JSS Mahavidyapeetha
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
MYSURU 570 006

"RAILWAY RESERVATION SYSTEM"

Project Report submitted in partial fulfillment of curriculum prescribed for Database Management System (CS510) for the award of the degree of

Bachelor in Engineering

in

Computer Science and Engineering

By

USN	Name
01JST17CS052	Eeshaan Achar
01JST17CS097	Navneet Gajanan Hosmane
01JST17CS186	Manjari Ranjan
01JST17CS187	Ramya Krishna K

Submitted to,
Dr. Manimala S
Assistant professor
Dept. of CS&E, JSS S&TU MYSURU

(Formerly Sri Jayachamarajendra College Of Engineering)
JSS Technical Institutions, Mysuru - 570 006

COLLEGE CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Title of Course: Database Management System
Subject Code: CS510
Semester: 5
Student Names and USN:

USN	Name
01JST17CS052	Eeshaan Achar
01JST17CS097	Navneet Gajanan Hosmane
01JST17CS186	Manjari Ranjan
01JST17CS187	Ramya Krishna K

Internal Assessment Marks awarded: .....

....

Head Of Department Sign of Teacher University Seal

DECLARATION

We, the undersigned here, declare that this dissertation is entirely our own original work and that, to the best of our knowledge, it has not been presented or submitted for any other degree or examination in any other university, and that all the sources we have used or quoted have been indicated and acknowledged by complete references.

TEAM MEMBERS:

1. Name: Eeshaan Achar
   Signature: .....
2. Name: Navneet Gajanan Hosmane
   Signature: .....
3. Name: Manjari Ranjan
   Signature: .....
4. Name: Ramya Krishna K
   Signature: .....

PROJECT SUPERVISOR:

Name: Dr. Manimala S

Signature: .....

Date: .....

ABSTRACT

As we already know, the Indian Railway Management System is very vast and equally complex. One of the manjor parts of it would be the seat reservation system. Online reservation of seats has become a significant and very important service nowadays, as the demand for travelling by trains has increased. Anyone who wishes to travel comfortably by train, at their convenience, must reserve their seats beforehand.

We provide a platform for the same through this application. Users (passengers) can check the availability of trains and seats, they can book tickets and also check the status of their reservation through PNR Enquiry. They can also obtain all information regarding a specific train in one tap, by entering the train number. Admins (railway system authorities) can use this application to edit and update train information.

The entire process of creating this application - from collecting the requirements to the testing phase - has been documented in this report.

ACKNOWLEDGEMENT

Throughout the process of doing this project, we had to take some help and guidance from a few people. They deserve our utmost gratitude.

We would like to express our special thanks to our professor Dr.Manimala S, who gave us the opportunity to do this great project on the topic "Railway Reservation System", and gave us guidance and consultations throughout.

Many of our classmates helped us in thinking of all the different cases that needed to be tested for. We express our hearty gratitude to all of them.

Lastly, we would like to say thanks to our fellow teammates without whom this project would not have been successful. We thouroughly enjoyed the process of making this project happen.

Contents

1 INTRODUCTION

Database is an organised collection of data. The data is typically organised to model aspects of reality in a way that supports processes requiring information. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database itsef and the end users or application programs, ensuring that data is consistently organised and remains easily accessible.

The DBMS manages three important things:

1. The data
2. The database engine that allows data to be accessed, locked and modified
3. The database schema, which defines the database's logical structure

These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures.

The DBMS can offer both logical and physical data independence. That means, it can protect users and applications from needing to know where the data is stored or having to be concerned about changes to the physical structure of data.

1.1 Problem Definition

The main purpose of maintaining a database for Railway Reservation System is to reduce the manual errors and even efforts involved in the booking of tickets. People would no longer have to stand in long queues, waiting for their turn to make a booking. Through the database approach, all of those operations will be much quicker and easier. PNR enquiry would be a much easier and feasible task.

This database would also make it easier and convenient for the concerned authorities to keep a record of all the reservations made, and of all the trains' information. Due to automation, many loopholes that may exist in manual maintenance of records may be removed. The speed of obtaining and processing data will be much faster. Reservation of seats for passengers will therefore be a much easier and convenient task. We designed and developed a railway reservation system to do just the same.

The railway reservation system we have developed, facilitates the passengers to enquire about the trains available on the basis of the source, destination and date entered. It also allows them to make reservation of seats, enquire about the status of their reservation and even enquire a specific train's information.

It allows the concerned authorities (admins) to add, edit and update train and station information.

1.2 Aim of the project

1. Allow users (passengers) to check the availability of trains and seats in a specific train.
2. Allow users to book tickets for their journey.
3. Allow users to check the status of their reservation and arrival time through PNR enquiry.
4. Allow users to obtain all information regarding a specific train like train name, type, source and destination by just entering a train's number.
5. Allow admins to add/delete trains, view trains, add stations, update train timings and delay if any.

1.3 Objectives

- The data sets and requirements wil be collected at the beginning.
- An ER diagram and schema diagram will be constructed.
- Soon after, an architectural design draft and DFD will be devised.
- After the system design is confirmed, the system implementation will be doneand an Android application will be developed.
- After the development, testing will be carried out and necessary revisions will be made.
- At the end, the final application will be deployed.

1.4 Scope

1. Trains do not have sleeper class. And compartments are fixed i.e, 2 general, 3 semi and 3 AC compartments.

2. There are no different fares designated for different classes.

3. There is no payment gateway. Passengers would have to go to their nearest station, show their PNR as proof to make the payment and get the ticket.

4. Number of seats in each train is limited to 28.

5. There is no restriction on date for which booking can be made. So one can make a booking for any time in the future.

2 SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Functional Requirements

Functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform.

Our project deals with railways database system. It will have user login and admin login. Admin will be responsible for entering details about trains and stations. They can also update delays and arrival time of train but for only those trains which already exist. Users will be provided with features like booking seats for their journey, view their PNR status and Train Information.

2.1.1 Homepage

IRCTC Pro provides a Homepage. Homepage provides links to all major components of the application.

1. **Stimulus / Response Sequences**

   Stimulus: User clicks the Admin Login button.
   Response Sequences: User is directed to the Admin Login page.

   Stimulus: User clicks Books Tickets button.
   Response Sequences: User is directed to Book Tickets page.

   Stimulus: User clicks PNR Enquiry button.
   Response Sequences: User is directed to PNR Enquiry page.

Stimulus: User clicks Train Information button.
Response Sequences: User is directed to Train Information page.

2.1.2 Admin Login

The user is directed to the Admin Login. Admin Login provides links to all the major components used by admin.

1. **Stimulus / Response Sequences**

   Stimulus: Admin is at Login Page. User enters correct username and password.
   Response Sequences: Admin is directed to Admin page.

   Stimulus: Admin enters wrong username and password.
   Response Sequences: A notifier is shown with message saying Authentication Failed.

2.1.3 Admin Page

After successful Log-In, the admin is directed to the Admin Page.

1. **Stimulus / Response Sequences**

   Stimulus: Admin clicks the Train button.
   Response Sequences: Admin is directed to the Edit Train Info page.

   Stimulus: Admin clicks Station button.
   Response Sequences: Admin is directed to Edit Station Info page.

   Stimulus: Admin clicks Delay button.
   Response Sequences: Admin is directed to Add Delay page.

   Stimulus: Admin clicks Train Timings button.
   Response Sequences: Admin is directed to Add Train Timings page.

2.1.4 Train

Admin is directed to Edit train info page. Admin has to choose between add, delete and view trains.

1. **Stimulus / Response Sequences**

   Stimulus: Admin clicks the Add Train button.
   Response Sequences: Admin is directed to the Add Train page.

   Stimulus: Admin clicks Delete Train button.
   Response Sequences: Admin is directed to Delete Train page.

   Stimulus: Admin clicks View All Trains button.
   Response Sequences: Admin is directed to View All Trains page.

2.1.5 Station

Admin is directed to Edit Station info page. Admin has to choose between add, delete and view Stations.

1. **Stimulus / Response Sequences**

   Stimulus: Admin clicks the Add Station button.
   Response Sequences: Admin is directed to the Add Station page.

   Stimulus: Admin clicks Delete Station button.
   Response Sequences: Admin is directed to Delete Station page.

   Stimulus: Admin clicks View All Stations button.
   Response Sequences: Admin is directed to View All Stations page.

2.1.6 Delay

Admin is directed to Add Delay Page. Admin can enter details of train to be entered.

1. **Stimulus / Response Sequences**

Stimulus: Admin enters the data for train number and delay (in hours) in the respective fields and then clicks on Submit.
Response Sequences: Value Entered by the Admin will be stored in database.

Stimulus: Admin enters the repeated data in the text view and then clicks on Submit.
Response Sequences: It will show a notifier saying Data Not Inserted.

Stimulus: Admin enters a train number which does not exist in database and then clicks on Submit.
Response Sequences: It will show a notifier saying Train not found.

2.1.7 Train Timings

Admin is directed to Train Timings Page. Admin can enter details of train timings.

1. **Stimulus / Response Sequences**

   Stimulus: Admin enters the data in the text view and then clicks on Submit.
   Response Sequences: Value Entered by the Admin will be stored in database.

   Stimulus: Admin enters the repeated data in the text view and then clicks on Submit.
   Response Sequences: It will show a notifier saying Train Already Present in Database.

   Stimulus: Admin enters the data about the train or station which does not exist in database and then clicks on Submit.
   Response Sequences: It will show a notifier saying Train or Station not found.

2.1.8 Add Train

Admin is directed to Add Train Page. Admin can enter details of train .

1. **Stimulus / Response Sequences**

   Stimulus: Admin enters the data in the text view and then clicks on Submit.
   Response Sequences: Value Entered by the Admin will be stored in database.

   Stimulus: Admin enters the repeated data in the text view and then clicks on Submit.
   Response Sequences: It will show a notifier saying Train Already Present in Database.

2.1.9 Delete Train

Admin is directed to Delete Train Page.

1. **Stimulus / Response Sequences**

   Stimulus: Admin enters the Train number and clicks on delete button.
   Response Sequences: The respective data of train is deleted from database.

2.1.10 View All Trains

Admin is directed to View All Trains page. Admin can view all trains' details in this page. This page will updated everytime admin adds or delete train details.

2.1.11 Add Station

Admin is directed to Add Station Page. Admin can enter details of Station.

1. **Stimulus / Response Sequences**

   Stimulus: Admin enters the data in the text view and then clicks on Submit.
   Response Sequences: Value Entered by the Admin will be stored in database.

Stimulus: Admin enters the repeated data in the text view and then clicks on Submit.
Response Sequences: It will show a notifier saying Data Not Inserted .

2.1.12 Delete Station

Admin is directed to Delete Station Page.

1. **Stimulus / Response Sequences**

   Stimulus: Admin enters the Station number and clicks on delete button.
   Response Sequences: The respective data of Station is deleted from database.

2.1.13 View All Stations

Admin is directed to View All Stations page. Admin can view all stations' details in this page. This page will be updated everytime admin adds or delete station details.

2.1.14 PNR Enquiry

PNR Enquiry is for users, through which passengers can check the status of their reservation.To check status they need to enter a PNR number provided to them while making reservations.

1. **Stimulus / Response Sequences**

   Stimulus: User has to enter the PNR and press Submit button.
   Response Sequences: It will show details like train name, source, destination, arrival time and name of the person by whom ticket was booked. It shows whether the train will be on time or it is delayed.

2.1.15 Book Tickets

Book Tickets Page is accessed by Passengers. They can book their journey for a specific date in a particular train and compartment.

1. **Stimulus / Response Sequences**

   Stimulus: User has to enter source, destination and date. Then user has to press search trains.
   Response Sequences: User is directed to a page, where lists of trains for that specific date is displayed.

   Stimulus: User chooses the train.
   Response Sequences: User is directed to Compartment Page. They will be shown all the available compartments for that specific train.

   Stimulus: User chooses the compartments.
   Response Sequences: Seat layout page will be shown to users.

   Stimulus: User selects the seat and click on book button.
   Response Sequences: User has to enter the his/her name and phone number. Ticket price will be shown.

   Stimulus: User clicks on confirm button.
   Response Sequences: Ticket Confirmation message will be shown to user and PNR will be provided.

2.1.16 Train Information

User is directed to Train Information Page. Here, User can see train details like source, destination, train name and timings of a specific train.

1. **Stimulus / Response Sequences**

   Stimulus: User enteres the train number and clicks on get info.
   Response Sequences: A dialog box appears with all the above mentioned train details.

2.2 Non Functional Requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

- **Security**-All the users will be provided with logins where they can login with their google account. All the records of reservations will be secure. Only Admins will have access to activities that are restricted to authorised users only.
- **Reliability**-The reliability of the overall project depends on the separate components. The main pillar of the system is the backup of the database which is continuously maintained and updated to reflect the most changes.
- **Maintainability**-A Relational database is used for maintaining the database and the application server takes care of the site, a re-initialisation of the project will be done.
- **Availability**-The system should be available at all times, meaning the user can access the application anytime. A customer friendly system which is easily accessible to the people around the world 24 hours.
- **Supportability**-The code and supporting modules of the system will be well documented and easy to understand.

2.2.1 Performance Requirements

Performance requirements define how well the system performs certain functions under specific conditions.
In order to assess the performance of a system the following must be clearly specified:

- **User Satisfaction**-The system is such that it stands up to the user expectations.
- **Response Time**- The response of all operation is good. This has been made possible by careful programming.
- **Portable**-The software is not architecture specific. It can be easily transferred to other platforms if needed.
- **User friendliness**-The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.
- **Platform**-The system is an android application that runs on all devices of OS version 4.3 and above.

3 SYSTEM DESIGN AND IMPLEMENTATION

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

Elements of a System

- **Architecture** - This is the conceptual model that defines the structure, behavior and more views of a system. We can use flowcharts to represent and illustrate the architecture.
- **Modules** - This are components that handle one specific tasks in a system. A combination of the modules make up the system.
- **Interfaces** - This is the shared boundary across which the components of a the system exchange information and relate.
- **Data** - This the management of the information and data flow.

3.1 ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

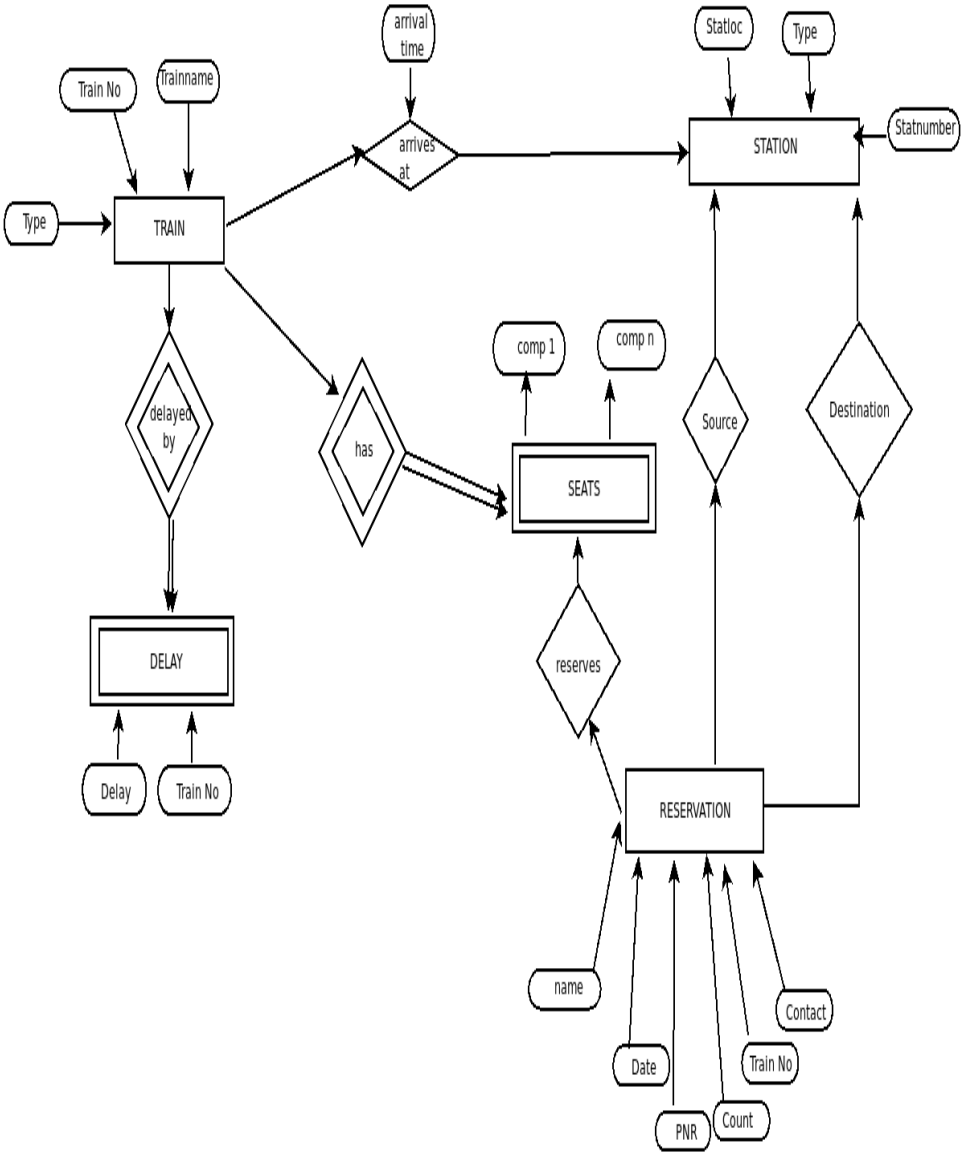

Figure 1: ER DIAGRAM

3.1.1 The Identified Entities

The subsection below lists and describes the identified entities of the design model.

1. **TRAIN**
   This entity member contains all details of all trains. It consist of train number,train name and the type of details.

2. **DELAY**
   This entity contains all the information about delay. The details are train number and delay in time.

3. **RESERVATION**
   This entity contains all the information about reservation made by users. It will have name of the person, contact number, Date of the journey, count whuch will be number of people travelling on one ticket, PNR and the train number.

4. **STATION**
   This entity contains all the information about stations. It will contain details about station location, station number and station name.

5. **SEATS** This entity contains information about seats. It will contain details about compartments.

3.1.2 The Identified Relationships

1. **Arrives At**
   This relation contains information about arrival time of the train at a specific station. It will contain values of station number, train number and arrival time.

2. **Delayed By**
   Delayed by is a relationship joining two entities namely train and delay. It maps delay to the respective trains.

3. **Has**
   Has is a relationship joining two entities namely train and seats. It maps compartments to the respective trains.

4. **Reserves**
   Reserves is a relationship joining two entities namely seats and reservation. It contains information about seat number.

5. **Source**
   Source is a relationship joining two entities namely reservation and station.

6. **Destination**
   Destination is a relationship joining two entities namely reservation and station.

3.2 Schema Diagram

In database terms, a schema is the organisation and structure of a database. A schema contains schema objects, which could be tables, columns, data types, views, stored procedures, relationships, primary keys, foreign keys, etc. A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

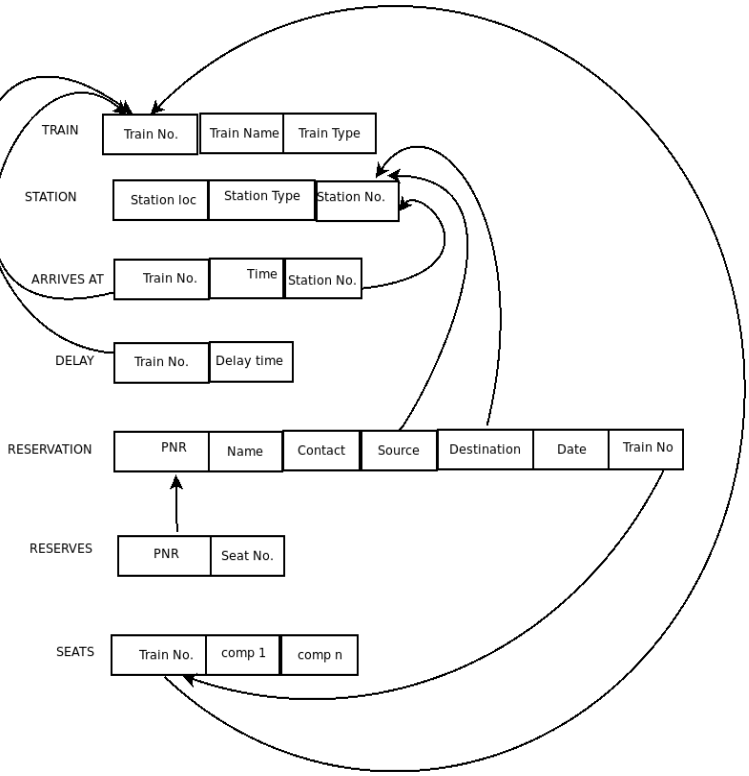

Figure 2: ER DIAGRAM

3.3 Data Flow Diagrams

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.
Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see mainly 3 levels in data flow diagram, which are: 0-level DFD and 1-level DFD.

3.3.1 0-level DFD

It is also known as context diagram.Its designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represent the entire system as single bubble with input and output data indicated by incoming/outgoing arrows.

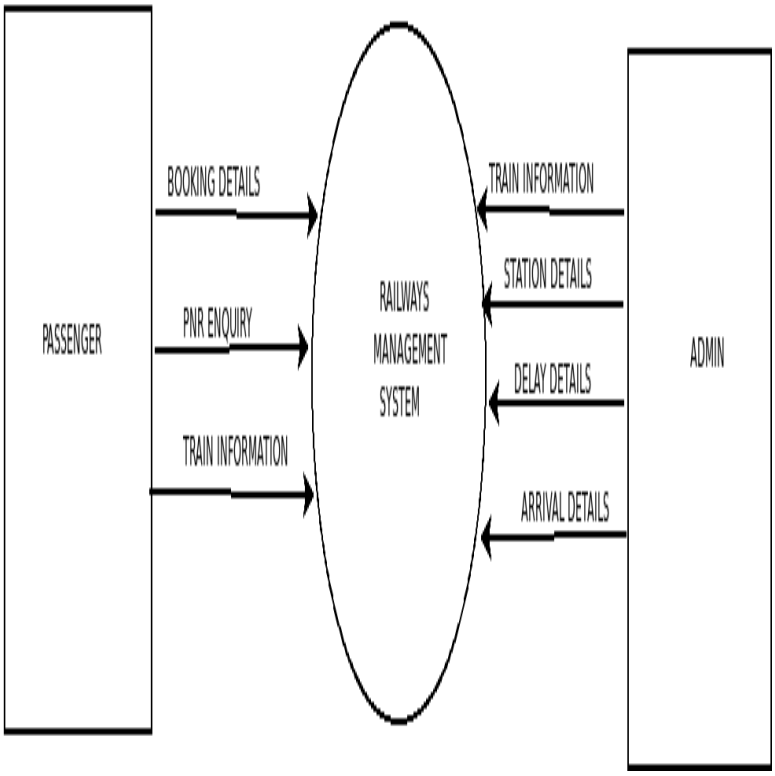

Figure 3: 0 - LEVEL

3.3.2 1-level DFD

In 1-level DFD, context diagram is decomposed into multiple bubbles/processes.in this level we highlight the main functions of the system and breakdown the high level process of 0-level DFD into subprocesses.

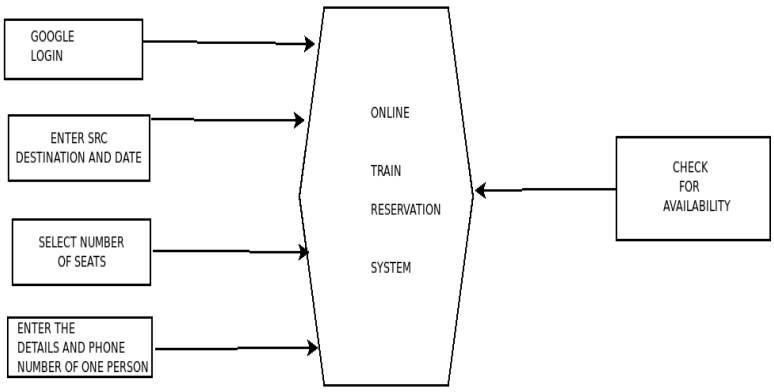

Figure 4: 1 - LEVEL(Reservation)

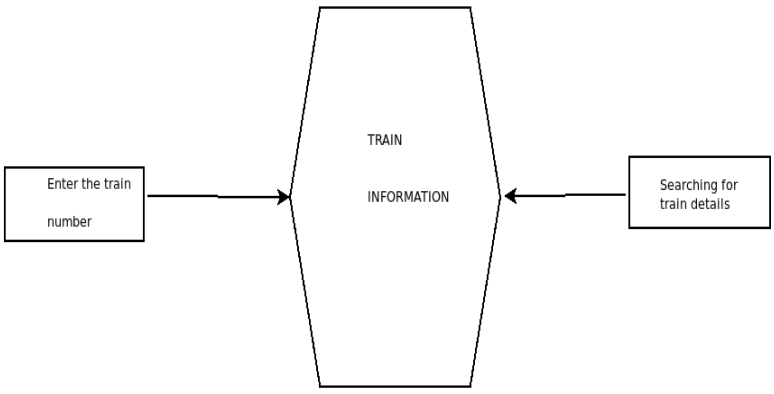

Figure 5: 1 - LEVEL(Train Information)

3.4 Modulewise Description

This section describes various modules in the proposed system, and the interaction among them.
The modules are listed based on the order in which they appear to the user. Throughout this section, interface refers to the modules which realise the user interface. Each interface is backed with another module which performs the functionalities.

3.4.1 Module 1: Admin Login

This is the first button in the homepage. It allows admin (concerned railway authorities) to login to the app and perform all the authorised users only activities.
Dependent modules: All the other modules.
Related Interface: Authentication interface.
Functionalities:

1. Allows admin to enter username and password, and login to the app.
2. In case of authentication failure, provides user with apt error messages.

3.4.2 Module 2: Edit train information

This is the first option that appears after admin login. It allows to add, delete and view trains.
Dependent modules: All other modules.
Related interface: Buttons for add train, delete train and view all trains.
Functionalities:

1. If Add Train is chosen:

   Fields for entering train number, train name and type will appear. After entering all the above fields, user can tap on Add Train to submit it. Confirmation message will be displayed.

2. If Delete Train is chosen:

   A single field for entering train number appears, and a delete button below it. Tapping on delete will remove that train from the database. Confirmation message will be displayed.

3. If View All Trains is chosen:

   A list of all the existing trains in the system appears in the ascending order of train numbers, along with train name and type.

3.4.3 Module 3: Edit station information

It allows admin to add, delete and view all stations. It is the second button after admin logs in.
Dependent modules: All other modules.
Related interface: Buttons for Add station, Delete station and View all stations.
Functionalities:

1. If Add station is chosen:

   Fields are provided to enter station location, station type and station number. After entering all the above values, user can tap on Add Station and confirmation message will be displayed.

2. If Delete Station is chosen:

   A single field is provided to enter station number. After entering the value and tapping on Delete button, confirmation message will be displayed.

3. If View All Stations is chosen:

   The list of all the stations will be displayed, along with the station number, name and type.

3.4.4 Module 4: Enter Train Timings

It is the third button after admin login. It allows the admin to enter train timings.
Dependent modules: All other modules.
Related Interface: Fields provided for entering Train number, station number and arrival time.
Functionalities:

1. The corresponding values for the above fields will be entered. One can add timings to a train that has previously been added to the database.

2. Confirmation message is displayed if timings are added successfully.

3. Error message is displayed if the entered train or station number does not exist.

3.4.5 Module 5: Add Delay

It is the last button that appears for the admin. It allows admin to add or update train delays if any.
Dependent modules: All other modules.
Related Interface: Fields to enter Train number and the delay time in hours.
Functionalities:

1. The values for the above mentioned fields will be entered. One can add delay to a train that has previously been added to the database.

2. Confirmation message is displayed if delay is added successfully.

3. Error message is displayed if the entered train number does not exist.

3.4.6 Module 6: Book Tickets

It is the second button in the homepage. It allows users (passengers) to book tickets for their desired trains and dates.
Dependent modules: PNR Enquiry.
Related Interface: Fields to enter Source, Destination and Date of journey.
Functionalities:

1. Passenger enters source, destination and Date of their journey and taps on search trains.

2. The list of available trains for that source, destination and date will be displayed. User can select the train of their choice. If no trains are available, appropriate message will be displayed.

3. After selection of train, user will be directed to a screen that consists of all the different compartments in that train. User can choose a compartment of their desire.

4. That will direct you to a screen that shows the seat layout of that compartment for that train. Here, the user can select the seats of their choice.

(a) The unavailable seats (seats already booked by someone else) will be shown in grey. These seats cannot be booked. If user attempts to do it, appropriate message will be displayed.

(b) The available seats will be shown in green. These seats can be booked by the user.

(c) After selection of seats, they turn blue.

After selection, user will tap on book.

5. This directs the user to another screen that displays the total fare of booking. It will also ask user to enter their name and phone number.

6. After entering the above values, user can tap on confirm to confirm booking.

7. The above activity will direct user to a screen that shows the booking confirmation message along with the PNR.

3.4.7 Module 7: PNR Enquiry

This is the third button in the homepage. It allows passengers to check the status of their reservation through the PNR they are assigned with.
Dependent Modules: None
Related Interface: A single field to enter PNR, along with a button that says 'Get Status'.
Functionalities:

1. User enters the PNR they have been assigned with and taps on Get Status button.

2. Appropriate message will be shown according to the status of the reservation.

   (a) Says Train is Cancelled if the train gets cancelled.

   (b) Says Train on time if the train is not cancelled and there is no delay. It also displays all details regarding the reservation such as Train number, Train name, journey date, boarding station code, arrival time at boarding station, alighting station code, arrival time at alighting station, and the name of the passenger who made that reservation.

(c) Says Train is delayed if the train is not cancelled but there is a delay. It shows by how much time the train has been delayed. It also displays all details regarding the reservation such as Train number, Train name, journey date, boarding station code, updated arrival time at boarding station, alighting station code, updated arrival time at alighting station, and the name of the passenger who made that reservation.

3. If invalid PNR is entered, error message will be displayed.

3.4.8 Module 8: View Train Information

This is the fourth button on the homescreen. It allows users to check all information regarding a specific train.
Dependent Modules: None.
Related Interface: A single field that asks for the train number to be entered, whose information is to be obtained.
Functionalities:

1. The train number for which the information is to be obtained, is entered in the above mentioned field. User then taps on Get Info button.

2. The above activity shows all related information to that specific train such as train name, train type, source and destination of that train's route.

3. If invalid train number is entered, appropriate error message is displayed.

3.5 SQL Queries

Create Table

db.execSQL("CREATE TABLE TRAINS (TRAIN_ NO TEXT PRIMARY KEY, TRAIN_ NAME TEXT, TRAIN_ TYPE TEXT);");

db.execSQL("CREATE TABLE STATIONS (STATION_ NO TEXT PRIMARY KEY, STATION_ LOCATION TEXT, STATION_ TYPE TEXT);");

db.execSQL("CREATE TABLE DELAY (TRAIN_ NO TEXT PRIMARY KEY, DELAY TEXT, FOREIGN KEY (TRAIN_ NO) REFERENCES TRAINS(TRAIN_ NO) ON DELETE CASCADE);");

db.execSQL("CREATE TABLE ARRIVES_ AT (TRAIN_ NO TEXT, STATION_ NO TEXT, TIME TEXT, PRIMARY KEY (TRAIN_ NO, STATION_ NO), FOREIGN KEY (TRAIN_ NO) REFERENCES TRAINS (TRAIN_ NO) ON DELETE CASCADE, FOREIGN KEY (STATION_ NO) REFERENCES STATIONS (STATION_ NO) ON DELETE CASCADE);");

db.execSQL("CREATE TABLE RESERVATIONS (PNR INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT, CONTACT TEXT, SOURCE TEXT, DESTINATION TEXT, DATE TEXT, TRAIN_ NO TEXT, FOREIGN KEY (SOURCE) REFERENCES STATIONS (STATION_ NO) ON DELETE SET NULL, FOREIGN KEY (DESTINATION) REFERENCES STATIONS (STATION_ NO) ON DELETE SET NULL);");

db.execSQL("CREATE TABLE RESERVED_ SEATS (PNR INTEGER, SEAT_ NO TEXT PRIMARY KEY, FOREIGN KEY (PNR) REFERENCES RESERVATIONS (PNR) ON DELETE CASCADE);");

db.execSQL("CREATE TABLE SEATS (TRAIN_ NO TEXT PRIMARY KEY, COACH1 TEXT, COACH2 TEXT, COACH3 TEXT, COACH4 TEXT, COACH5 TEXT, COACH6 TEXT, COACH7 TEXT, COACH8 TEXT, COACH9 TEXT, COACH10 TEXT, FOREIGN KEY (TRAIN_ NO) REFERENCES TRAINS (TRAIN_ NO) ON DELETE CASCADE);");

db.execSQL("CREATE TRIGGER ADD_ SEATS AFTER INSERT ON TRAINS BEGIN INSERT INTO SEATS VALUES (NEW.TRAIN_ NO, 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu', 'uuuuuuuuuuuuuuuuuuuuuuuuuuuuuu'); END");

Insert Values

db.execSQL("INSERT INTO TRAINS VALUES ('100', 'Mangalore Mysore Express', 'Express'), ('101', 'Mangalore Mysore Express', 'Express'), ('200', 'Tippu Express', 'Express'), ('201', 'Tippu Express', 'Express'), ('300', 'MYS SBC Passenger', 'Passenger'), ('301', 'MYS SBC Passenger', 'Passenger');");

db.execSQL("INSERT INTO STATIONS VALUES ('MYS01', 'Mysore', 'Junction'), ('MLR01', 'Mangalore', 'Central'), ('BLR01', 'Bangalore', 'Junction'), ('BLR02', 'Bangalore', 'Central');");

db.execSQL("INSERT INTO ARRIVES_ AT VALUES ('100', 'MYS01', '0700'), ('100', 'MLR01', '1400'), ('101', 'MYS01', '1400'), ('101', 'MLR01', '0500'), ('200', 'MYS01', '1400'), ('200', 'BLR02','1700'), ('201', 'MYS01', '1700'), ('201', 'BLR02','1400'), ('300', 'MYS01', '1400'), ('300', 'BLR01', '1800'), ('301', 'MYS01', '1800'), ('301', 'BLR01', '1400');");

db.execSQL("INSERT INTO RESERVATIONS VALUES (256, 'You found and Easter Egg', '123', 'MYS01', 'MLR01', '24/12/19','100'), (null, 'Some random dude', '1234', 'MYS01', 'MLR01', '24/12/19','100');");

db.execSQL("INSERT INTO RESERVED_ SEATS VALUES (256, '100COACH11'), (256, '100COACH12'), (257, '100COACH110');");

db.execSQL("UPDATE SEATS SET COACH1 = 'bbuuuuuuubuuuuuuuuuuuuuuuuuu' WHERE TRAIN_ NO = '100'");

3.6 Implementation Details

Implementation is the stage of the project when the theoretical design is turned out into a working system. Therefore it can be considered to be the most critical stage in achieving a successful new system and in giving the user the assurance that the new system will work and be effective.

3.6.1 Architectural Model

The model chosen for the product is a 3-tier architecture design.The 3-tier architecture separates its tiers from each other based on how the different users use the data present in the database.The tiers are Database tier, Application tier and User tier.

3.6.2 Front End

This is the client-side development, or the user view of the product. End-users operate on this tier and they know nothing about any existence of the database beyond this layer. At this layer, multiple views of the database is provided by the application. All views are generated by applications that reside in the application tier or the middleware. It is implemented using Android Studio, and based on the code written, a user can see and interact with the components directly.

3.6.3 Back End

This is the most important component which helps in storing the entire data of the users. A database management system is used for easy storing and accessing of data from the database. The product uses SQLite as the database management system, which was built in Android Studio.

3.6.4 Connectivity

This layer consists the core of the proposed system. It provides functionalities of I/O to DBMS. It also performs the computation needed to summarize the data. It is used for retrieving the data from the database and displaying on to the application for the user to access and update. Similarly the data entered by the user can be entered into the database using this layer. Therefore, at this tier the application server and the programs that access the database reside. For a user, this application tier

presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application. At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database. Sqlite comes in built with Android Studio. Therfore, we don't need a special layer for connectivity between backend and frontend.

3.7 User Interface

Figure 6: Homepage that appears as soon as the app is opened

Figure 7: After Correct Login of admin, the admin page opens

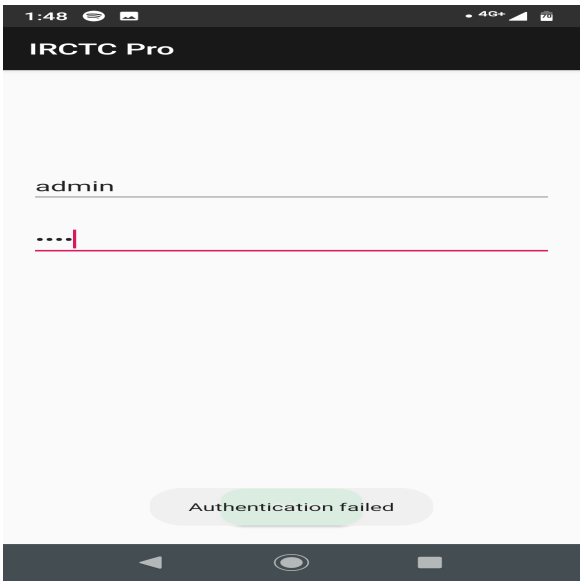

Figure 8: After Incorrect Login, error message is shown

Figure 9: Edit Train Page

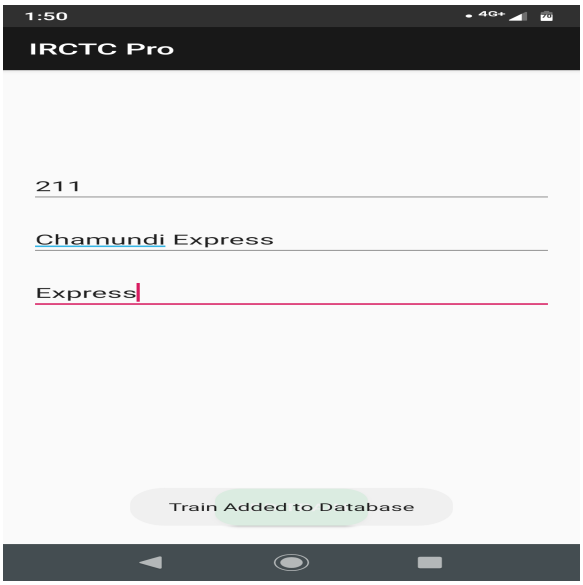

Figure 10: Add Train

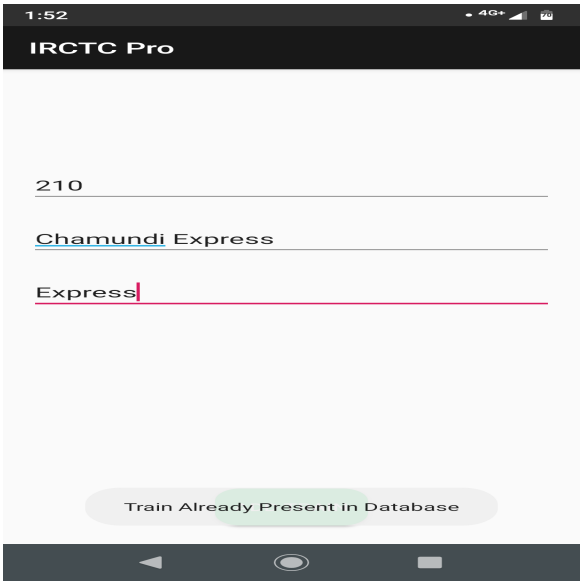

Figure 11: Add Existing Train Details- apt message will be shown

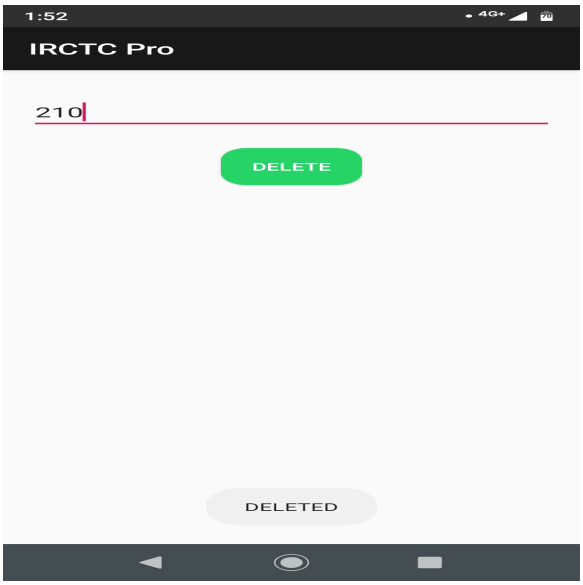

Figure 12: Delete Train Details

Figure 13: Viewing Train Details

Figure 14: Station Page

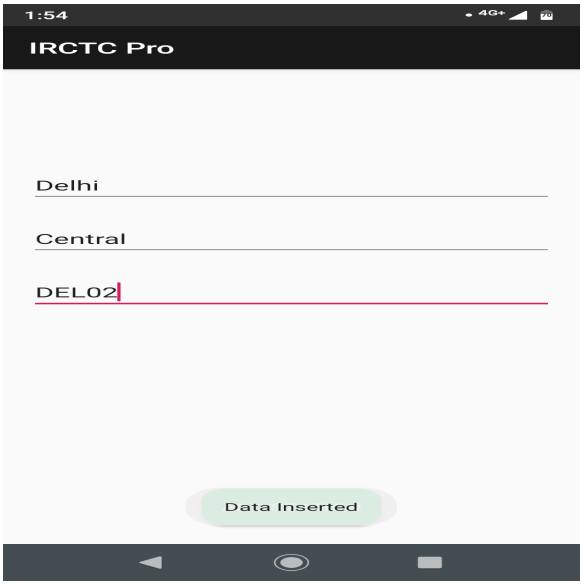

Figure 15: Add Station

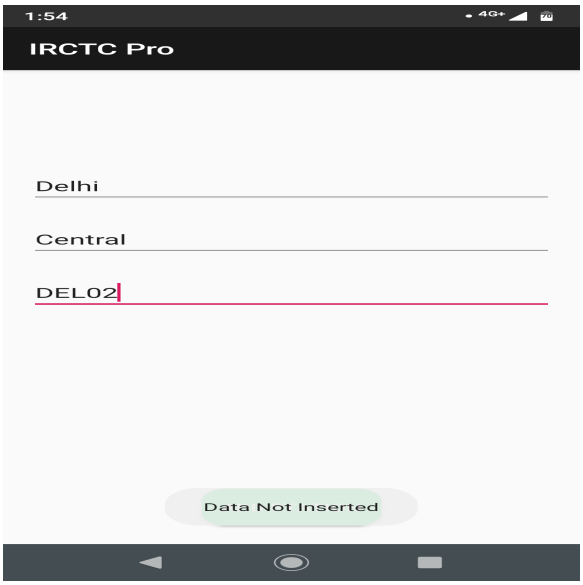

Figure 16: Add Existing Station Details- apt message will be shown

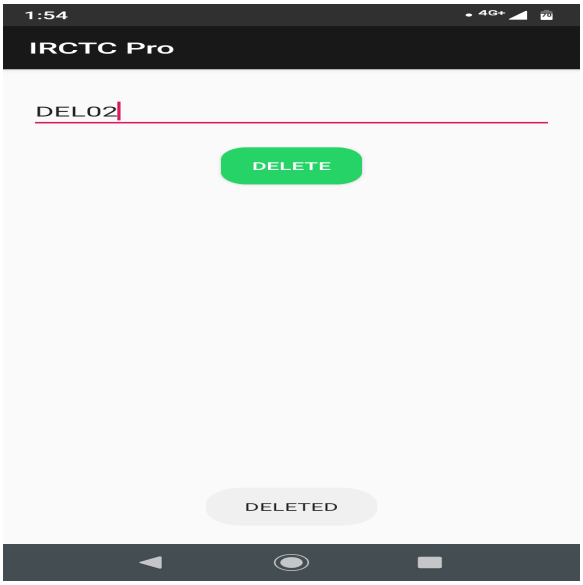

Figure 17: Delete Station Details

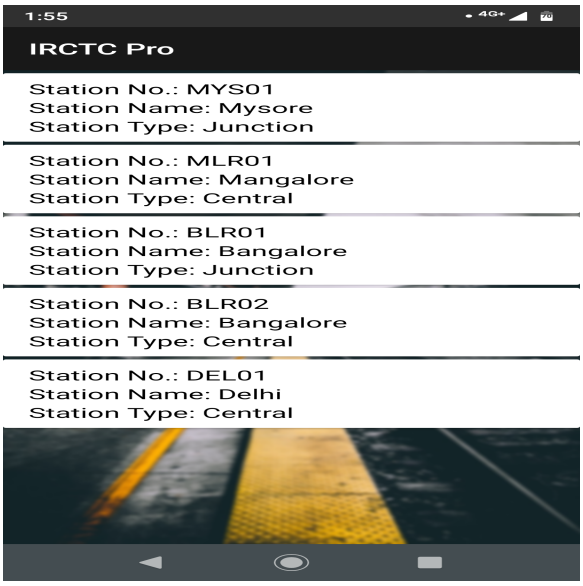

Figure 18: Viewing Station Details

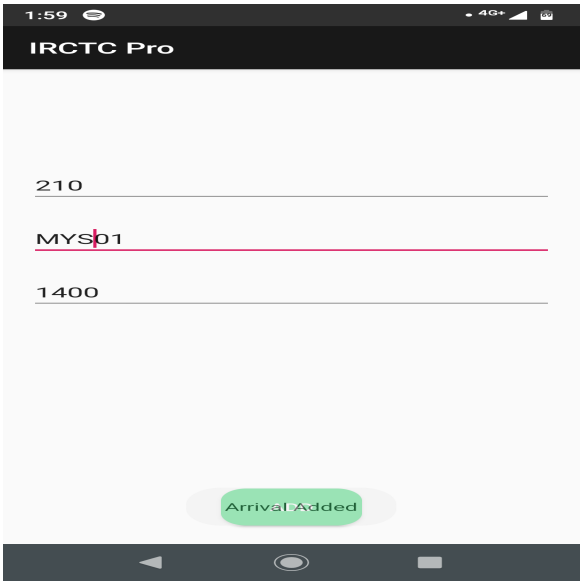

Figure 19: Adding Train Timings

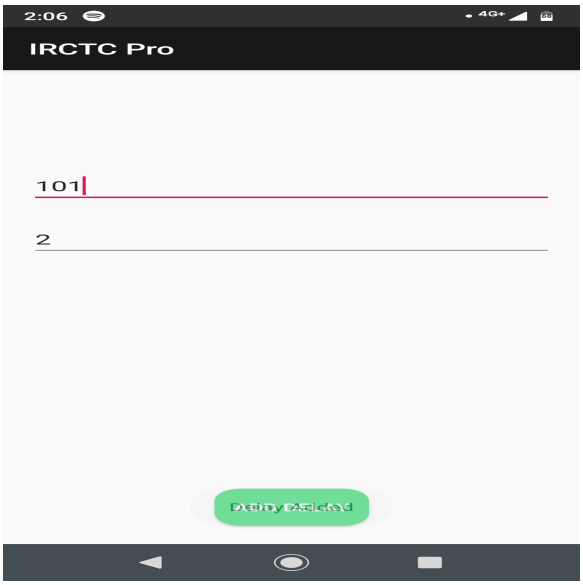

Figure 20: Adding Delay

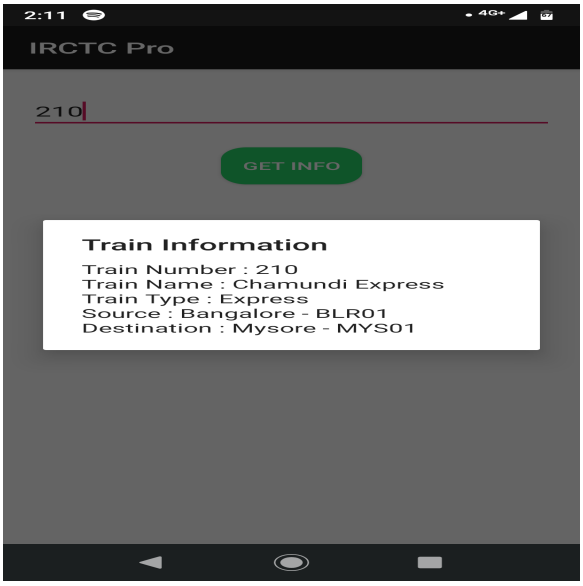

Figure 21: Train information page

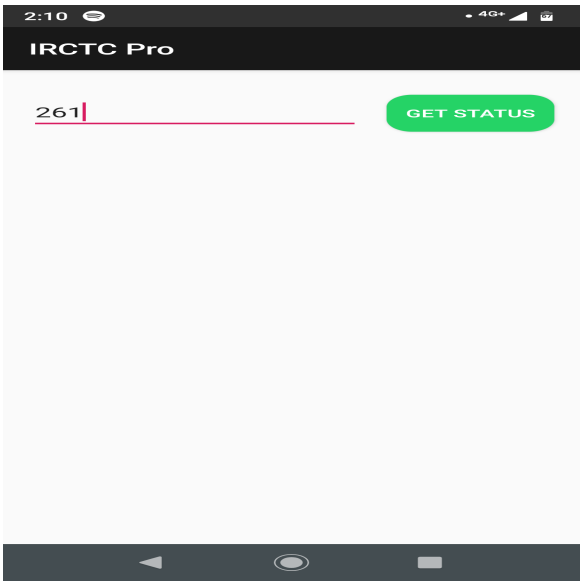

Figure 22: Entering PNR Number for enquiry

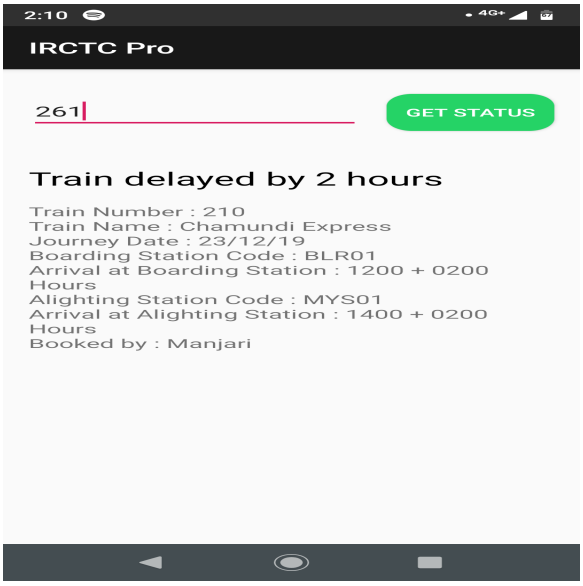

Figure 23: PNR Details when the train is delayed

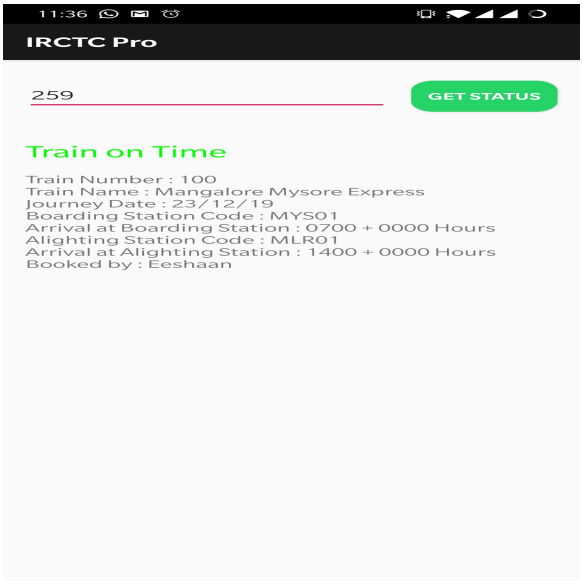

Figure 24: PNR details when train is on time

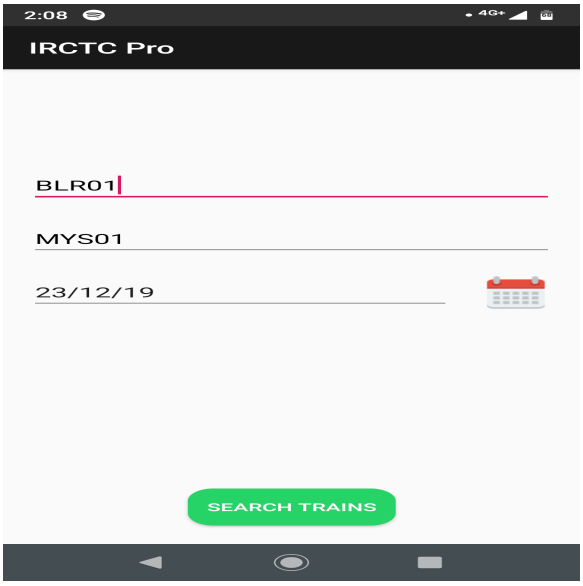

Figure 25: Booking Step 1: Entering source, destination and date

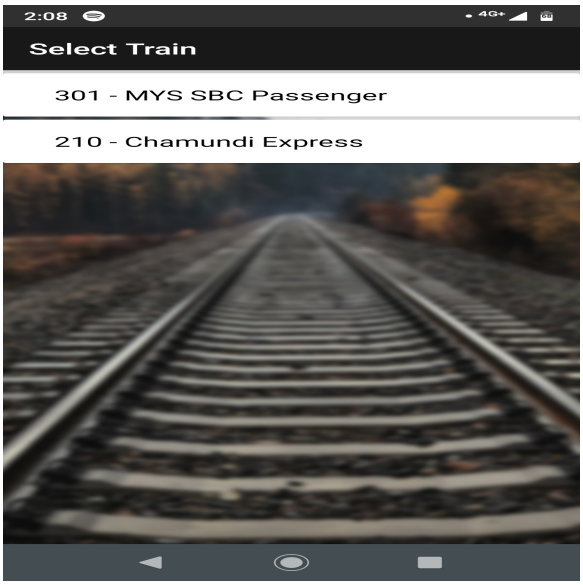

Figure 26: Booking Step 2:Choosing among the available trains

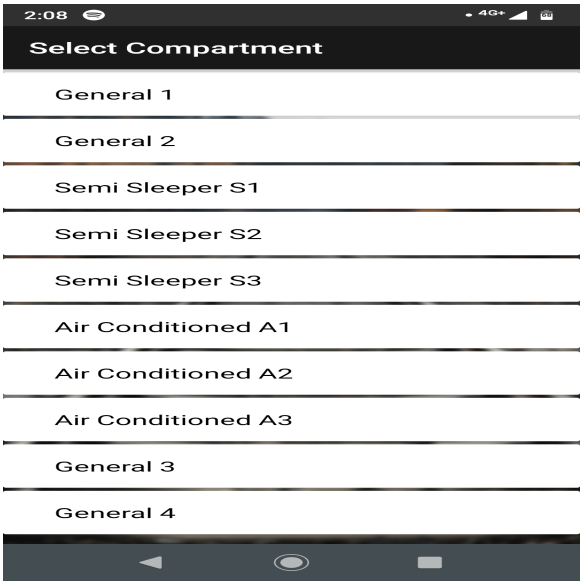

Figure 27: Booking Step 3: Choosing among the available compartments

Figure 28: Seat layout

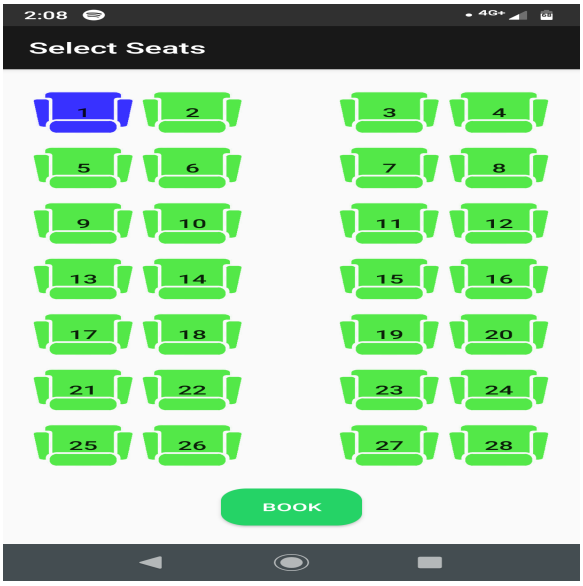

Figure 29: Selection of seats (marked in blue)

Figure 30: Already booked seats that are unavailable (shown in grey)

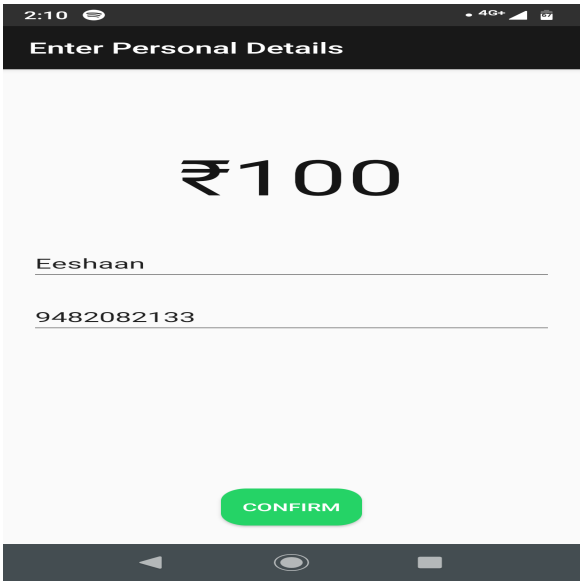

Figure 31: Displaying total fare and entering name and contact details

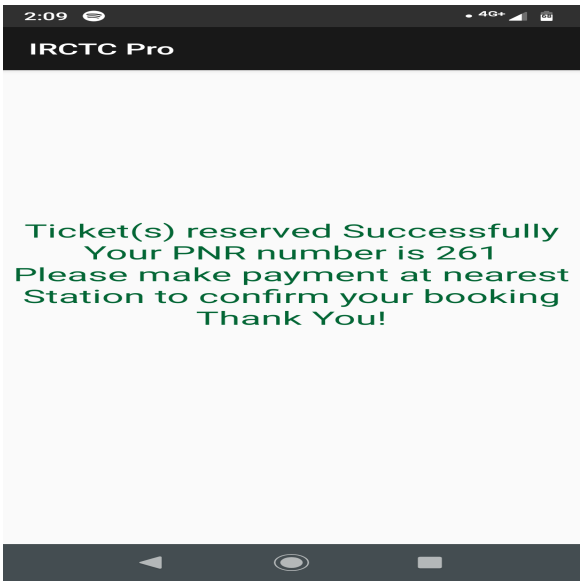

Figure 32: Booking confirmation and PNR is provided

4 SYSTEM TESTING

Testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.
We will consider two types of testing:

- Whitebox Testing
  White box testing is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.

- Blackbox Testing
  Black box testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester.

4.1 Admin Login

Test id	**Feature**	**Input**	**Expected output**	**Actual Output**	**Remarks**
TC 1	Correct Login	Admin admin	Admin should be directed to admin page	Admin was directed to next page	Passed
TC 2	Incorrect Login	admin add	A Notifier should be displayed telling Authentication Failed	A notifier was displayed telling Authentication Failed	Passed

4.2 Admin Page

Test id	Feature	Input	Expected output	Actual Output	Remarks
TC 3	Train Page	Click on train button	Admin should be directed to train page	Admin was directed to train page	Passed
TC 4	Delay Page	Click on Delay button	Admin should be directed to delay page	Admin was directed to delay page	Passed
TC 5	Station Page	Click on Station button	Admin should be directed to station page	Admin was directed to station page	Passed
TC 6	Train Timings Page	Click on Train Timings button	Admin should be directed to train timings page	Admin was directed to train timings page	Passed

4.3 Train

Test id	Feature	Input	Expected output	Actual Output	Remarks
TC 7	Add Train Page	Click on Add Train button	Admin should be directed to add train page	Admin was directed to add train page	Passed
TC 8	Delete Train Page	Click on Delete Train button	Admin should be directed to delete train page	Admin was directed to delete train page	Passed
TC 9	View all Trains Page	Click on View all Trains button	Admin should be directed to View all Trains page	Admin was directed to view all train page	Passed

4.4 Station

Test id	Feature	Input	Expected output	Actual Output	Remarks
TC 10	Add Station Page	Click on Add Station button	Admin should be directed to add station page	Admin was directed to add station page	Passed
TC 11	Delete Station Page	Click on Station button	Admin should be directed to delete station page	Admin was directed to delete station page	Passed
TC 12	View all Station Page	Click on View all Station button	Admin should be directed to View all stations page	Admin was directed to view all stations page	Passed

4.5 Delay

Test id	Feature	Input	Expected output	Actual Output	Remarks
TC 13	Delay Details	Admin enters all the required data about delay	When click on submit button, data should be entered	Data was entered and notifier was shown	Passed
TC 14	Delay Details	Admin enters repeated data about delay	When click on submit button, data shouldnot be entered	Data wasnot entered and notifier was shown	Passed
TC 15	Delay Details	Admin enters data of Train which doesnot exist	When click on submit button, data shouldnot be entered	Data wasnot entered and notifier was shown	Passed

4.6 Train Timings

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 16	Train Timings Details	Admin enters all the required data about train timings	When click on submit button, data should be entered	Data was entered and notifier was shown	Passed
TC 17	Train Timings Details	Admin enters repeated data about train timings	When click on submit button, data shouldnot be entered	Data wasnot entered and notifier was shown	Passed
TC 18	Train Timings Details	Admin enters data of Train about train timings which doesnot exist	When click on submit button, data shouldnot be entered	Data wasnot entered and notifier was shown	Passed

4.7 Add Train

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 19	Add Train Details	Admin enters all the required data about train	When click on submit button,data should be inserted in database and notifier should shown for the same	Data was entered and notifier was shown	Passed

Test id	**Feature**	**Input**	**Expected output**	**Actual Output**	**remarks**
TC 20	Add Train Details	Admin enters repeated data about train	When click on submit button,data should-not be inserted in database and notifier should shown for the same	Data wasnot entered and notifier was shown	Passed

4.8 Delete Train

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 21	Delete Train	Admin has to enter train number	data related to that train should be deleted	Data was deleted from database	Passed

4.9 Delete Station

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 22	Delete Station	Admin has to enter station number	data related to that station should be deleted	Data was deleted from database	Passed

4.10 Add Station

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 23	Add Station Details	Admin enters all the required data about Station	When click on submit button,data should be inserted in database and notifier should shown for the same	Data was entered and notifier was shown	Passed

Test id	**Feature**	**Input**	**Expected output**	**Actual Output**	**remarks**
TC 24	Add Station Details	Admin enters repeated data about Station	When click on submit button,data shouldnot be inserted in database and notifier should shown for the same	Data wasnot entered and notifier was shown	Passed

4.11 PNR Enquiry

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 25	PNR Enquiry and train is on time	Enter the correct PNR	Booking details of the respective PNR should be displayed with 'Train on time' message	Booking Details was displayed with apt message	Passed.
TC 26	PNR Enquiry and train is delayed	Enter correct PNR	Booking details shown along with 'Train Delayed' message and updated arrival time	Booking details were displayed with apt message for delay and updated arrival time	Passed
TC 27	PNR Enquiry	Enter the incorrect PNR	It should show invalid PNR	It showed invalid PNR	Passed.

4.12 Train Information

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 28	Train Information	Enter train number	information of respective train should be displayed	Information about train was displayed	Passed.

4.13 Book Tickets

Test id	Feature	Input	Expected output	Actual Output	remarks
TC 29	Booking Tickets	User has to enter source,destination and date. after this,he/she has to choose no.of seats and confirm booking	Booking Confirmation message sgould be displayed to user. The reserves seats should be available for further booking	Confir mation Message was displayed and seats were not available for further booking	Passed.

5 CONCLUSION AND FUTURE WORK

Our application IRCTC Pro was successfully designed and developed. It meets all the critical requirements specified in the requirements section. Android Studio IDE was used to develop the software. The backend was implemented using SQLite that is a built-in in Android Studio. All the specified features are implemented and tested for. Revisions were made wherever necessary and the Android application was deployed.
In the future, we would like to expand the project to fulfil more features that may be as follows:

1. A website could be developed so that it would be easier for admins to access it over mobile applications.

2. The number of seats and compartments can be increased to meet real life requirements, and the fares can be assigned to each class respectively.

3. A payment gateway can be included so that people would not have to go to the station to pay for and get their tickets.

4. A restriction can be employed deciding on how many days further into the calendar a person can make their reservation.

REFERENCE

For a better understanding of certain concepts and the existing softwares for railway reservation system, these links were used:

1. https://www.irctc.co.in
2. http://www.indianrail.gov.in/enquiry/SEAT/SeatAvailability.html?locale=en
3. https://www.guru99.com/functional-vs-non-functional-requirements.html