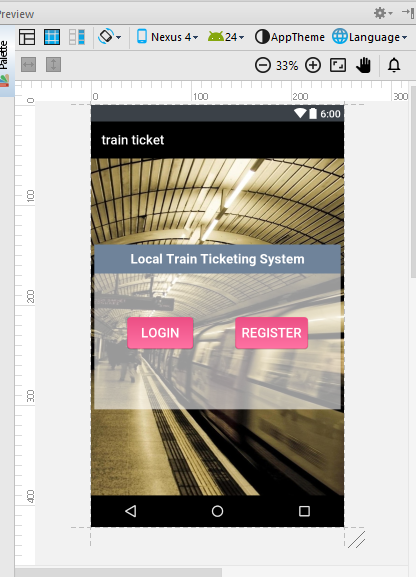
android:textColor="#FFFFFF"

android:textSize="20dp"

android:textStyle="bold" />

</RelativeLayout>

</RelativeLayout>



**Fig 5.15 Homepage**

**2.activity\_register\_page.xml**

Once user open the application to book a ticket user need to register with certain information.

<?xml version=”1.0” encoding=”utf-8”?>

<RelativeLayout xmlns:android=”http://schemas.android.com/apk/res/android”

xmlns:tools=”http://schemas.android.com/tools”

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”

tools:context=”com.example.fashkl.project.SelectPage”>

<ImageView

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”

android:background=”@drawable/ferriswheel” />

<ScrollView

android:id=”@+id/scrollView”

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”

android:layout\_marginTop=”10dp”>

<TableLayout

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”

android:background=”#9333”

android:orientation=”vertical”>

<TableRow android:gravity=”center\_horizontal”>

<TextView

android:id=”@+id/textView”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_alignParentTop=”true”

android:layout\_centerHorizontal=”true”

android:layout\_gravity=”center\_horizontal”

android:layout\_marginTop=”10dp”

android:text=”Registeration”

android:textAppearance=”?android:attr/textAppearanceLarge”

android:textColor=”@color/whiteColor”

android:textSize=”30dp”

android:textStyle=”bold” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_marginBottom=”10dp”

android:layout\_marginLeft=”10dp”

android:layout\_marginRight=”10dp”

android:layout\_marginTop=”10dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView2”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”User Id”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/RuserIdFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:inputType=”number”

android:background=”@android:color/white”

android:ems=”10”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”10dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView3”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Name”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/RnameFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:padding=”5dp” />

</TableRow><TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”10dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView8”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Password”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/RpasswordFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:inputType=”textPassword”

android:background=”@android:color/white”

android:ems=”10”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”10dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView4”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Address”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/RaddressFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”10dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView5”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Age”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/RageFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:inputType=”number”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”10dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView6”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Mobile No”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/RmobileNoFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:inputType=”phone”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”10dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView7”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Card No”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/RcardNoFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:inputType=”number”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”10dp”

android:gravity=”center\_horizontal”

android:orientation=”horizontal”>

<Button

android:id=”@+id/RsubmitBtn”

android:layout\_height=”30dp”

android:background=”@drawable/mybtn3”

android:text=”Register”

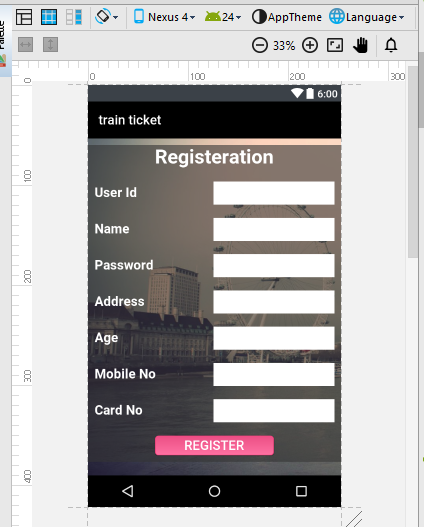
android:textColor=”@color/whiteColor”

android:textSize=”20sp” />

</TableRow>

</TableLayout></ScrollView>

</RelativeLayout>

****

**Fig5.16 register page**

**3.activity\_login\_page.xml**

Once registration completed user need to login with unique username and password.

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/back2"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".LoginPage">

<ProgressBar

android:id="@+id/login\_progress"

style="?android:attr/progressBarStyleLarge"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="8dp"

android:visibility="gone" />

<ScrollView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="@drawable/shape"

android:layout\_centerHorizontal="true">

<RelativeLayout

android:layout\_width="330dp"

android:layout\_height="280dp"

android:layout\_centerHorizontal="true"

>

<AutoCompleteTextView

android:id="@+id/userIdFld"

android:layout\_width="200dp"

android:layout\_height="35dp"

android:inputType="number"

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true"

android:layout\_marginLeft="130dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="80dp"

android:background="@color/whiteColor"

android:ems="10"

android:hint="user id"

android:padding="5dp" />

<TextView

android:id="@+id/UserIDtextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_above="@+id/passwordTxtView"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_marginLeft="10dp"

android:layout\_marginStart="10dp"

android:labelFor="@id/userIdFld"

android:text="User Id :-"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<TextView

android:id="@+id/passwordTxtView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/UserIDtextView"

android:layout\_alignStart="@+id/UserIDtextView"

android:layout\_below="@+id/userIdFld"

android:layout\_marginTop="52dp"

android:labelFor="@id/userIdFld"

android:text="Password :-"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<EditText

android:id="@+id/passwordFld"

android:layout\_width="200dp"

android:layout\_height="35dp"

android:layout\_alignBottom="@+id/passwordTxtView"

android:layout\_alignEnd="@+id/userIdFld"

android:layout\_alignLeft="@+id/userIdFld"

android:layout\_alignRight="@+id/userIdFld"

android:layout\_alignStart="@+id/userIdFld"

android:background="@android:color/white"

android:ems="10"

android:hint="password"

android:inputType="textPassword"

android:padding="5dp" />

<Button

android:id="@+id/loginRegistBtn"

android:layout\_width="wrap\_content"

android:layout\_height="35dp"

android:textSize="15dp"

android:layout\_alignLeft="@+id/passwordFld"

android:layout\_alignStart="@+id/passwordFld"

android:layout\_below="@+id/passwordFld"

android:layout\_marginTop="20dp"

android:background="@drawable/mybtn3"

android:focusable="true"

android:text="Login"

android:textColor="@color/whiteColor" />

<TextView

android:id="@+id/logLrgTxt"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_toEndOf="@+id/passwordTxtView"

android:layout\_toRightOf="@+id/passwordTxtView"

android:text="Login"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textColor="@android:color/white"

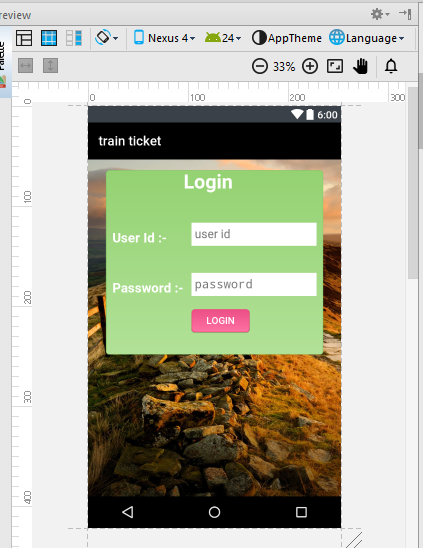
android:textSize="30dp"

android:textStyle="bold" />

</RelativeLayout>

</ScrollView>

</RelativeLayout>



**Fig 5.17 login page**

**4.activity\_book\_ticket\_page.xml**

after login the app user gets the available trains so he must choose source and destination and fare details will be store in database.

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.example.fashkl.project.SelectPage">

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/ando3" />

<ScrollView

android:id="@+id/scrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="6dp">

<TableLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#9333"

android:orientation="vertical">

<TableRow

android:layout\_margin="10dp"

android:gravity="center\_horizontal">

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="10dp"

android:gravity="center\_horizontal"

android:text="Book Ticket"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textColor="@color/whiteColor"

android:textSize="30dp"

android:textStyle="bold" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Line Type"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<Spinner

android:id="@+id/BKlineTypeSpnr"

android:layout\_width="0dp"

android:layout\_height="35dp"

android:background="@android:color/white"

android:entries="@array/lineType"

android:spinnerMode="dialog" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="From Station"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<Spinner

android:id="@+id/BKfromStationSpnr"

android:layout\_width="0dp"

android:layout\_height="35dp"

android:background="@color/whiteColor"

android:entries="@array/Stations"

android:spinnerMode="dialog" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="To Station "

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<Spinner

android:id="@+id/BKtostationSpnr"

android:layout\_width="0dp"

android:layout\_height="35dp"

android:background="@android:color/white"

android:entries="@array/Stations"

android:padding="5dp"

android:spinnerMode="dialog" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView8"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Cost"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<EditText

android:id="@+id/BkCostFld"

android:layout\_width="0dp"

android:layout\_height="35dp"

android:background="@android:color/white"

android:focusable="false"

android:inputType="numberDecimal"

android:padding="5dp"

android:paddingLeft="5dp" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView5"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Journy Type"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<RadioGroup

android:id="@+id/BkjournyTypeRB"

android:orientation="horizontal">

<RadioButton

android:id="@+id/BksingleRBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:backgroundTint="@color/whiteColor"

android:text="single"

android:textColor="@color/whiteColor"

android:textStyle="bold" />

<RadioButton

android:id="@+id/BkreturnRBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:backgroundTint="@color/whiteColor"

android:text="Return"

android:textColor="@color/whiteColor"

android:textStyle="bold" />

</RadioGroup>

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView6"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Toatal"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<EditText

android:id="@+id/BktotalFld"

android:layout\_width="0dp"

android:layout\_height="35dp"

android:background="@android:color/white"

android:focusable="false"

android:inputType="numberDecimal"

android:padding="5dp"

android:paddingLeft="5dp" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView7"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Balance"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<EditText

android:id="@+id/BkbalabnceFld"

android:layout\_width="0dp"

android:layout\_height="35dp"

android:background="@android:color/white"

android:ems="10"

android:focusable="false"

android:inputType="numberDecimal"

android:padding="5dp" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:gravity="center\_horizontal"

android:orientation="horizontal">

<Button

android:id="@+id/BKBookBtn"

android:layout\_height="30dp"

android:background="@drawable/mybtn3"

android:text="Book"

android:textColor="@color/whiteColor"

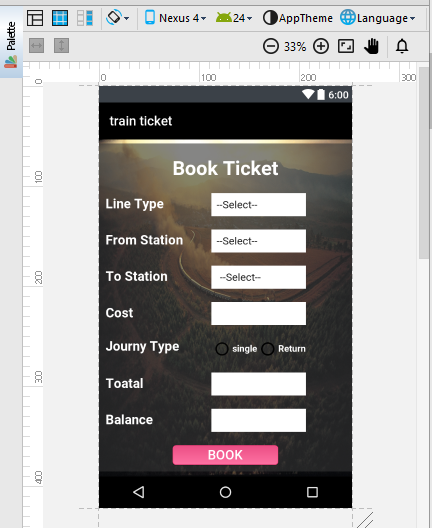
android:textSize="20sp" />

</TableRow>

</TableLayout>

</ScrollView>

</RelativeLayout>



**Fig 5.18 Book Tickets**

**5.activity\_ticket\_details\_page.xml**

Once booking Tickets confirm passenger can verify the details in this page.

<?xml version=”1.0” encoding=”utf-8”?>

<RelativeLayout xmlns:android=”http://schemas.android.com/apk/res/android”

xmlns:tools=”http://schemas.android.com/tools”

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”

tools:context=”com.example.fashkl.project.SelectPage”>

<ImageView

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”

android:background=”@drawable/back3” />

<ScrollView

android:id=”@+id/scrollView”

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”>

<TableLayout

android:layout\_width=”match\_parent”

android:layout\_height=”match\_parent”

android:background=”#9333”

android:layout\_marginTop=”35dp”

android:orientation=”vertical”>

<TableRow

android:layout\_marginTop=”15dp”

android:gravity=”center\_horizontal”>

<TextView

android:layout\_span=”2”

android:id=”@+id/textView”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_alignParentTop=”true”

android:layout\_centerHorizontal=”true”

android:layout\_gravity=”center\_horizontal”

android:layout\_marginTop=”10dp”

android:text=”Ticket Details”

android:textAppearance=”?android:attr/textAppearanceLarge”

android:textColor=”@color/whiteColor”

android:textSize=”30dp”

android:textStyle=”bold” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_marginTop=”10dp”

android:orientation=”horizontal”

android:layout\_marginLeft=”15dp”

android:layout\_marginRight=”15dp”

android:layout\_marginBottom=”15dp”>

<TextView

android:id=”@+id/textView2”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:paddingRight=”15dp”

android:text=”Ticket ID”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/TKuserIdFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:padding=”5dp”

android:editable=”false” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”15dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView3”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”To”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/TktoFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:editable=”false”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”15dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView4”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”From”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/TkformFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:editable=”false”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”15dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView5”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Type”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/TktypeFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:inputType=”numberDecimal”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:layout\_margin=”15dp”

android:orientation=”horizontal”>

<TextView

android:id=”@+id/textView6”

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:text=”Date”

android:textColor=”@color/whiteColor”

android:textSize=”20dp”

android:textStyle=”bold” />

<EditText

android:id=”@+id/TkdateFld”

android:layout\_width=”200dp”

android:layout\_height=”35dp”

android:background=”@android:color/white”

android:ems=”10”

android:focusable=”false”

android:inputType=”phone”

android:padding=”5dp” />

</TableRow>

<TableRow

android:layout\_width=”wrap\_content”

android:layout\_height=”wrap\_content”

android:layout\_gravity=”center\_horizontal”

android:gravity=”center\_horizontal”

android:layout\_margin=”15dp”

android:orientation=”horizontal”>

<Button

android:id=”@+id/TkclearBtn”

android:layout\_width=”110dp”

android:layout\_height=”35dp”

android:background=”@drawable/mybtn3”

android:text=”Clear”

android:textColor=”@color/whiteColor”

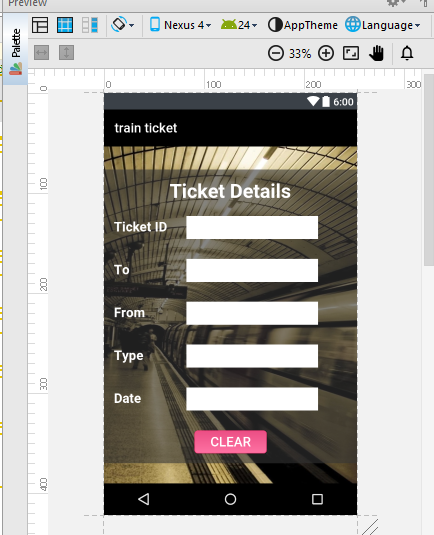
android:textSize=”20sp” />

</TableRow>

</TableLayout>

</ScrollView>

</RelativeLayout>



**Fig 5.19 Ticket details**

**6.activity\_add\_balance\_page.xml**

After ticket booking money will be redeemed from R-wallet and once money fund to booking passenger can add fare amount from online banking or visa card.

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.example.fashkl.project.SelectPage">

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/newyork" />

<ScrollView

android:id="@+id/scrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TableLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="30dp"

android:background="#9333"

android:orientation="vertical">

<TableRow

android:layout\_margin="15dp"

android:gravity="center\_horizontal">

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="10dp"

android:text="Add Balance"

android:textAppearance="?android:attr/textAppearanceLarge"

android:textColor="@color/whiteColor"

android:textSize="30dp"

android:textStyle="bold" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginBottom="15dp"

android:layout\_marginLeft="15dp"

android:layout\_marginRight="15dp"

android:layout\_marginTop="10dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="User Id"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<EditText

android:id="@+id/BuserIdFld"

android:layout\_width="200dp"

android:inputType="number"

android:layout\_height="35dp"

android:background="@android:color/white"

android:ems="10"

android:padding="5dp" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="15dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Balance"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<EditText

android:id="@+id/BbalanceFld"

android:layout\_width="200dp"

android:layout\_height="35dp"

android:focusable="false"

android:background="@android:color/white"

android:ems="10"

android:inputType="numberDecimal"

android:padding="5dp" />

</TableRow>

<!--<TableRow-->

<!--android:layout\_width="wrap\_content"-->

<!--android:layout\_height="wrap\_content"-->

<!--android:layout\_gravity="center\_horizontal"-->

<!--android:layout\_margin="15dp"-->

<!--android:orientation="horizontal">-->

<!--<TextView-->

<!--android:id="@+id/textView4"-->

<!--android:layout\_width="wrap\_content"-->

<!--android:layout\_height="wrap\_content"-->

<!--android:text="Card No"-->

<!--android:textColor="@color/whiteColor"-->

<!--android:textSize="20dp"-->

<!--android:textStyle="bold" />-->

<!--<EditText-->

<!--android:id="@+id/BcardNoFld"-->

<!--android:layout\_width="200dp"-->

<!--android:layout\_height="35dp"-->

<!--android:background="@android:color/white"-->

<!--android:ems="10"-->

<!--android:inputType="number"-->

<!--android:padding="5dp" />-->

<!--</TableRow>-->

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="15dp"

android:orientation="horizontal">

<TextView

android:id="@+id/textView6"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Amount"

android:textColor="@color/whiteColor"

android:textSize="20dp"

android:textStyle="bold" />

<EditText

android:id="@+id/BamountFld"

android:layout\_width="200dp"

android:layout\_height="35dp"

android:background="@android:color/white"

android:ems="10"

android:inputType="numberDecimal"

android:padding="5dp" />

</TableRow>

<TableRow

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_margin="15dp"

android:gravity="center\_horizontal"

android:orientation="horizontal">

<Button

android:id="@+id/BsubmitBtn"

android:layout\_height="30dp"

android:background="@drawable/mybtn3"

android:text="Submit"

android:textColor="@color/whiteColor"

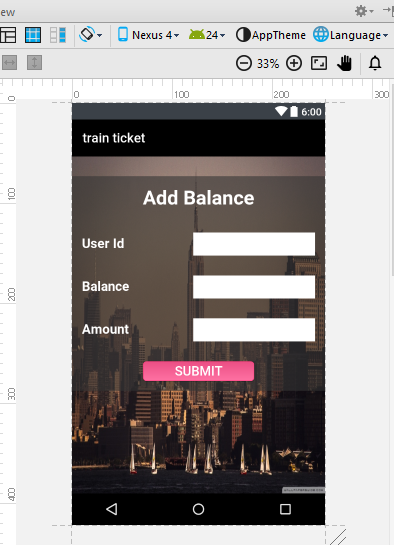
android:textSize="20sp" />

</TableRow>

</TableLayout>

</ScrollView>

</RelativeLayout>

****

**Fig5.20 Add balance**

**7.activity\_view\_user\_page.xml**

user details can be shown on the app he/she also modify the details.

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:background="@drawable/ando2"

tools:context="com.example.fashkl.project.TicketDetailsPage">

<ScrollView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<RelativeLayout

android:layout\_width="350dp"

android:layout\_height="300dp"

android:background="#9333"

android:layout\_centerVertical="true"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge"

android:text="Search Ticket"

android:textSize="30dp"

android:textColor="@color/whiteColor"

android:id="@+id/searchTxtv"

android:layout\_marginTop="30dp"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<AutoCompleteTextView

android:layout\_width="200dp"

android:layout\_height="40dp"

android:id="@+id/autoCompleteTextView"

android:padding="5dp"

android:hint="ticket id"

android:background="@color/whiteColor"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="130dp"

android:inputType="number" />

<Button

android:id="@+id/searchBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="search"

android:textColor="@color/whiteColor"

android:background="@drawable/mybtn3"

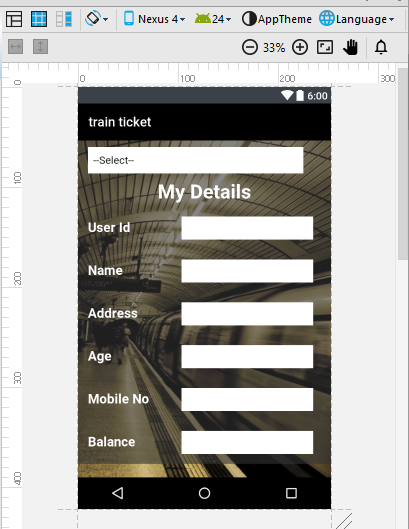
android:textStyle="bold"

android:layout\_marginTop="200dp"

android:layout\_centerHorizontal="true"/>

</RelativeLayout>

</ScrollView></RelativeLayout>



**Fig 5.21 user details**

**Chapter-6**

**TESTING**

**Software testing** is a process of executing a program or application with the intent of finding the **software** bugs. It can also be stated as the process of validating and verifying that a **software** program or application or product: Meets the business and technical requirements that guided it's design and development.

**6.1 BLACK-BOX TESTING**

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

## 6.2 White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called **glass testing** or **open-box testing**. In order to perform **white-box** testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

## 6.3 Unit Testing

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

## 6.4 Integration Testing

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration .

In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations.

**6.5 System Testing**

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards. This type of testing is performed by a specialized testing team.

System testing is important because of the following reasons:

* System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.
* The application is tested thoroughly to verify that it meets the functional and technical specifications.
* The application is tested in an environment that is very close to the production environment where the application will be deployed.
* System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

**6.6 TestCases**

|  |
| --- |
| Test case#: Test case name:Booking Ticket  Designed by: Sub system:  System: Ticket Booking app Designed date:  Execution by: Execution Date:  Short Description: |

|  |
| --- |
| Pre-conditions:  The user has download App- He has to register with certain specifications.  Login the App with username & password  The System display main menu  View train Details |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sno.** | **Action** | **Expected system Response** | **Pass/fail** | **comment** |
| 1 | Login | System shows the Homepage |  |  |
| 2 | Ticket Booking |  |  |  |
| 3 | R-wallet | Money will be redeemed by users R-wallet |  |  |
| 4 | QR-code | Ticket Generated |  |  |

|  |
| --- |
| Post conditions:  Ticket will be saved in DATA BASE. |

Test happy with

Test Also failure condition

**Chapter-7**

**Conclusion**

This application will gather the information regarding the travelling options between stations along with their timing and fares. This application will combine number of functionalities into one.

**Chapter-8**

**Bibiliography**

Citiation standards in this reference are provided for:

Books

References

Books

**Books**

[1] Belen cruz Zapata . Android Application Development. October 2013.

[2] Adam gerber and cligton craig . Learn Android Studio. 14 sep 2014.

[3] Neil smyth . Android studio essential . 14 sep 2014.

[4] Puthembekar. Web technology. 12 june 2016.

**Reports**

[1] Clifton craig, adam gerber, learning studio, 14 sep 2014.

[2] Mike wolfson, 2013

[3]paul deitel, Harvey deitel, abbey deitel, 13 otc 2014.

**Online Sources:**

[1] <https://developer.android.com/index.html>

[2] [https://www.google.co.in/webhp?sourceid=chrome- instant&ion=1&espv=2&ie=UTF-8#q=android%20studio%20tutorial](https://www.google.co.in/webhp?sourceid=chrome-%20%20%20%20%20%20%20%20%20instant&ion=1&espv=2&ie=UTF-8#q=android%20studio%20tutorial)

[3] <https://www.tutorialspoint.com/android/android_studio.htm>

[4] http://www.androidauthority.com/android-studio-tutorial-beginners-637572/ .