THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY

VELLORE-02

MINI PROJECT REPORT

ONLINE TICKET BOOKING ANDROID APPLICATION PROJECT

Submitted by

SANTHOSH P

SIVA KA

VASANTH V

In partial fulfilment for the award of degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY, CHENNAI – 600 025.

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY

VELLORE-02

BONAFIDE CERTIFICATE

This is to certify that the mini project report entitled **ONLINE TICKET BOOKING ANDROID APPLICATION** Project is a bonafide record of the project work done by **SANTHOSH P (513119104031), SIVA KA (513119104035), VASANTH V (513119104044),** during the academic year 2020-2021 towards the partial fulfilment of the requirement of the award of B-E Degree in **COMPUTER SCIENCE AND ENGINEERING** of Anna University, Chennai -600025.

Guided by

Prof N.JAGADEESWARI, M.E.,
Assistant professor,
Department of CSE,
TPGIT,Vellore.

Dr.RAHILA BILAL M.E.,Ph.D.,
Head of the department/CSE,
TPGIT,Vellore.

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

Project work is a practical application of our knowledge that we gather during the course. To make a project work successful, eminent person's involvement is very much required.

We thank our beloved principal **Dr. M. ARULARASU, M.E., Ph.D.,** who always served as a source of inspiration and encouragement.

We express our grateful thanks to **Dr. RAHILA BILAL, M.E., Ph.D.,** who has contributed his valuable suggestions in doing this project.

We have no words to express thank to our project supervisor, **Prof. N.JAGADEESWARI, M.E,** who gave fruitful ideas and constantly guided us to finish this project.

We extend our thanks to all the Teaching and Non-Teaching Staff of the Department of Mechanical Engineering and others sole help and cooperation in successful completion of this project.

SANTHOSH P	**(513119104031)**
SIVA KA	**(513119104035)**
VASANTH V	**(513119104044)**

TABLE OF CONTENTS

ABSTRACT

The Online ticket reservation system is used to reserve or book the ticket in online by the passengers. The passengers don’t have to go directly to the station to reserve or book the ticket. They don’t have to wait in the queue. They can book the ticket in online with their mobile phone or laptop from anywhere and at anytime. There are various applications for booking the tickets individually. But in this system the users (passengers) are provide to book their tickets from the one platform. In this platform they can book their tickets for Cab, Bus, Train and Flight. The tickets are directly booked to the available seats in system at the particular time the passengers prefer for. In this system we provide them with the user friendly and comfortable using system. The main objective of this project is to give a useful and easy access to all the tickets booking for the Cab, Bus, Train and Flight at the system (Application) for the users.

CHAPTER 1

INTRODUCTION

This project is an android application for the online ticket booking or reservation for cab, bus, train and flight, reserve seats, cancellation of reservation and different types of route enquiries used on securing quick reservations. This is the application where you can access all those four modes of transportation ticket from your place. The staff at the ticket counter is using an internal system to sell tickets at the counter and customers ask for information and this brings a lot of inconveniences to customers while they are at an important work. The type of system being used at the counter is an internal system which is manually used in selling the tickets. The problems facing the company are that customers have to go to the counter to buy ticket or ask for schedule, customers will also have to queue up for a long time in order to secure a ticket and will also need to pay cash when they buy the ticket. In this situation our application is very help for the passengers to book or reserve ticket.

Badariah, (2007) emphasized that the Online Transport Booking System which was developed at Politeknik Kota Kuala Terengganu (PKKT) was to make sure that users could make their online booking or reservations to their desired transport companies with facilities provided by the new system. He pointed out that the methodology and technology being used in this new transport system could be applied to other areas of activities. The user who wants to use the transport must make an application to book the transport before boarding.

Similarly, after considering the type of system which Badariah adopted, this project will be designed with the same aim of presenting the customers of Imo Transport Company with the opportunity of making reservations at the comfort of their homes or offices without being faced with the challenges of queuing at counters before embarking on any journey. This project will also enlighten prospective customers and users of the system on the need to patronize the system as it displays more advantages over the old system by providing an easy to use Graphic User interface (GUI) interaction

CHAPTER 2

SYSTEM ANALYSIS

2.1 REQUIREMENT ANALYSIS

This phase is done for understanding what all are the improvements needed by the user for overcoming the drawbacks of the current system. The problem could be automating an existing manual process, developing a new automated system, or combination of these two.

The emphasis in the requirement analysis is to identify what are the users expect from the system, not how the system will achieve those requirements. If the client and developers don’t understand the limitations of the system and failed to know the actual aim for developing the new one, i.e., if requirement analysis is not properly done, it will lead to an inefficient system. So, before starting design, we should analyse the system and collect data from users, which are useful to our project. It must be able to get the answers of the following questions.

- Who will use our system?
- What are they expect from our app?
- What are their basic needs?

2.2 REQUIREMENT SPECIFICATION

The process of establishing the services, the system should provide and the constraints under which it should operate is called requirement analysis. System requirements should set out what the system must do rather than how it is done. A requirement definition is a statement in natural language plus illustration, which defines the constraints under which the proposed system must operate. The document is also called functional specification. It serves as a contract between the system user and software developer.

Firstly a requirement definition is written and then it is expanded to requirement specification. The software design is based directly on the requirement specification documents must specify all functional and performance requirements.

2.3 FEASIBILITY STUDY

Feasibility study is a procedure that identifies describes and evaluates candidate system and selects the best system for the job. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be cost effective from a business point of view and if it can develop given existing budgetary constraints. The key considerations involved in the feasibility analysis are economic, technical, behavioural and operational.

2.3.1 ECONOMIC FEASIBILITY

The economic analysis is to determine the benefits and savings that are expected from the candidate system and compare them with cost. The system is economically feasible, as the organization possesses the hardware and software resources required for the functioning of the system. Any additional resources, if required, can also be easily acquired.

2.3.2 TECHNICAL FEASIBILITY

It centres on the existing computer system and to what extent it can support the proposed addition. Since the minimum requirements of the system like IIS of the server and a browser on the client, are met by any average user.

2.3.3 OPERATIONAL FEASIBILITY

The system operation is the longest phase in the development lifecycle of a system. So, operational feasibility should be given much importance. The users of the system don’t need through training on the system. All they are expected to know to operate the system is the basic netsurfing knowledge. It has user friendly interface.

2.3.4 BEHAVIOURAL FEASIBILITY

In today’s world, where computer is an inevitable entity, the system like auction site, which requires no special efforts than surfing the net are enjoying wide acceptance. Thus the organization is convinced that the system is feasible.

2.4 EXISTING SYSTEM

The already existing system has the feature of the tickets that can be book or reserve on the journey station itself and there is a system where the passengers can book their tickets online from their place but this system has the module for individual transport the passengers need to install individually for a particular system.

2.5 PROPOSED SYSTEM

The proposed system has the new features of booking the tickets all different kind of modules at the same place at the single module. This application has the modules for all the system in it.

CHAPTER 3

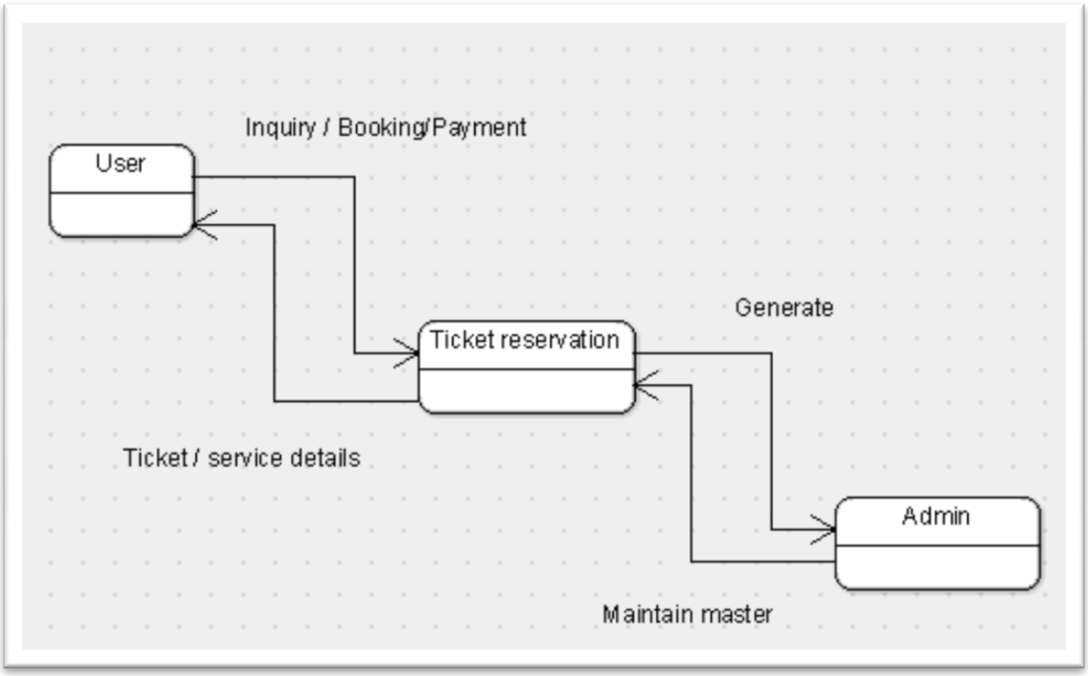
SYSTEM DESIGN

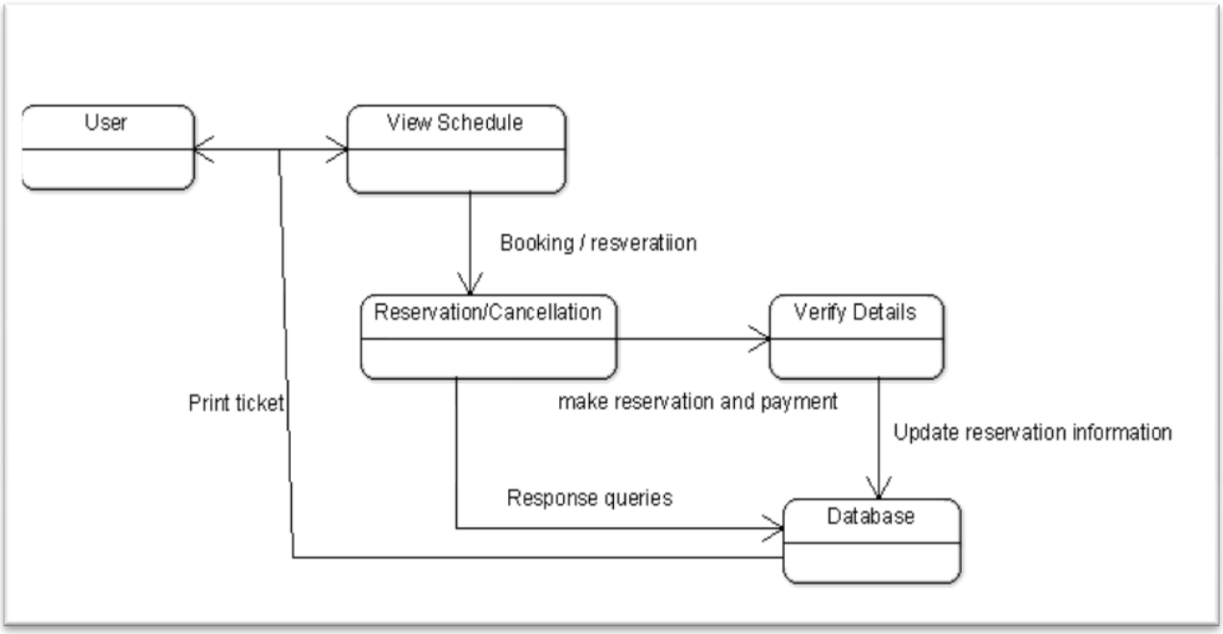
The most creative and challenging of the system life cycle is system design. The term design describes a final system and the process by which it is developed. The design phase focuses on the detailed implementation of the system recommended in the feasibility study.

DATA FLOW DIAGRAM

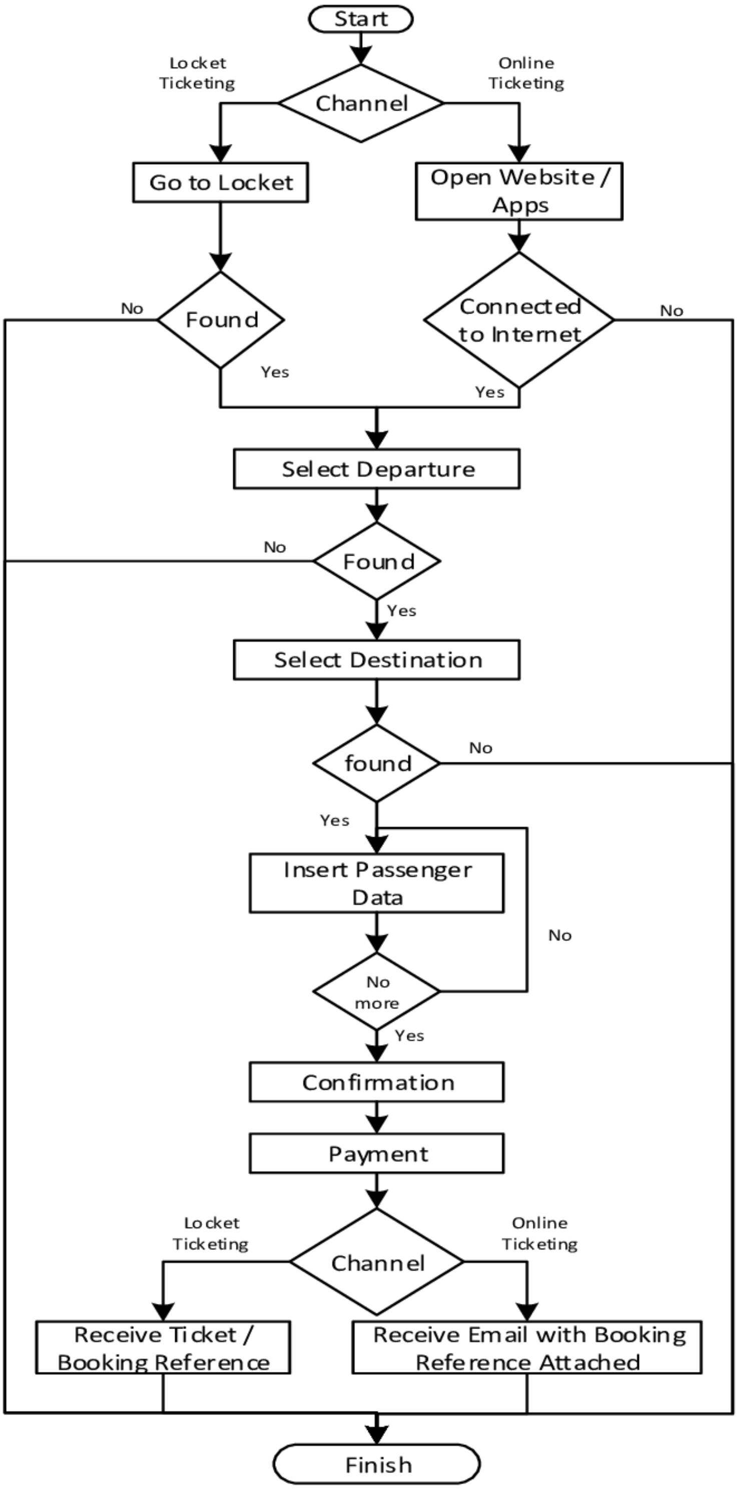
The DFD is a network representation of the system. They are excellent mechanism for communicating with customers during requirement analysis. A DFD, also known as bubble chart, which clarify system requirements identifying major transformations. It is the starting point in the system design and decomposes the requirement specification down to the lowest level.

A DFD represents data flow between individual statement and blocks of statement in a routine, data flow between sequential routines, data flow between concurrent processes or a distributed computing system where each node represents a geographically remote processing unit. DFD are quite valuable for establishing naming conventions and names of systems, files, and data links. It describes what flow rather than how they are proposed, so it doesn’t depend on hardware, software and data structures or file organizations.


Inquiry / Booking/Payment
User
Generate
Ticket reservation
Ticket / service details
Admin
Maintain master


User
View Schedule
Booking / resveratiion
Reservation/Cancellation
Verify Details
Print ticket
make reservation and payment
Update reservation information
Response queries
Database

FLOW CHART:

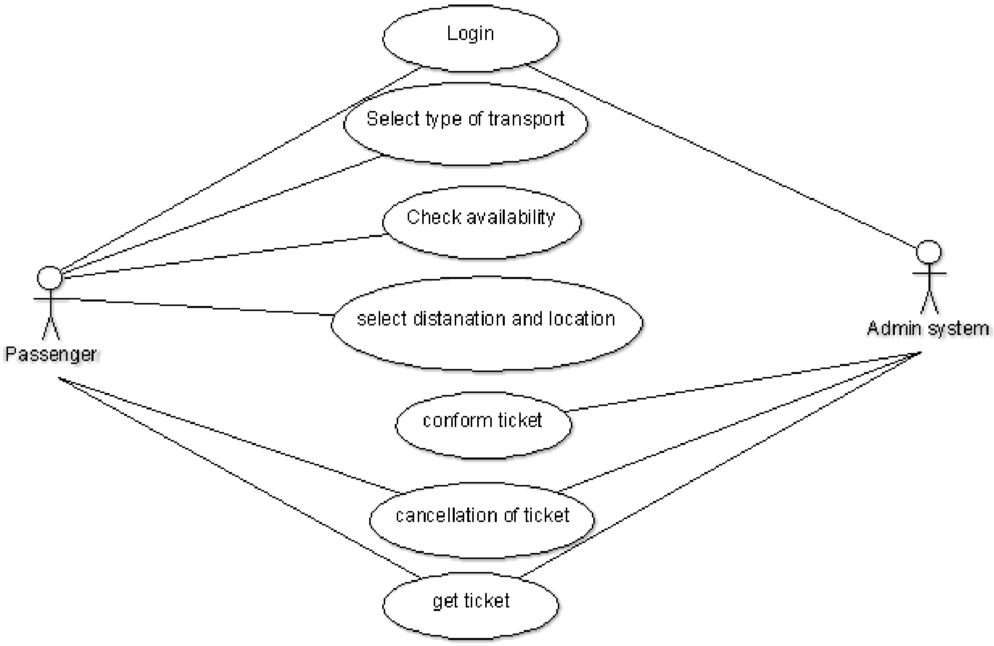

Start
Locket Ticketing
Online Ticketing
Channel
Go to Locket
Open Website / Apps
No
Found
Yes
Connected to Internet
No
Yes
Select Departure
No
Found
Yes
Select Destination
found
No
Yes
Insert Passenger Data
No
No more
Yes
Confirmation
Payment
Locket Ticketing
Online Ticketing
Channel
Receive Ticket / Booking Reference
Receive Email with Booking Reference Attached
Finish

CHAPTER 4

UML DIAGRAMS

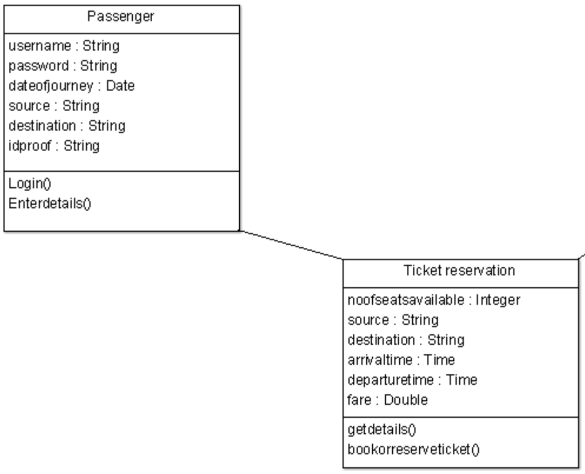
4.1 USE CASE DIAGRAM

It is used to describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.



4.2 CLASS DIAGRAM

A class diagram in the unified modelling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. Top compartment have the class name, middle compartment the attributes and the bottom compartment with operations.



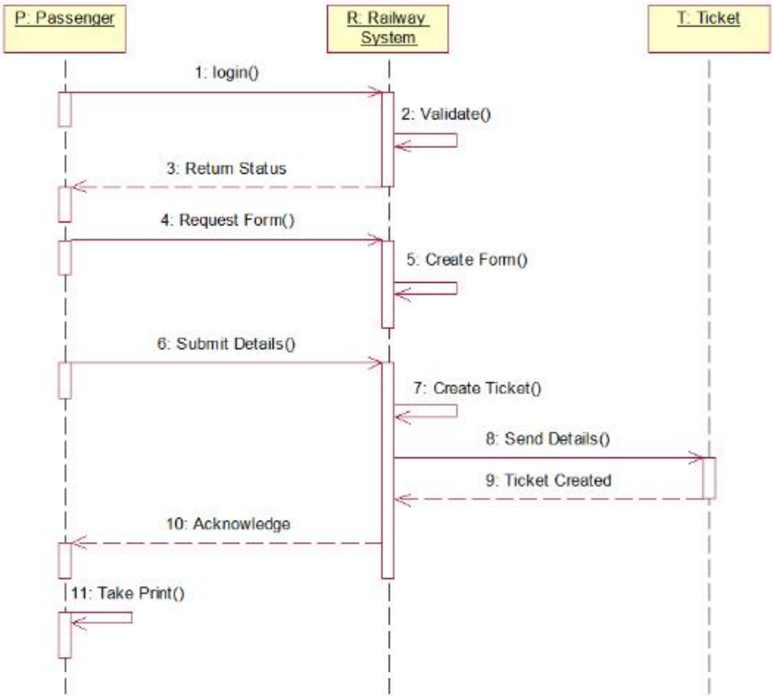
4.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

There are two dimensions

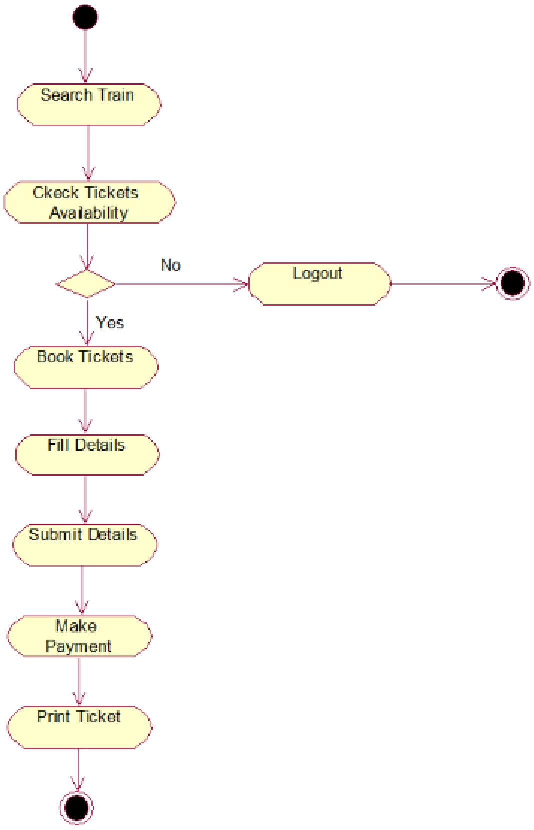
1.Veritcal dimension-represent time.

2.Horizontal dimension-represent different objects.



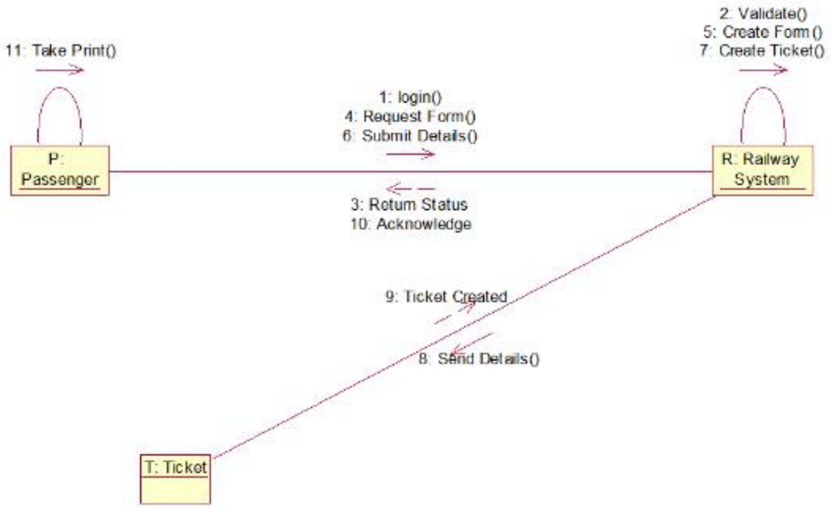
4.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step by step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as a rounded box containing the name of the operation.



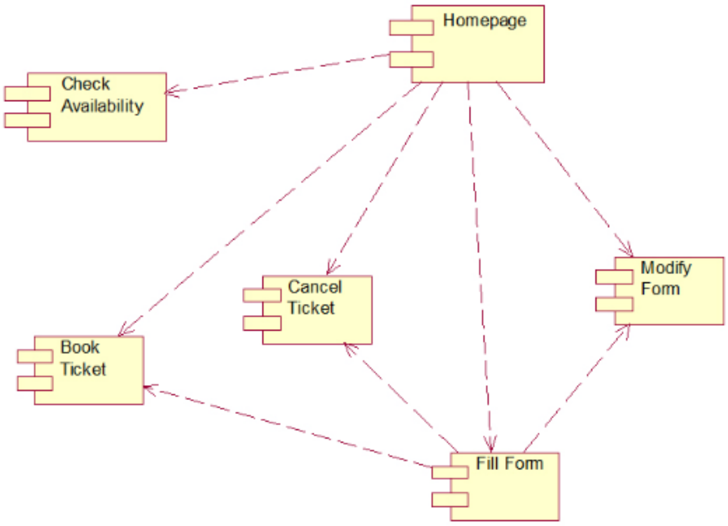
4.5 COLLABORATION DIAGRAM

A collaboration diagram, also called a communication diagram or interaction diagram,. A sophisticated modelling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time.



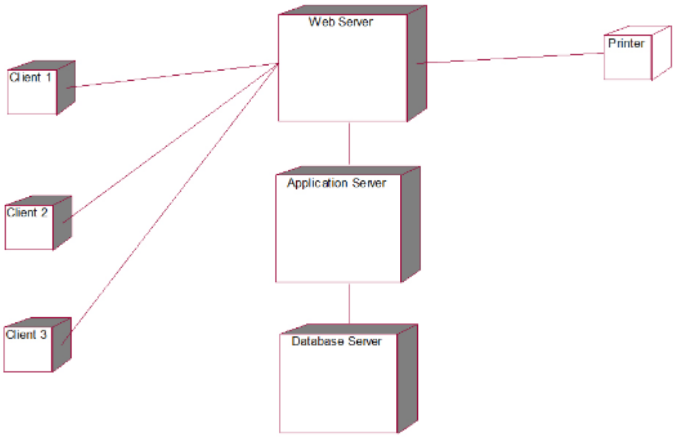
4.6 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



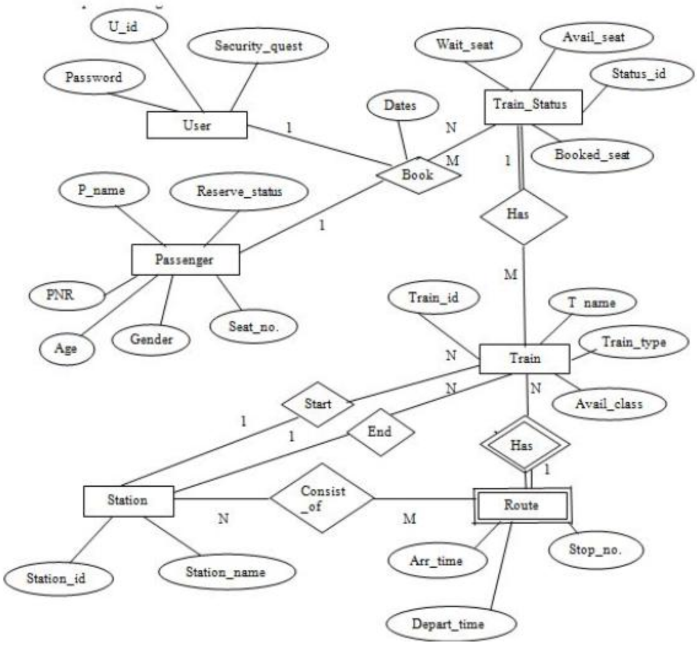
4.7 DEPLOYMENT DIAGRAM

In UML, deployment diagrams model the physical architecture of a system. Deployment diagrams show the relationships between the software and hardware components in the system and the physical distribution of the processing.



4.8 ER DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.



CHAPTER 5

SYSTEM SPECIFICATION

5.1 HARDWARE REQUIREMENTS

RAM : 8 GB RAM

Processor : Intel core i5-10300H CPU @2.50GHz,2496 Mhz,4 Cores

Monitor : 15 VGA Colour

System : 64 bit operating system

5.2 SOFTWARE REQUIREMENTS

Operating System Server : Windows 11

Coding Language : Java, Xml

Database : Firebase

Technologies : XML ,Java ,Android API version above 28

CHAPTER 6

SOFTWARE TOOLS USED

FRONT END

About XML in android: XML stands for Extensible Markup Language.XML is a markup language much like HTML used to describe data.XML tags are not predefined in XML. We must define our own Tags.XML as itself as well readable both by human and machine. Also, it is scalable and simple to develop. In Android we use xml for designing our layouts because xml is lightweight language so it doesn’t make our layout heavy.

BASICS OF USER INTERFACE:

The whole concept of Android User Interface is defined using the hierarchy of View and View group objects. A View group is an invisible container that organizes child views. These child views are other widgets which are used to make the different parts of UI.

Every Android application screen has some components like button, Text or images. These are contained inside the View Group. Layouts are the best examples for View Groups. The different types of layout in android are Linear Layout, Relative Layout, Absolute Layout, T able Layout and Frame Layout.

Different XML Files Used in Android:

1. Layout XML Files:

Layout xml files are used to define the actual UI (User interface) of our application. It holds all the elements (views) or the tools that we want to use in our application. Like the TextView’s, Button’s and other UI elements.

2. Manifest xml File(Mainfest.xml):

This xml is used to define all the components of our application. It includes the names of our application packages, our Activities, receivers, services and the permissions that our application needs. For Example – Suppose we need to use internet in our app then we need to define Internet permission in this file.

3. Strings xml File (strings.xml):

This xml file is used to replace the Hard-coded strings with a single string. We define all the strings in this xml file and then access them in our app(Activity or in Layout XML files) from this file. This file enhance the reusability of the code.

4. Styles xml File(styles.xml):

This xml is used to define different styles and looks for the UI(User Interface) of application. We define our custom themes and styles in this file.

5. Drawable xml Files:

These are those xml files that are used to provide various graphics to the elements or views of application. When we need to create a custom UI we use drawable xml files. Suppose if we need to define a gradient colour in the background of Button or any custom shape for a view then we create a Drawable xml file and set it in the background of View.

6. Colour xml File (colors.xml):

This file is used to define the colour codes that we used in our app. We simply define the colour's in this file and used them in our app from this file.

7. Dimension xml File (dimens.xml):

This xml file is used to define the dimensions of the View's. Suppose we need a Button with 50dp (density pixel) height then we define the value 50dp in dimens.xml file and then use it in our app from this file.

BACK END:

JAVA

Java is a language which is used at a major scale for the development of mobile applications. Some of the other languages are there which can be used to develop the Android application like C and C++ using Native Development Kit (NDK), but Java is recommended to develop the applications. In available IDE i.e. Android Studio, Java is default language to code whereas C and C++ are in options. It supports native code also, so C and C++ are applicable to code for the mobile application. An SDK (Software Development Kit) is used for Android development which has a bunch of libraries instead of JVM.

As we all know that Java and XML are used in the development of a mobile application. It is used in Android development to write the back-end logic or business logic. It is a high-level language. XML is used to design the UI(User Interface). It is an object oriented language but not pure object-oriented because of the presence of built-in data types.

Role of Java in Android App Development:

Java plays an important role in development of Android applications because business logic is written in Java. You can say that knowledge of core java is must for the development of android application. Knowledge of advance Java is a plus point for the development. With the knowledge of advance Java, you can add new features to the application. Let’s start to discuss the role of Java in android development by taking a look at the features of Java which makes it popular language for the Android development.

- Platform Independent
- Object-oriented
- Secure
- Interface

CHAPTER 7

IMPLEMENTATION

The proposed system is based on Android Operating system which will remind the users to take medicines on time through notification and automatic alarm ringing system. Android is a Linux-based operating system designed primarily for touch screen mobile devices such as smart phones and tablet computers, developed by Google in conjunction with the Open Handset Alliance. Android was built from the ground-up to enable developers to create compelling mobile applications that take full advantage of all a handset has to offer. The system is specified on android operating system only because the market share of Android is high. Android also comes with an application development framework (ADF), which provides an API for application development and includes services for building GUI applications, data access, and other component types. The framework is designed to simplify the reuse and integration of components. Android apps are built using a mandatory XML manifest file. The manifest file values are bound to the application at compile time. This file provides essential information to an Android platform for managing the life cycle of an application.

CHAPTER 8

SYSTEM TESTING

Method Accepted for system testing

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before leave operation commences. An elaborate testing of data is prepared and the system is based using the test data. While testing errors noted and corrections are measured. The users are trained to operate the developed system.

Testing objectives: Testing is a process of executing a program with the intent of finding an error. Good test case is one that has a probability of finding an as yet undiscovered error. A successful test is one that uncovers an uncovered error.

Testing Principle:

•All tests should be traceable to end user requirements.

•Tests should be planned before long test begins.

•Testing should begin on small scale and progress towards testing in large.

•Exhaustive testing is not possible.

•To be most effective, testing should be conducted by an independent party.

Testing strategies:

A strategy for software testing integrates software test cases into a series of well planned steps that result in a successful construction of software. Software testing is a broader topic for what is referred to as verification and validation. Verification refers to the set of activities that ensure that the software that has been built is traceable to customer’s requirements.

Testing Steps:

•Unit Testing

•Integration Testing

•Acceptance Testing

Unit Testing

Unit testing means testing each units of design separately. Here in this project we tested each unit of design separately and verify that there were no errors. For this testing each design is run individually after executing each page if there any error occurs correction mechanism is done instantly.

Integration Testing

In our project we combine many units module to form a sub system. These sub systems are then tested. This is done to see whether the modules can be integrated properly. Based on integration testing some changes made to the design.

Acceptance testing

The goal of acceptance testing is to see if the software meets all the requirements as needed. The testing was performed by data of all the users of the system. It was found that the software meets all the requirements of the as needed.

CHAPTER 9

CODING

XML CODE:

activity_main.xml:

```
<?xml version="1.0" encoding="UTF-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/topView1"
        android:layout_width="0sp"
        android:layout_height="150sp"
        android:background="@color/dark_magenta"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.526"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/topText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="40sp"
        android:text="Project Title"
        android:textColor="@color/white"
        android:textSize="34sp"
        android:textStyle="bold"
        app:layout_constraintTop_toTopOf="@+id/topView1"
        tools:ignore = "HardcodedText"/>

    <View
        android:id="@+id/View1"
        android:layout_width="0sp"
        android:layout_height="100sp"
        android:background="@drawable/ic_wave"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/topView1" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="347sp"
        android:layout_height="250sp"
        android:layout_marginStart="32sp"
        android:layout_marginTop="96dp"
        android:layout_marginEnd="32sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/View1"
        app:srcCompat="@drawable/starting_page_image" />
```

activityLoginPage:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LoginPage">

    <TextView
        android:id="@+id/topView"
        android:layout_width="0sp"
        android:layout_height="150sp"
        android:background="@color/dark_magenta"
        android:textColor="#E1BEE7"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <View
        android:id="@+id/view2"
        android:layout_width="0sp"
        android:layout_height="100sp"
        android:background="@drawable/ic_wave"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/topView"
    app:layout_constraintVertical_bias="0.0" />

<ImageView
    android:id="@+id/imageView4"
    android:layout_width="97sp"
    android:layout_height="96sp"
    android:layout_marginStart="16sp"
    android:layout_marginTop="32sp"
    android:layout_marginEnd="24sp"
    android:contentDescription="TODO"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/topText"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/ic_baseline_person_white_24" />

    <EditText
    android:id="@+id/login_email"
    android:layout_width="0sp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32sp"
    android:layout_marginEnd="32sp"
    android:background="@drawable/round_border"
    android:drawableStart="@drawable/ic_baseline_email_24"
    android:drawablePadding="16sp"
    android:ems="10"
    android:hint="Email"
    android:inputType="text"
    android:minHeight="48sp"
    android:padding="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/view2" />

<EditText
    android:id="@+id/login_password"
    android:layout_width="0sp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32sp"
    android:layout_marginTop="16sp"
    android:layout_marginEnd="32sp"
    android:background="@drawable/round_border"
    android:drawableStart="@drawable/ic_baseline_lock_24"
    android:drawablePadding="16sp"
    android:ems="10"
    android:hint="Password"
    android:inputType="textPassword"
    android:minHeight="48sp"
    android:padding="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/login_email" />

<Button
    android:id="@+id/loginbtn"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="32sp"
    android:layout_marginTop="16sp"
    android:layout_marginEnd="32sp"
    android:background="@drawable/round_bg"
    android:text="LOGIN"
    android:textColor="@color/white"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/login_password"
    tools:ignore = "HardcodedText"/>

<TextView
    android:id="@+id/forget_Password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32sp"
    android:layout_marginTop="16sp"
    android:layout_marginEnd="32sp"
    android:text="Forget Password?"
    android:textColor="@color/dark_magenta"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/loginbtn" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="32sp"
    android:layout_marginTop="32sp"
    android:layout_marginEnd="32sp"
    android:text="OR"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/forget_Password" />

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="0sp"
    android:layout_height="30sp"
    android:layout_marginStart="32sp"
    android:layout_marginTop="16sp"
    android:layout_marginEnd="32sp"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4">

    <ImageView
        android:id="@+id/googleLogo"
        android:layout_width="218sp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        app:srcCompat="@drawable/google" />

    <ImageView
        android:id="@+id/facebookLogo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
            android:layout_weight="1"
            app:srcCompat="@drawable/fb" />

    </LinearLayout>

    <TextView
        android:id="@+id/goto_sign_up"
        android:layout_width="0sp"
        android:layout_height="wrap_content"
        android:layout_marginStart="32sp"
        android:layout_marginTop="32sp"
        android:layout_marginEnd="32sp"
        android:layout_weight="1"
        android:gravity="center_horizontal"
        android:text="New User?Sign Up"
        android:textColor="@color/dark_magenta"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/linearLayout" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

activityHomePage:

```
<?xml version="1.0" encoding="UTF-8" ?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
android:background="@drawable/background">

<View
    android:id="@+id/homeView"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:background="@drawable/background2" />

<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:padding="20dp">

    <ImageView
        android:id="@+id/home_menu_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        app:srcCompat="@drawable/ic_baseline_menu_24" />

    <TextView
        android:id="@+id/homeText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome"
        android:textColor="@color/white"
        android:layout_below="@+id/home_menu_btn"
        android:textSize="42sp"
        android:textStyle="bold"
        tools:ignore="HardcodedText" />

<ImageView
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_alignParentRight="true"
    android:layout_margin="10dp"
    android:src="@drawable/ic_project_logo" />

<TabHost
    android:id="@+id/tabHost"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/homeView"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <FrameLayout
            android:id="@android:id/tabcontent"
```

```
android:layout_width="match_parent"
android:layout_height="match_parent">

<LinearLayout
    android:id="@+id/tab1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="1dp"
    android:background="@drawable/taxi_new_new"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/taxiQuotes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="30dp"
            android:alpha="0.6"
            android:background="@drawable/text_border"
            android:padding="10dp"
            android:text="Our customers and drivers must wear a protective mask during the entire journey."
            android:textColor="@color/black"
            android:textSize="25sp"
            tools:ignore="HardcodedText" />

        <Button
            android:id="@+id/cab_book_btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/taxiQuotes"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:layout_marginTop="25dp"
            android:background="@drawable/image_border2"
            android:text="Book Now"
            android:textSize="22sp"
            android:textStyle="italic"
            tools:ignore="HardcodedText" />
    </RelativeLayout>
</LinearLayout>

<LinearLayout
    android:id="@+id/tab2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="1dp"
    android:background="@drawable/bus_new_new"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/busQuotes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
                    android:layout_margin="30dp"
                    android:alpha="0.6"
                    android:background="@drawable/text_border"
                    android:padding="10dp"
                    android:text="Our customers and drivers must wear a protective mask
during the entire journey."
                    android:textColor="@color/black"
                    android:textSize="25sp"
                    tools:ignore="HardcodedText" />


                <Button
                    android:id="@+id/bus_book_btn"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_below="@+id/busQuotes"
                    android:layout_centerHorizontal="true"
                    android:layout_centerVertical="true"
                    android:layout_marginTop="25dp"
                    android:background="@drawable/image_border2"
                    android:text="Book Now"
                    android:textSize="22sp"
                    android:textStyle="italic"
                    tools:ignore="HardcodedText" />
            </RelativeLayout>
        </LinearLayout>


        <LinearLayout
            android:id="@+id/tab3"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="1dp"
```

```
    android:background="@drawable/train_new_new"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/trainQuotes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="30dp"
            android:alpha="0.6"
            android:background="@drawable/text_border"
            android:padding="10dp"
            android:text="Our customers and drivers must wear a protective mask
during the entire journey."
            android:textColor="@color/black"
            android:textSize="25sp"
            tools:ignore="HardcodedText" />

        <Button
            android:id="@+id/train_book_btn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/trainQuotes"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:layout_marginTop="25dp"
            android:background="@drawable/image_border2"
            android:text="Book Now"
            android:textSize="22sp"
            android:textStyle="italic"
            tools:ignore="HardcodedText" />
    </RelativeLayout>
</LinearLayout>

<LinearLayout
    android:id="@+id/tab4"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="1dp"
    android:background="@drawable/flight_new"
    android:orientation="vertical">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/flightQuotes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="30dp"
            android:alpha="0.4"
            android:background="@drawable/text_border"
            android:padding="10dp"
            android:text="Our customers and drivers must wear a protective mask during the entire journey."
            android:textColor="@color/black"
            android:textSize="25sp"
            tools:ignore="HardcodedText" />
```

```
                    <Button
                        android:id="@+id/flight_book_btn"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_below="@+id/flightQuotes"
                        android:layout_centerHorizontal="true"
                        android:layout_centerVertical="true"
                        android:layout_marginTop="25dp"
                        android:background="@drawable/image_border2"
                        android:text="Book Now"
                        android:textSize="22sp"
                        android:textStyle="italic"
                        tools:ignore="HardcodedText" />
                </RelativeLayout>
            </LinearLayout>


        </FrameLayout>
    </LinearLayout>
  </TabHost>
</RelativeLayout>
```

JAVA CODE

Main Activity.java:

```
package com.example.bookingapplication;


import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class MainActivity extends AppCompatActivity {

    FirebaseAuth mAutho;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        mAutho = FirebaseAuth.getInstance();

       new Handler().postDelayed(new Runnable() {
           @Override
           public void run() {
               Intent intent = new Intent(MainActivity.this,LoginPage.class);
               startActivity(intent);
           }
       },3000);
    }
}
```

LoginPage.java:

```
package com.example.bookingapplication;
```

```
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class  LoginPage extends AppCompatActivity {

    Button login;
    EditText email, password;
    TextView gotoSignUp,forgetPassword;
    FirebaseAuth mAuth;



    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login_page);
    email = (EditText) findViewById(R.id.login_email);
    password = (EditText) findViewById(R.id.login_password);
    login = (Button) findViewById(R.id.loginbtn);
    gotoSignUp = (TextView) findViewById(R.id.goto_sign_up);
    mAuth = FirebaseAuth.getInstance();
    login.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            loginUser();
        }
    });
    gotoSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(LoginPage.this, Sign_up_page.class));
        }
    });
}

private void loginUser() {
    String lemail = email.getText().toString();
    String lpwd = password.getText().toString();

    if (TextUtils.isEmpty(lemail)) {
        email.setError("Please Enter Email!");
        email.requestFocus();
        } else if (TextUtils.isEmpty(lpwd)) {
            password.setError("Please Enter  Password");
            password.requestFocus();
        }
        else
        {
            mAuth.signInWithEmailAndPassword(lemail, lpwd).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        Toast.makeText(LoginPage.this, "Login Successful",
Toast.LENGTH_LONG).show();
                        startActivity(new Intent(LoginPage.this, HomePageActivity.class));
                    } else {
                        Toast.makeText(LoginPage.this, "Login Failed" +
task.getException().getMessage(), Toast.LENGTH_LONG).show();
                    }
                }
            });
        }
    }
}
```

signUpPage.java :

```
package com.example.bookingapplication;


import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap;

public class Sign_up_page extends AppCompatActivity {

    FirebaseAuth mAuth;
    DatabaseReference mdref;
    EditText emailText, userName, address, password;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up_page);

        Button signUpBtn = (Button) findViewById(R.id.signUpBtn);
```

```
TextView gotoLogin = (TextView) findViewById(R.id.goto_login);
emailText = (EditText) findViewById(R.id.sign_up_userEmail);
userName = (EditText) findViewById(R.id.sign_up_user_Name);
address = (EditText) findViewById(R.id.sign_up_address);
password = (EditText) findViewById(R.id.sign_up_pass);

mAuth = FirebaseAuth.getInstance();
mdref = FirebaseDatabase.getInstance().getReference();
signUpBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        String email = emailText.getText().toString();
        String user_name = userName.getText().toString();
        String user_address = address.getText().toString();
        String pwd= password.getText().toString();

        HashMap<String, Object> hashmap = new HashMap<>();
        hashmap.put("name", user_name);
        hashmap.put("email", email);
        hashmap.put("Address", user_address);
        hashmap.put("Password",pwd);

        if (TextUtils.isEmpty(email)) {
            emailText.setError("Email Cannot Be Empty");
            emailText.requestFocus();
        } else if (TextUtils.isEmpty(user_name)) {
            userName.setError("Please Enter User Name..!");
            userName.requestFocus();
        } else if (TextUtils.isEmpty(pwd)) {
            password.setError("Please Enter a Password..");
            password.requestFocus();
        } else if (TextUtils.isEmpty(pwd)) {
            password.setError("please ReEnter Your Password..");
            password.requestFocus();


        } else {
            mAuth.createUserWithEmailAndPassword(email,
pwd).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
              if (task.isSuccessful()) {
                mdref.child("users").child(user_name).setValue(hashmap);


                Toast.makeText(Sign_up_page.this, "SignUp Successfully",
Toast.LENGTH_LONG).show();


                startActivity(new Intent(Sign_up_page.this, LoginPage.class));
              } else {
                Toast.makeText(Sign_up_page.this, "SignUp Failed :" +
task.getException().getMessage(), Toast.LENGTH_LONG).show();


              }
            }
            });
        }
    }


});
gotoLogin.setOnClickListener(new View.OnClickListener() {
    @Override
```

```
            public void onClick(View view) {


            }
        });


    }
}
```

HomePage.java:

```
package com.example.bookingapplication;


import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TabHost;
import android.widget.TextView;


import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;


import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

```
public class HomePageActivity extends AppCompatActivity {
    TextView homeUserEmail;

    FirebaseAuth firebaseAuth;

    ImageView homeMenuBtn;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home_page);
        firebaseAuth = FirebaseAuth.getInstance();

        homeMenuBtn=(ImageView) findViewById(R.id.home_menu_btn);

        homeMenuBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(HomePageActivity.this,MenuActivity.class));
            }
        });

        homeUserEmail = (TextView) findViewById(R.id.homeUserEmail);
        if (firebaseAuth.getCurrentUser() == null) {
            finish();
            startActivity(new Intent(HomePageActivity.this, Sign_up_page.class));
        }
```

```
TabHost host = (TabHost) findViewById(R.id.tabHost);
host.setup();

TabHost.TabSpec spec = host.newTabSpec("Tab One");
spec.setContent(R.id.tab1);
spec.setIndicator("Cab");
host.addTab(spec);



spec = host.newTabSpec("Tab Two");
spec.setContent(R.id.tab2);
spec.setIndicator("Bus");
host.addTab(spec);

spec = host.newTabSpec("Tab Three");
spec.setContent(R.id.tab3);
spec.setIndicator("Train");
host.addTab(spec);

spec = host.newTabSpec("Tab Four");
spec.setContent(R.id.tab4);
spec.setIndicator("Flight");
host.addTab(spec);

Button cab_btn = (Button) findViewById(R.id.cab_book_btn);
Button bus_btn = (Button) findViewById(R.id.bus_book_btn);
Button train_btn = (Button) findViewById(R.id.train_book_btn);
Button flight_btn = (Button) findViewById(R.id.flight_book_btn);
```

```
    cab_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(HomePageActivity.this, cabActivity.class));
        }
    });

    bus_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(HomePageActivity.this, Bus_List_View.class));
        }
    });
    train_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(HomePageActivity.this, train_list_view.class));
        }
    });
    flight_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new Intent(HomePageActivity.this,flightListView.class));
        }
    });

}
protected  void onStart() {
        super.onStart();
        FirebaseUser user = firebaseAuth.getCurrentUser();
        String userID = user.getUid();
        homeUserEmail.setText(user.getEmail());
    }

}
```

trainActivity.java:

```
package com.example.bookingapplication;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import com.example.bookingapplication.databinding.ActivityMainBinding;

public class trainActivity extends AppCompatActivity {


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

}
```

flightActivity.java:

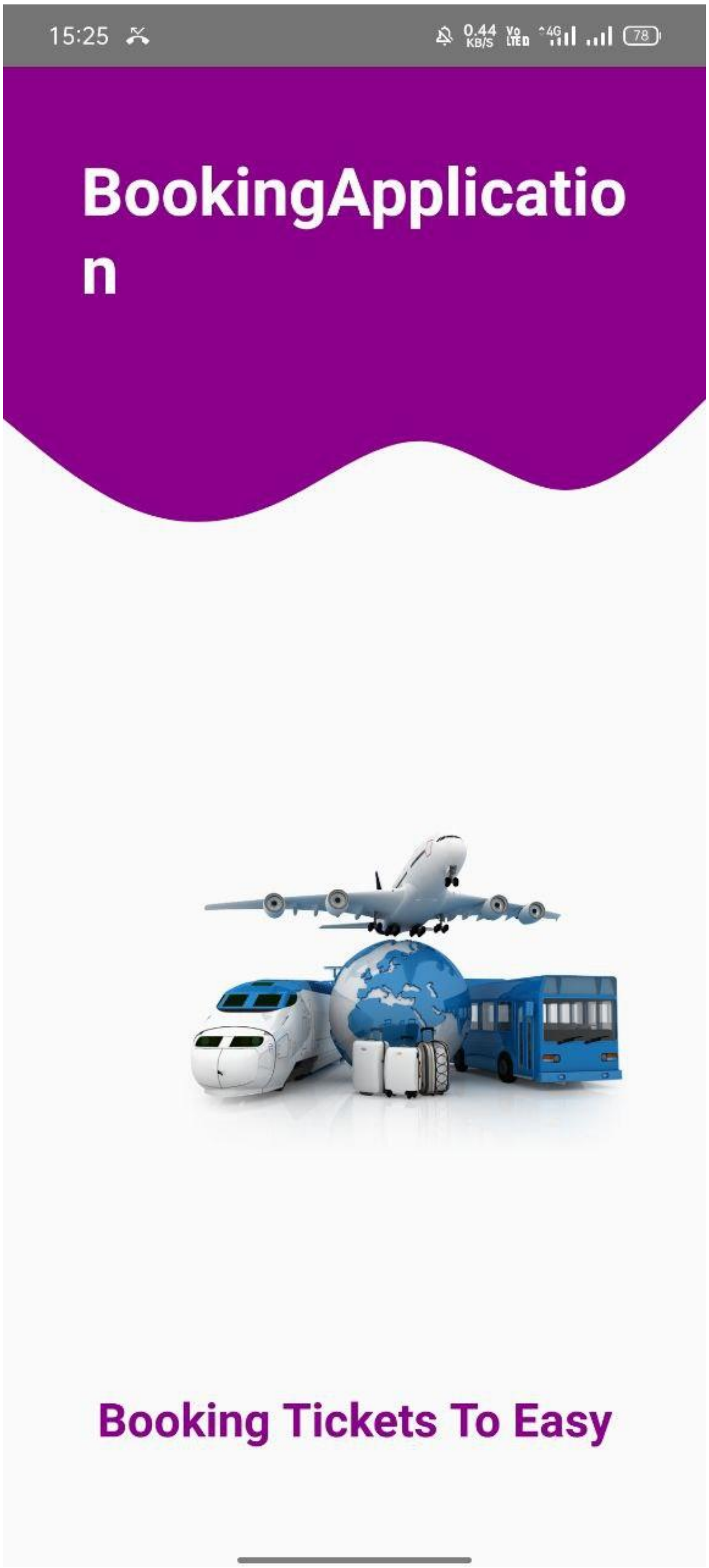
```
package com.example.bookingapplication;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import com.example.bookingapplication.databinding.ActivityMainBinding;

public class flightActivity extends AppCompatActivity {


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

}
```

BusActivity.java:

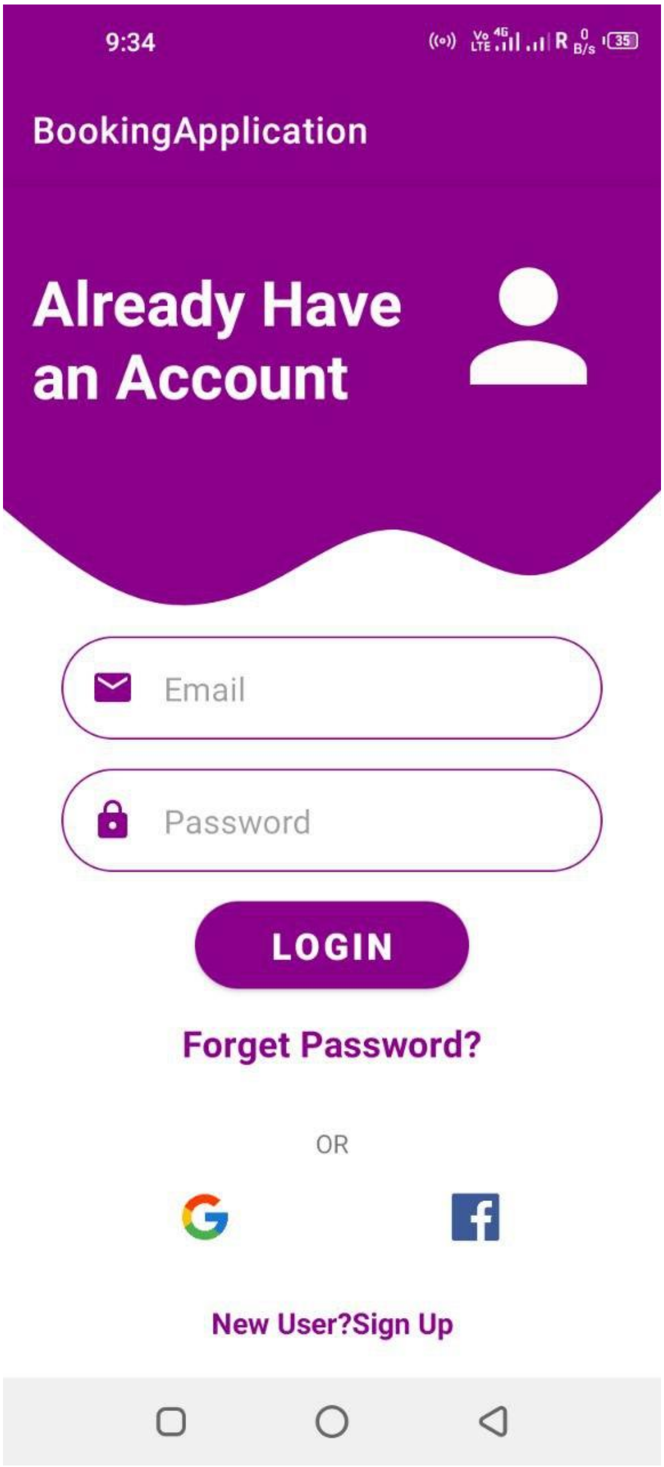
```
package com.example.bookingapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.Toast;

public class busActivity extends AppCompatActivity {

    AutoCompleteTextView fromLocation;
    AutoCompleteTextView dropLocation;
    String _fromLocation, _dropLocation;
    Button busSearchBtn;



    String[] locations = {"Chennai", "Toothukkudi", "Erode", "Vellore", "Tiruppur",
"Tiruchirappalli", "Tirunelveli", "Madurai", "Coimbatore"};

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

        fromLocation = (AutoCompleteTextView) findViewById(R.id.bus_pickUp_location);
        dropLocation = (AutoCompleteTextView) findViewById(R.id.bus_drop_location);

        ArrayAdapter<String> pickupAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, locations);
        fromLocation.setThreshold(1);
        fromLocation.setAdapter(pickupAdapter);

        fromLocation.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                _fromLocation = pickupAdapter.getItem(i);
            }
        });

        ArrayAdapter<String> dropAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, locations);
        dropLocation.setThreshold(1);
        dropLocation.setAdapter(dropAdapter);

        dropLocation.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
                _dropLocation = dropAdapter.getItem(i);
            }
        });

        busSearchBtn = (Button) findViewById(R.id.bus_search_btn);

        busSearchBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(busActivity.this, _fromLocation + "\n"+_dropLocation,
Toast.LENGTH_SHORT).show();
            }
        });
    }
```

CHAPTER 10

SCREEN SHOTS



LOGIN PAGE

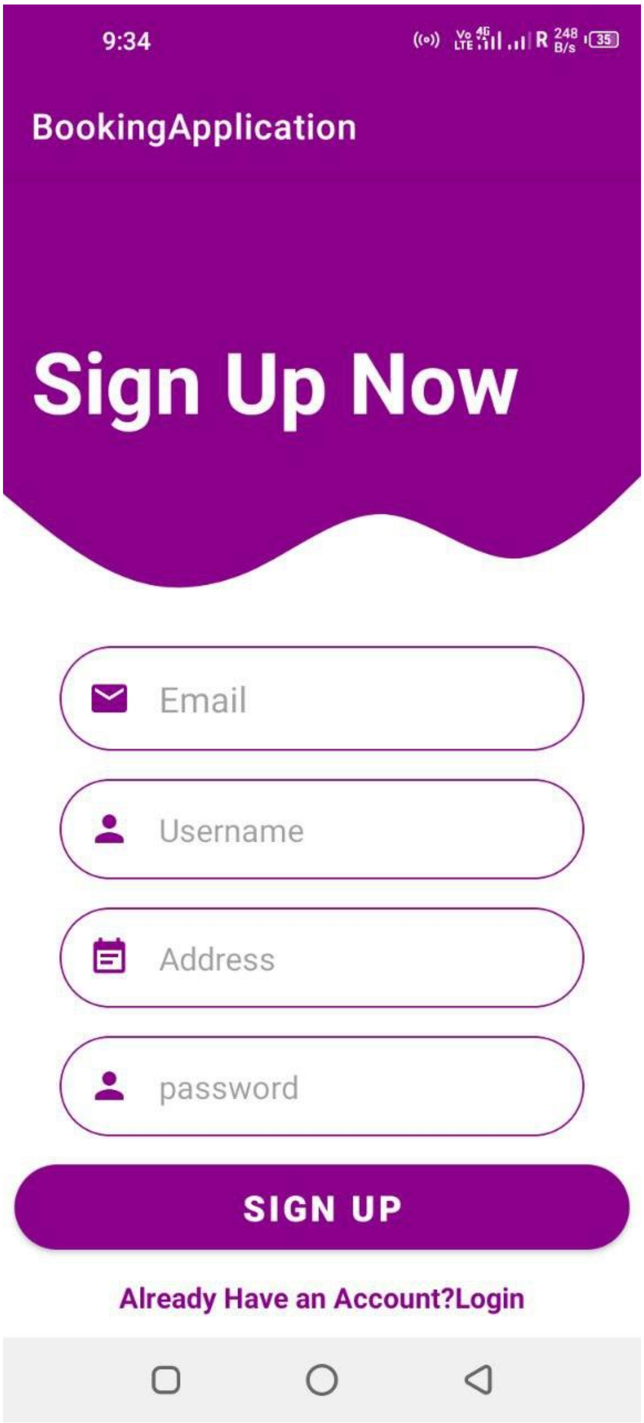


9:34

BookingApplication

Already Have an Account

SIGN IN PAGE



Sign Up Now

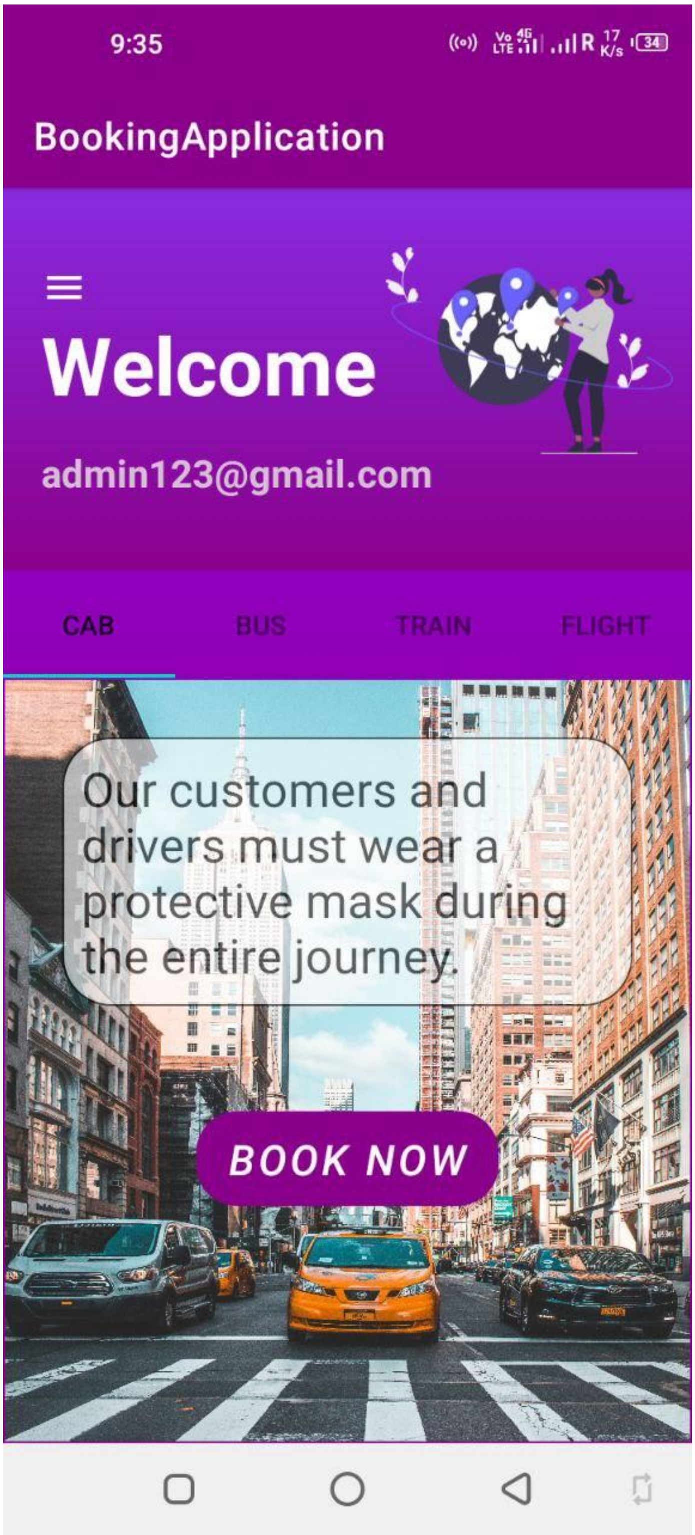
Email

Username

Address

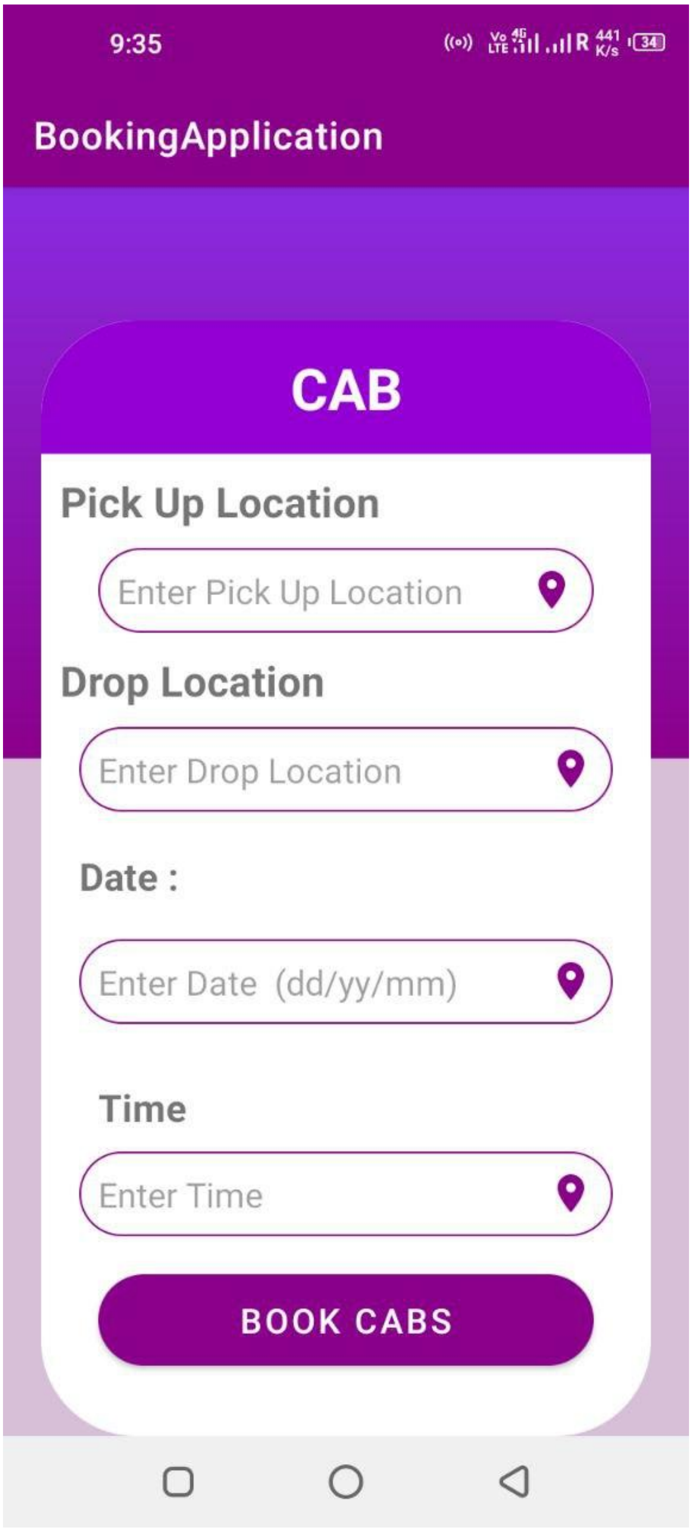
password

CAB HOME PAGE



Our customers and drivers must wear a protective mask during the entire journey.

LOCATION CHOOSING PAGE



9:35

BookingApplication

CAB

Pick Up Location

Enter Pick Up Location

Drop Location

Enter Drop Location

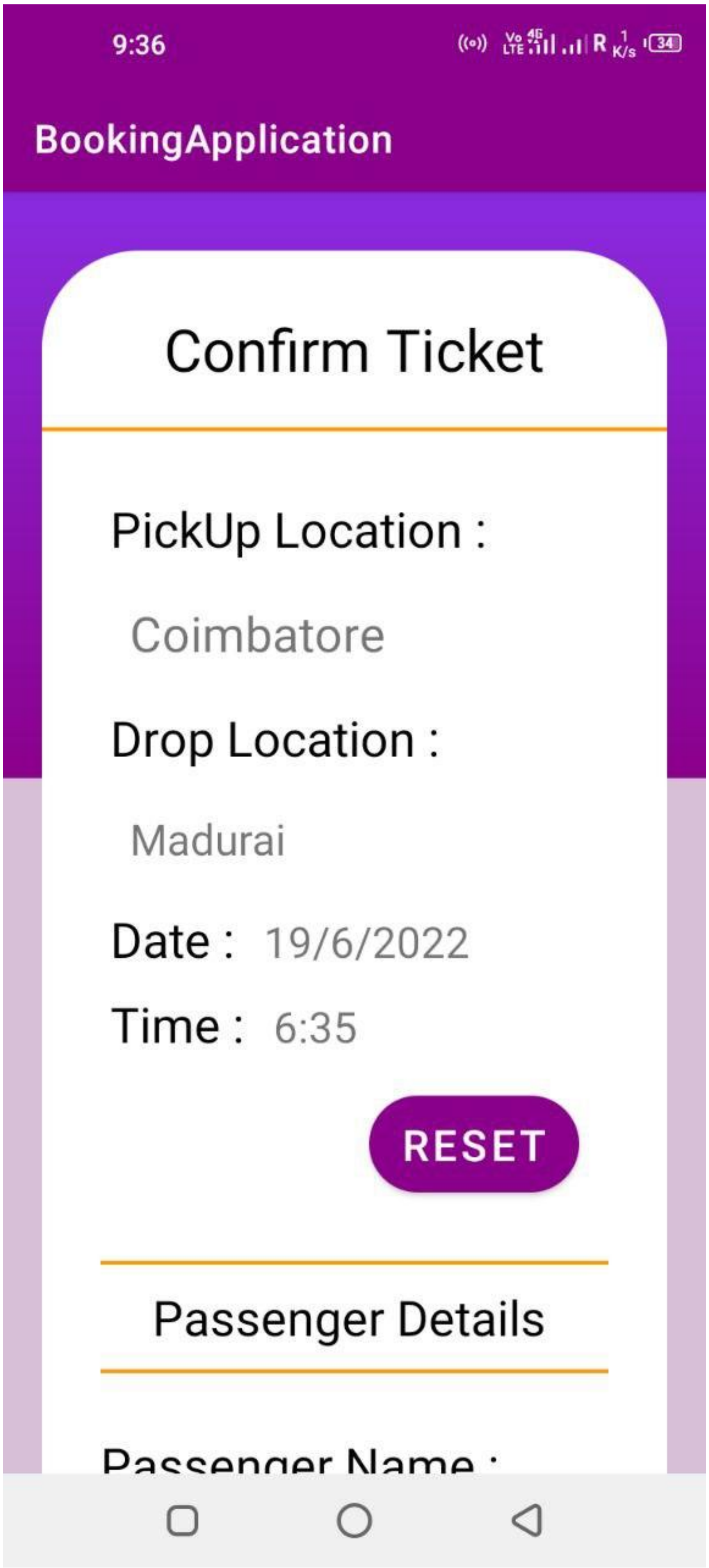
Date :

Enter Date (dd/yy/mm)

Time

Enter Time

CONFIRMING TICKET PAGE



Confirm Ticket

PickUp Location :

Coimbatore

Drop Location :

Madurai

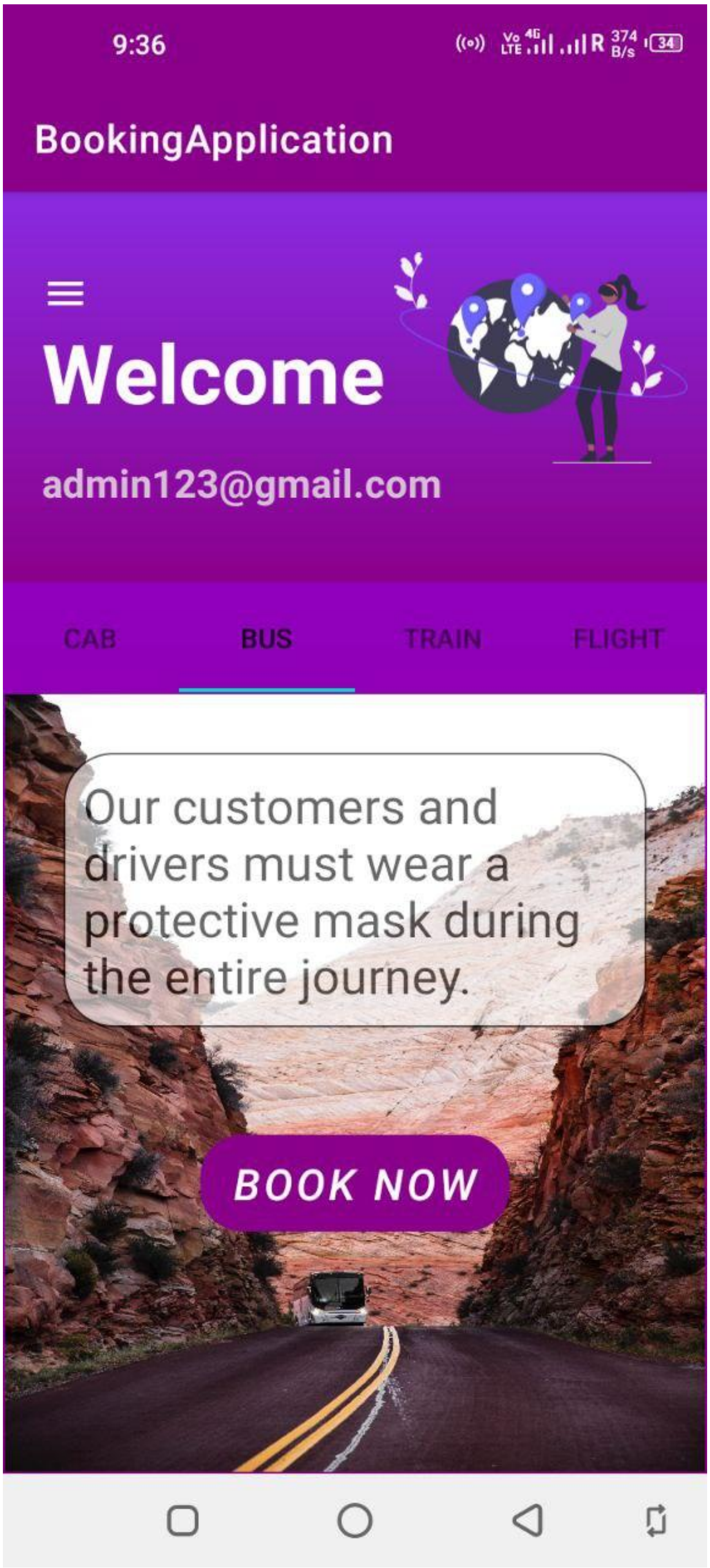
Date : 19/6/2022

Time : 6:35

Passenger Details

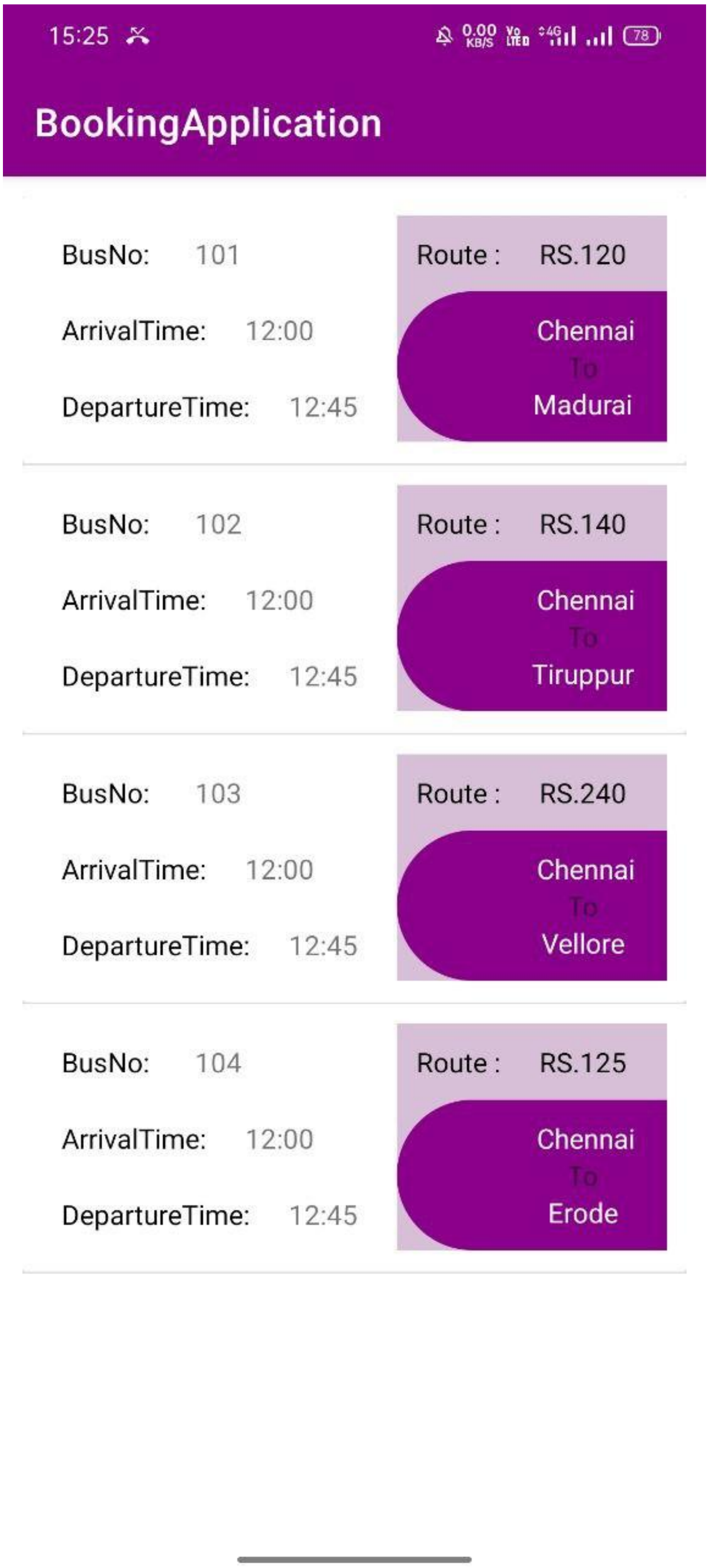
Passenger Name :

BUS HOME PAGE



Our customers and drivers must wear a protective mask during the entire journey.

CHOOSING LOCATION PAGE



15:25

BookingApplication

BusNo: 101

ArrivalTime: 12:00

DepartureTime: 12:45

Route : RS.120

Chennai
To
Madurai

BusNo: 102

ArrivalTime: 12:00

DepartureTime: 12:45

Route : RS.140

Chennai
To
Tiruppur

BusNo: 103

ArrivalTime: 12:00

DepartureTime: 12:45

Route : RS.240

Chennai
To
Vellore

BusNo: 104

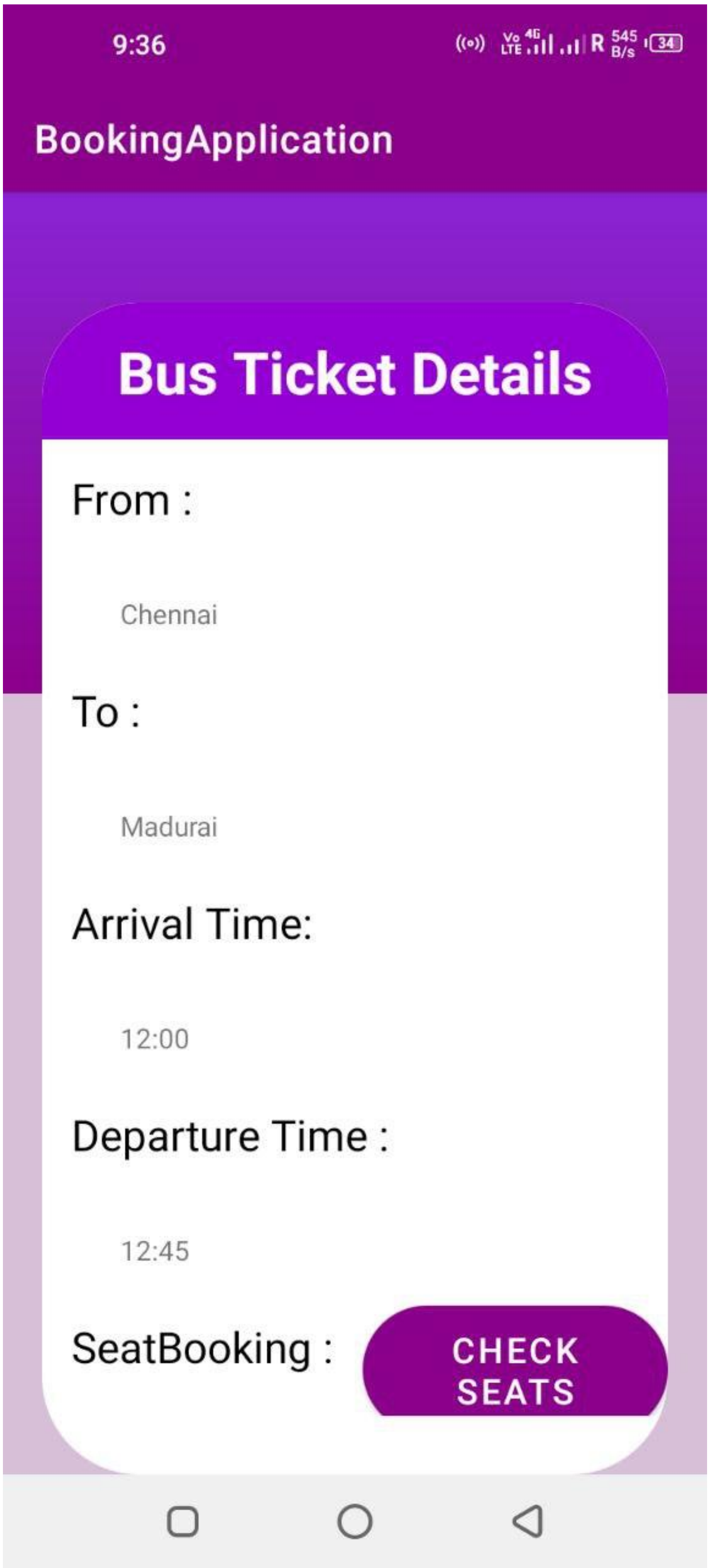
ArrivalTime: 12:00

DepartureTime: 12:45

Route : RS.125

Chennai
To
Erode

DETAILS PAGE



9:36

BookingApplication

Bus Ticket Details

From :

Chennai

To :

Madurai

Arrival Time:

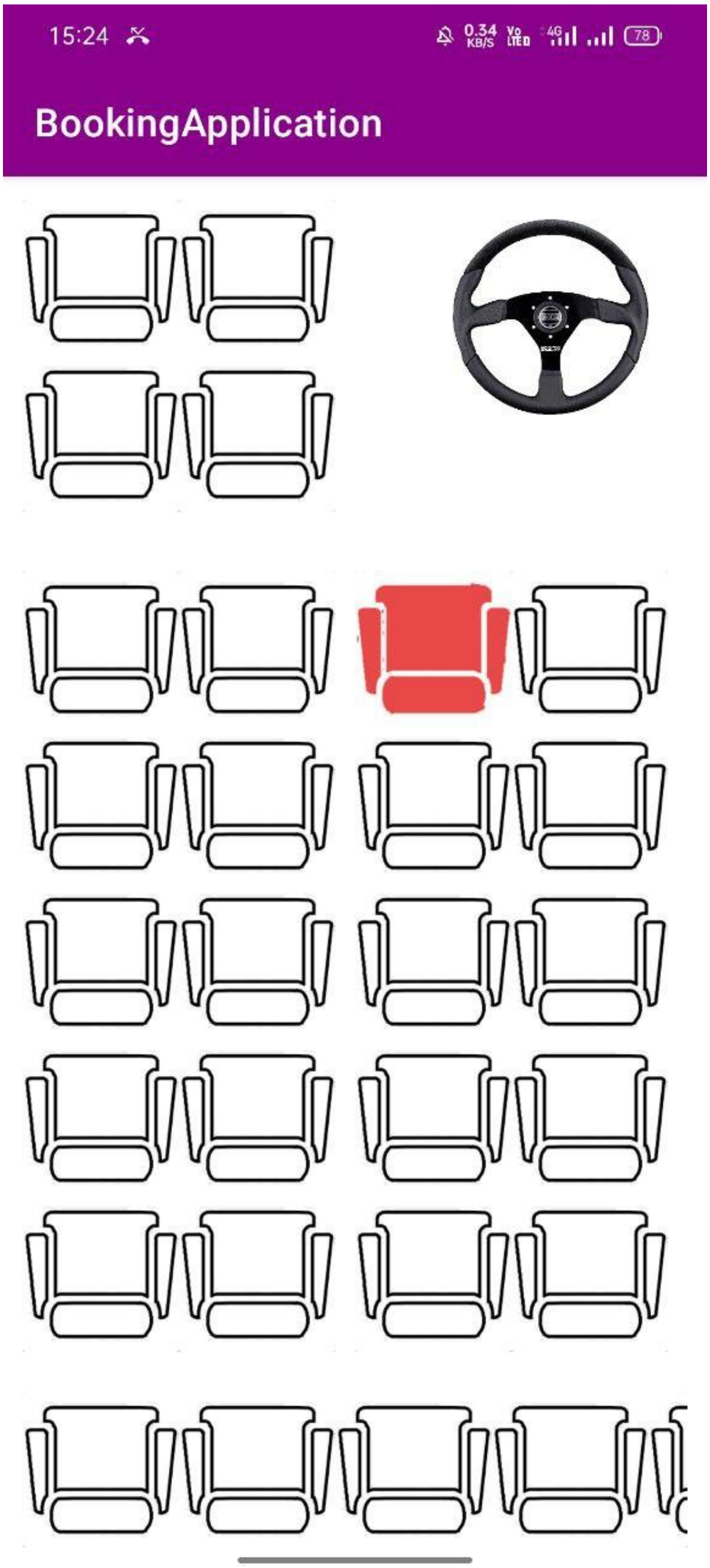
12:00

Departure Time :

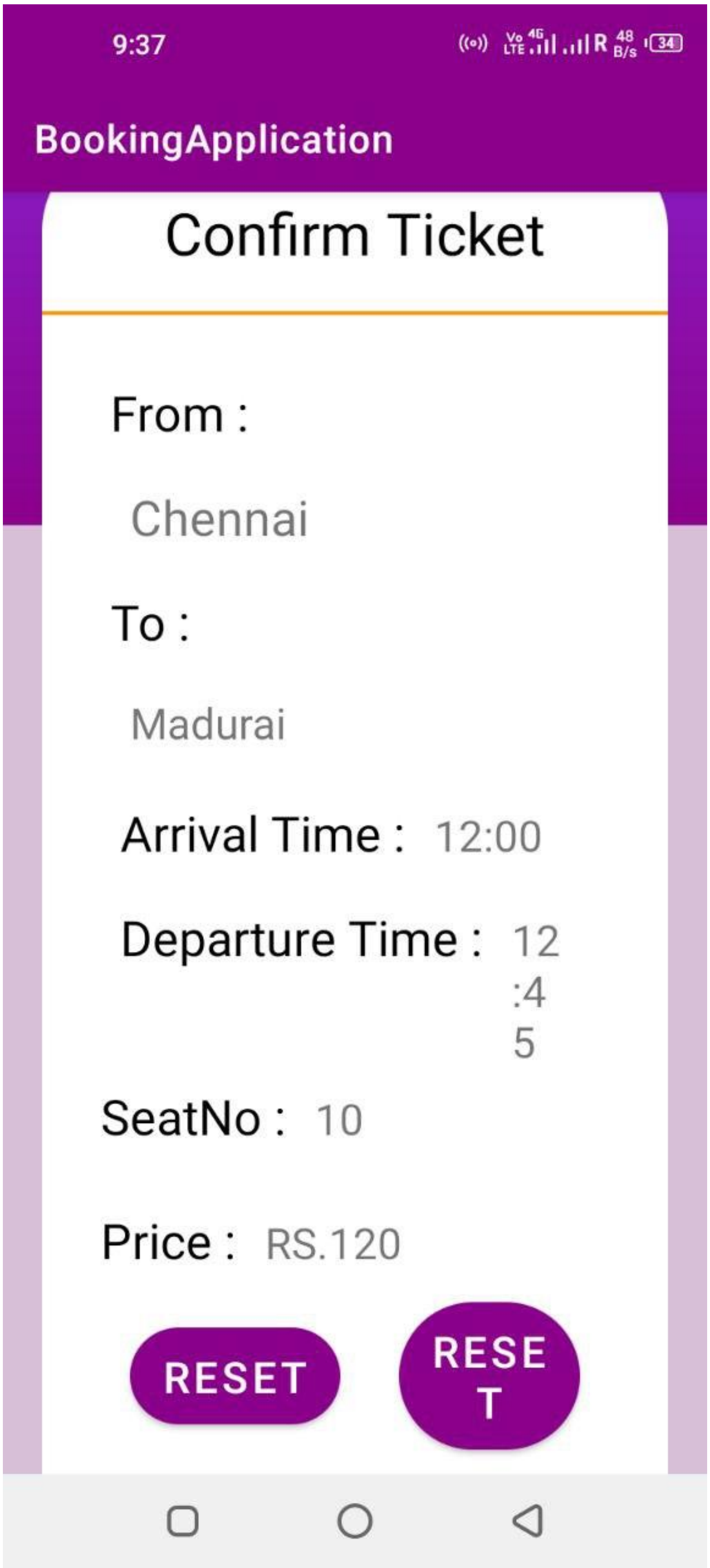
12:45

SeatBooking :

SEAT CHOOSING PAGE



CONFIRMING PAGE



9:37

BookingApplication

Confirm Ticket

From :

Chennai

To :

Madurai

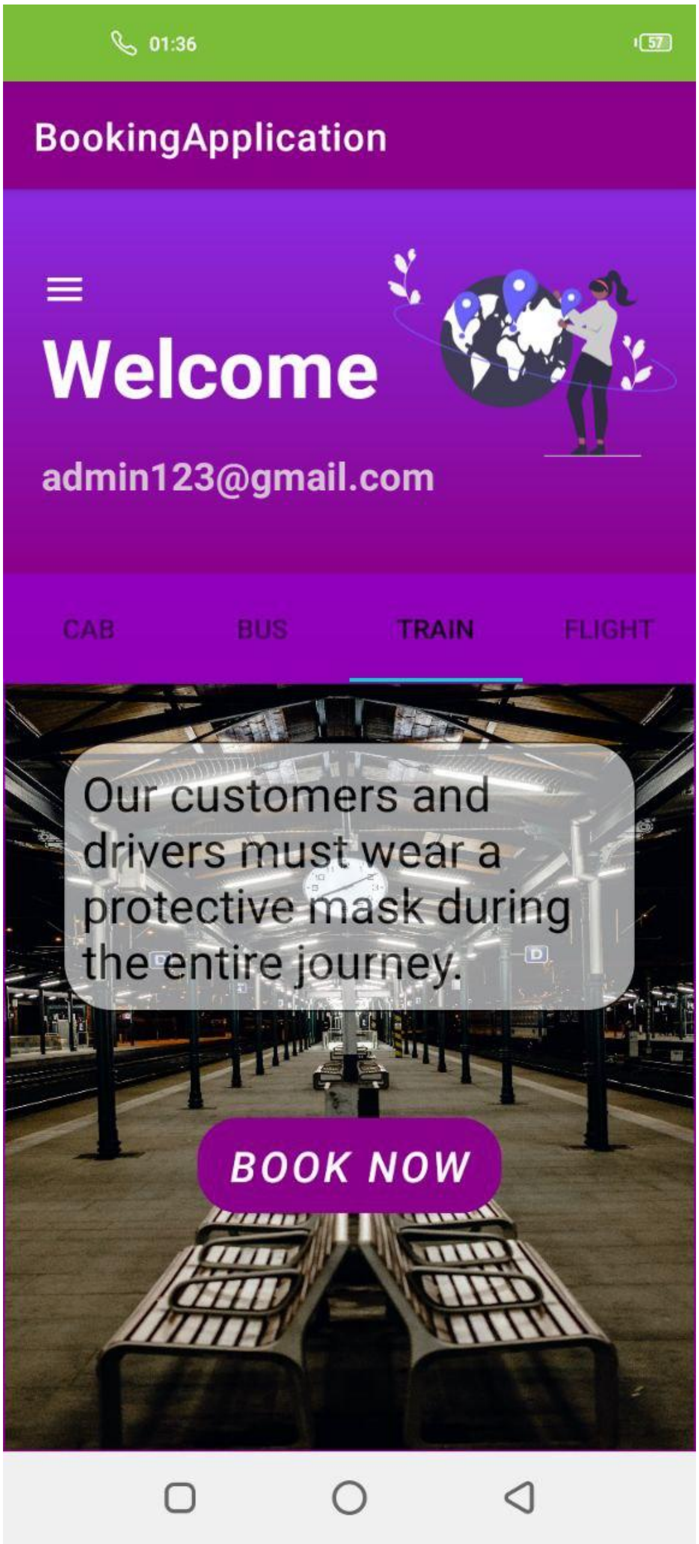
Arrival Time : 12:00

Departure Time : 12
:4
5

SeatNo : 10

Price : RS.120

TRAIN HOME PAGE

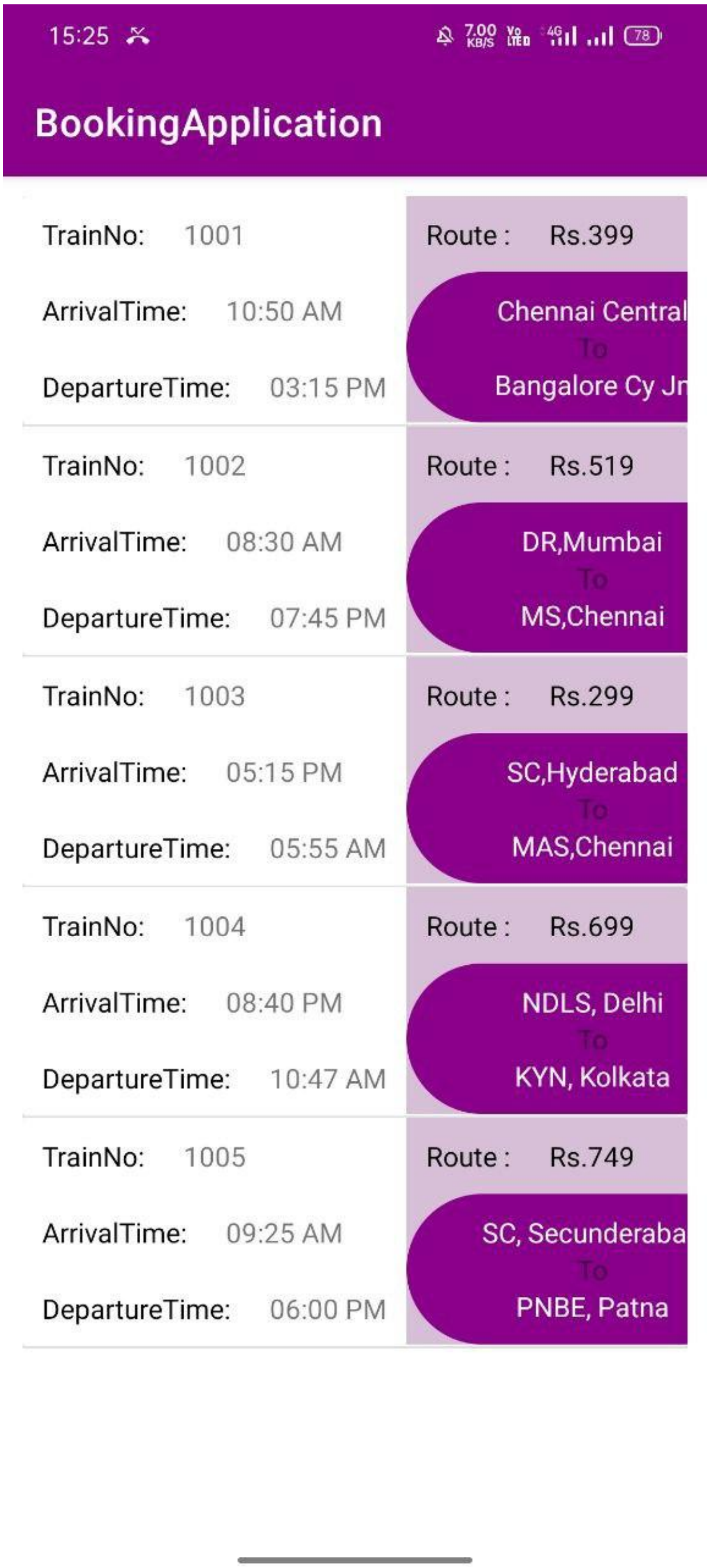


Welcome

admin123@gmail.com

Our customers and drivers must wear a protective mask during the entire journey.

CHOOSING PAGE



15:25

BookingApplication

TrainNo: 1001
ArrivalTime: 10:50 AM
DepartureTime: 03:15 PM

Route : Rs.399
Chennai Central
To
Bangalore Cy Jn

TrainNo: 1002
ArrivalTime: 08:30 AM
DepartureTime: 07:45 PM

Route : Rs.519
DR,Mumbai
To
MS,Chennai

TrainNo: 1003
ArrivalTime: 05:15 PM
DepartureTime: 05:55 AM

Route : Rs.299
SC,Hyderabad
To
MAS,Chennai

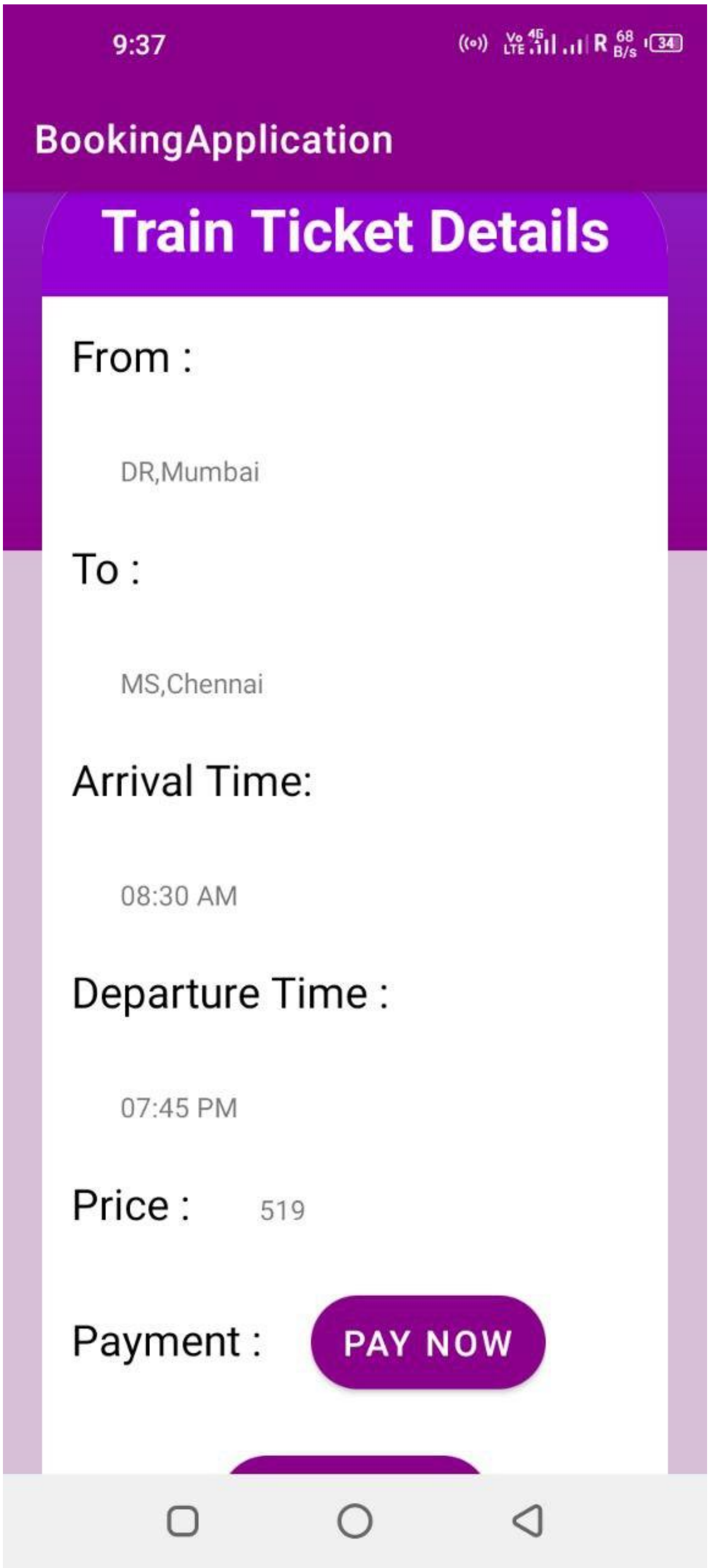
TrainNo: 1004
ArrivalTime: 08:40 PM
DepartureTime: 10:47 AM

Route : Rs.699
NDLS, Delhi
To
KYN, Kolkata

TrainNo: 1005
ArrivalTime: 09:25 AM
DepartureTime: 06:00 PM

Route : Rs.749
SC, Secunderaba
To
PNBE, Patna

CONFIRMING PAGE



9:37

BookingApplication

Train Ticket Details

From :

DR,Mumbai

To :

MS,Chennai

Arrival Time:

08:30 AM

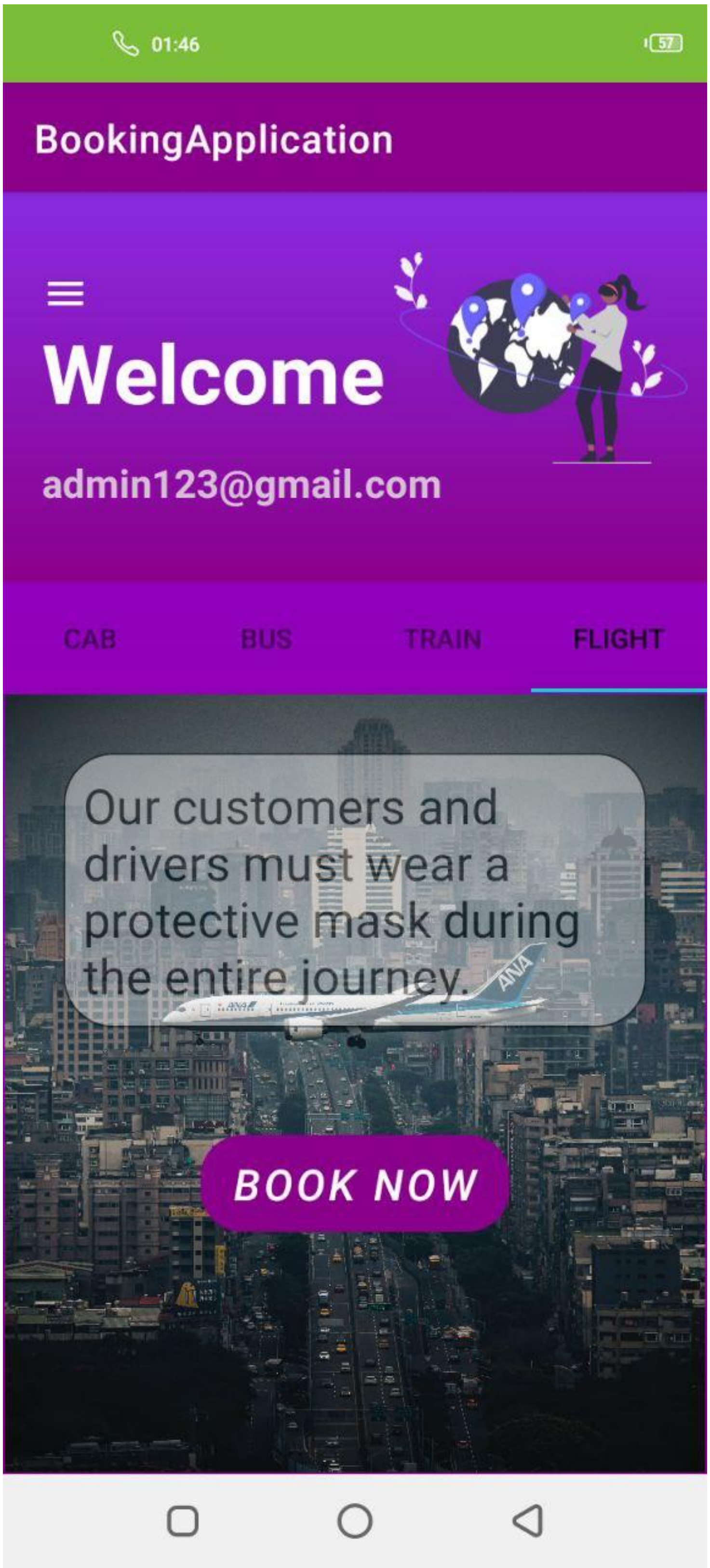
Departure Time :

07:45 PM

Price : 519

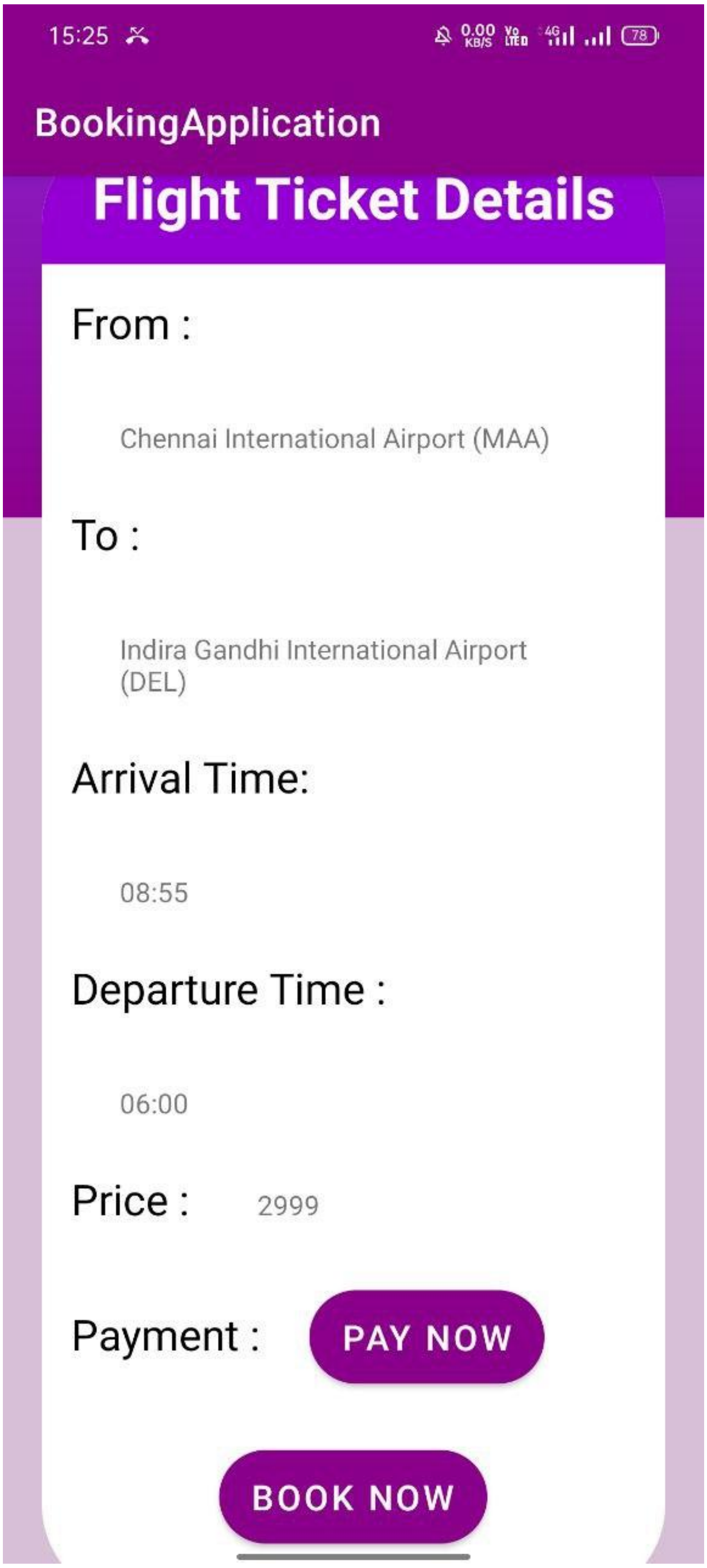
Payment :

FLIGHT HOME PAGE



Our customers and drivers must wear a protective mask during the entire journey.

CHOOSING PAGE



15:25

BookingApplication

Flight Ticket Details

From :

Chennai International Airport (MAA)

To :

Indira Gandhi International Airport (DEL)

Arrival Time:

08:55

Departure Time :

06:00

Price : 2999

CHAPTER 11
CONCLUSION

Online ticket booking system has been developed successfully. System performance is also found to be satisfactory. This is a user-friendly application. Through this application, the cost can be reduced and efficiency is increased. There are several procedures that can be selected by customers. With the help of this application customers can book tickets, can know the status of a flight, bus or trains, a Source station and destination can be chosen according to their choice, can select seats, can choose the time, and pay through the portal after reaching the station or airport. Thus online ticket booking system target internal and external audiences. Online ticket booking system is very big to maintain but it always provides excellent facilities to accomplish the goal and help to reduce a complex paperwork process through a mobile application. This can be a benefit using online ticket booking system application rather searching on several websites. With the help of online ticket booking system records are maintained and the database is updated with time to time. Through Online ticket booking system, technologies and features have been introduced.

Reference

Barton G., Lang Z., Desai, N., & Walker, J. R. (2014).U.S. Patent No. 8,887,230. Washington DC: U.S. Patent and Trademark Office.

Loo, J.,Mauri, J. L., & Ortiz, J. H. (Eds.). (2016).Mobile ad hoc networks: current status and future trends. CRC Press.

Dalmasso I., Datta S. K., Bonnet, C., & Nikaein N. (2013, July). Survey, comparison and evaluation of cross platform mobile application development tools. In Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International (pp. 323-328). IEEE