DASC5300 – FOUNDATION OF COMPUTING

Programming Assignment 3

RAILWAY RESERVATION

Team Members:

Hari Krishna Aravakantha
UTA ID: 1002122487
Email ID: hxa2489@mavs.uta.edu

Abhishek Saraf
UTA ID: 1002125030
Email ID: axs5030@mavs.uta.edu

HONOR CODE:

I pledge, on my honor, to uphold UT Arlington’s tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Signatures:

HARI KRISHNA ARAVAKANTHA	ABHISHEK SARAF
Teammate 1	**Teammate 2**

TASK:

To Create a simple and friendly GUI interface that would be able to perform the following tasks using either of JAVA programming using JDBC, or C/C++/C# programming with ODBC/Oracle or Python programming or PHP/MySQL or other programming languages to develop a GUI interface. The user will have to type or select the query's input parameters and post the question to your program. The program needs to return all result's rows.

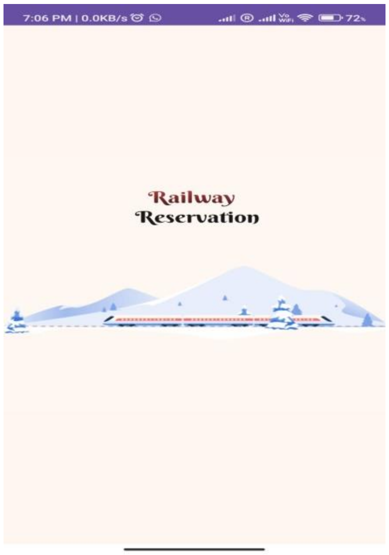
We have used Android studio using kotlin programming language for GUI development and sqlite for backend programming.

App Icon:

We have created an Android Mobile application named as "**Railway Reservation**", Includes 6 features which executes 6 queries.

The title of the app is shown when it is opened, and a train is moving in the background.

APP ICON

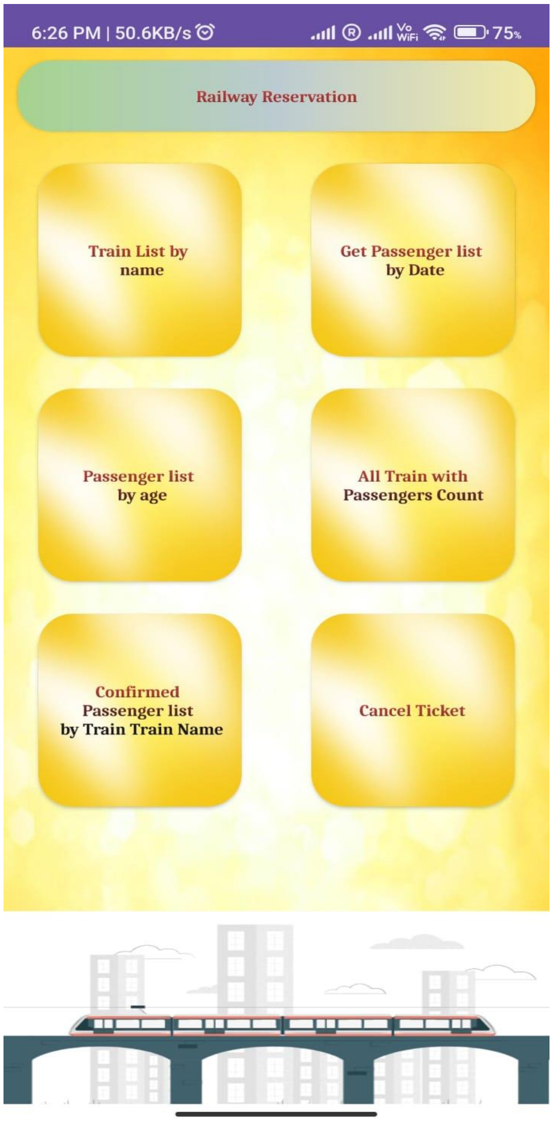


Railway
Reservation

Screen Display

Homepage:

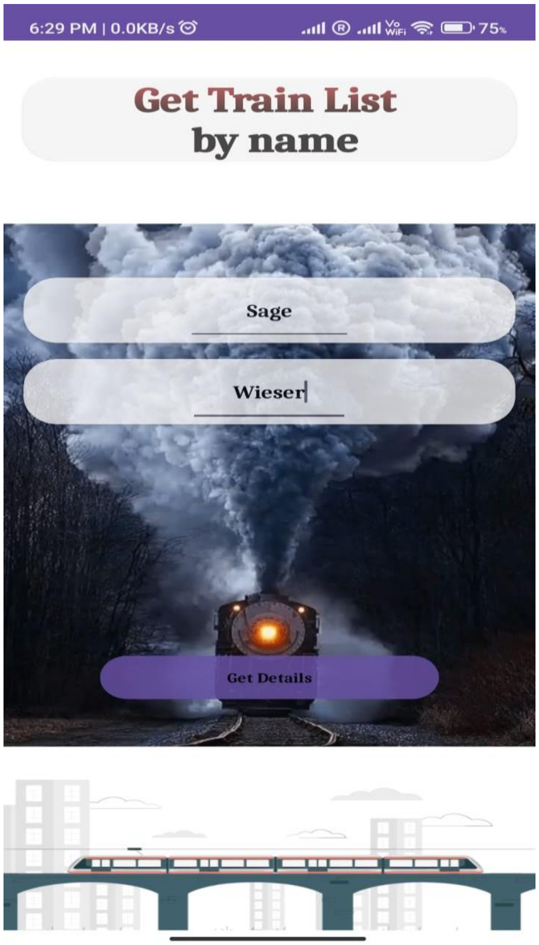
All six queries will appear on the screen as six features, and by clicking on any of them, we are able to verify it.


6:26 PM | 50.6KB/s
75%
Railway Reservation
Train List by name
Get Passenger list by Date
Passenger list by age
All Train with Passengers Count
Confirmed Passenger list by Train Train Name
Cancel Ticket

1) Train List by name:

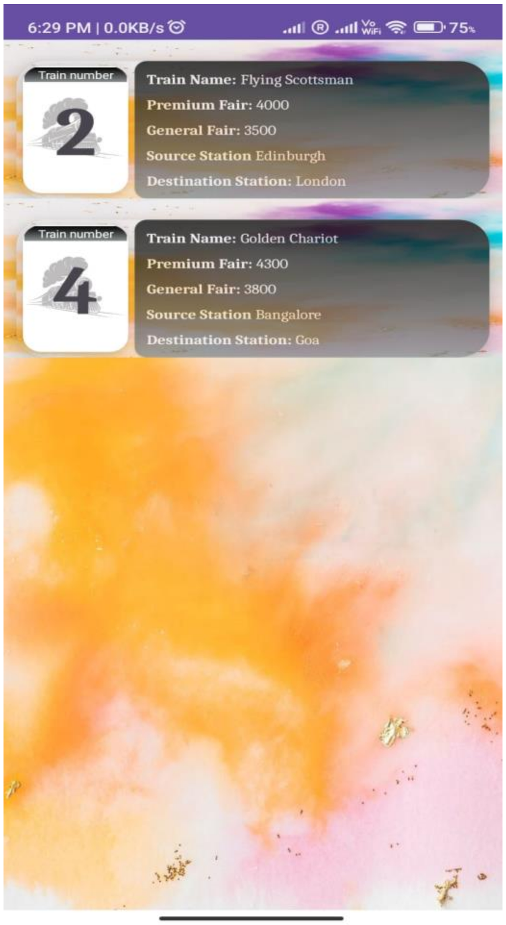
By clicking on the 1st feature, it will redirect to the following page.
User input the passenger’s last name and first name and retrieve all trains they are booked on.

```
db.rawQuery("SELECT *" +
            "FROM Passengers p\n" +
            "INNER JOIN booked b ON p.SSN = b.passenger_ssn\n" +
            "INNER JOIN Train t ON b.train_number = t.train_number\n" +
            "WHERE p.last_name = ? AND p.first_name =?",details)
var details = arrayOf(lastName, firstName)
```



Sage

Wieser



Train number 2

Train Name: Flying Scottsman
Premium Fair: 4000
General Fair: 3500
Source Station Edinburgh
Destination Station: London

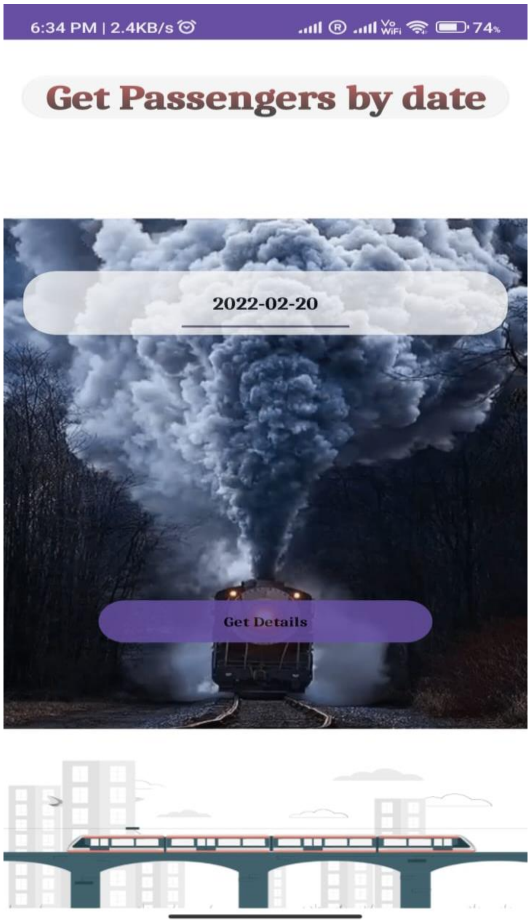
Train number 4

Train Name: Golden Chariot
Premium Fair: 4300
General Fair: 3800
Source Station Bangalore
Destination Station: Goa

2) Get Passenger List by Date:

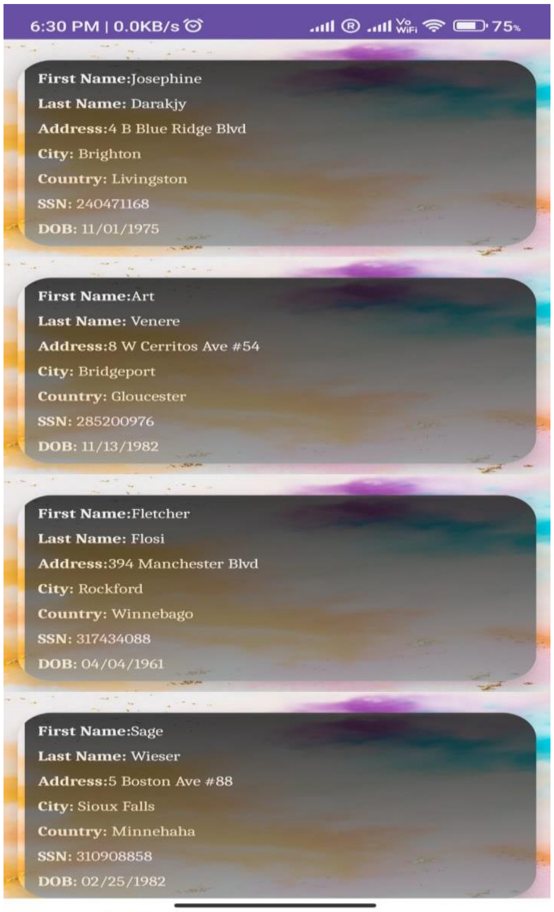
User input the Date and list of passengers travelling on entered date with confirmed tickets displays on UI.”
By clicking on the 2nd feature, it will redirect to the following page

```
db.rawQuery("SELECT *" +
             "FROM Passengers p\n" +
             "INNER JOIN booked b ON p.SSN = b.passenger_ssn\n" +
             "INNER JOIN Train t ON b.train_number = t.train_number\n" +
             "INNER JOIN train_status ts ON t.train_name = ts.Train_name\n" +
             "where  b.status=? and ts.train_date=?"
          ,details)
var details = arrayOf("Booked",day)
```



Get Passengers by date

2022-02-20



First Name: Josephine
Last Name: Darakjy
Address:4 B Blue Ridge Blvd
City: Brighton
Country: Livingston
SSN: 240471168
DOB: 11/01/1975

First Name: Art
Last Name: Venere
Address:8 W Cerritos Ave #54
City: Bridgeport
Country: Gloucester
SSN: 285200976
DOB: 11/13/1982

First Name: Fletcher
Last Name: Flosi
Address:394 Manchester Blvd
City: Rockford
Country: Winnebago
SSN: 317434088
DOB: 04/04/1961

First Name: Sage
Last Name: Wieser
Address:5 Boston Ave #88
City: Sioux Falls
Country: Minnehaha
SSN: 310908858
DOB: 02/25/1982

3) Get Train and Passenger details by age:

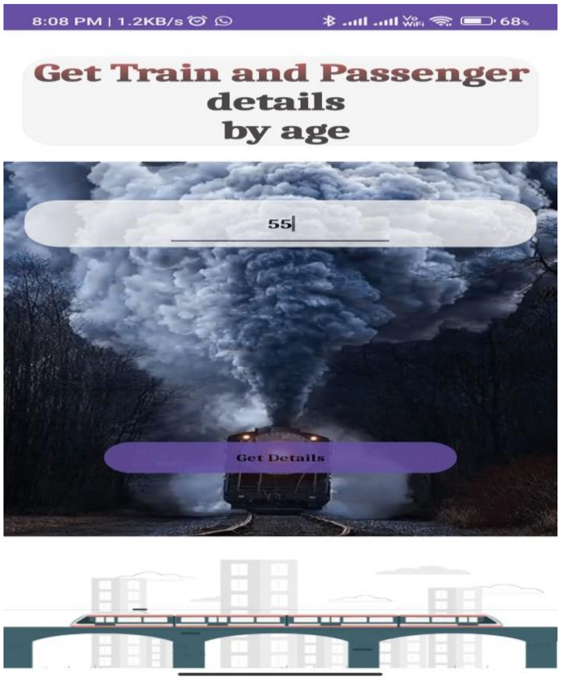
User input the age of the passenger (50 to 60) and UI display the train information (Train Number, Train Name, Source and Destination) and passenger information (Name, Address, Category, ticket status) of passengers who are between the ages of 50 to 60.”
By clicking on the 3rd feature, it will redirect to the following page

Input the age between 50 to 60 retrieve train information and passenger information whose age is the input age.

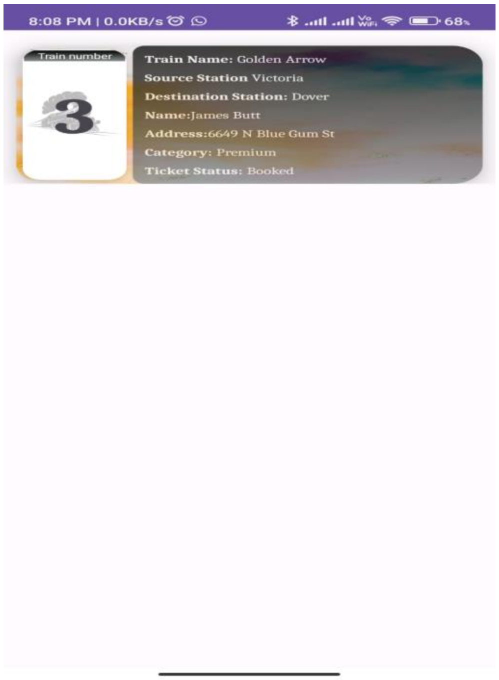
```
db.rawQuery("SELECT Train.Train_Number, Train.train_name,
Train.Source_Station, Train.Destination_Station, Passengers.first_name,
Passengers.last_name, Passengers.address, Booked.Ticket_Type, Booked.Status
\n" +
"              FROM Booked \n" +
"              JOIN passengers ON Booked.passenger_ssn = Passengers.SSN \n" +
"              JOIN Train ON Booked.Train_Number = Train.Train_Number \n" +
"              WHERE strftime('%Y', ?) - CAST(substr(Passengers.bdate,7,4) as
INTEGER)="+ "CAST(? as INTEGER) and (CAST(? as INTEGER) between
CAST(? as INTEGER) and CAST(? as INTEGER))",arr
)
var arr= arrayOf("now",age,age,"50","60")
```

if we want print the details of passenger and train details by any user age we can use following query

```
db.rawQuery("SELECT Train.Train_Number, Train.train_name,
Train.Source_Station, Train.Destination_Station, Passengers.first_name,
Passengers.last_name, Passengers.address, Booked.Ticket_Type, Booked.Status
\n" +
             "              FROM Booked \n" +
             "              JOIN passengers ON Booked.passenger_ssn =
Passengers.SSN \n" +
             "              JOIN Train ON Booked.Train_Number =
Train.Train_Number \n" +
                "                    WHERE strftime('%Y', ?) -
CAST(substr(Passengers.bdate,7,4) as INTEGER)="+ "CAST(? as
INTEGER)",arr
        )
```



55



3

Train Name: Golden Arrow
Source Station Victoria
Destination Station: Dover
Name: James Butt
Address:6649 N Blue Gum St
Category: Premium
Ticket Status: Booked

4) All Trains with passenger count:

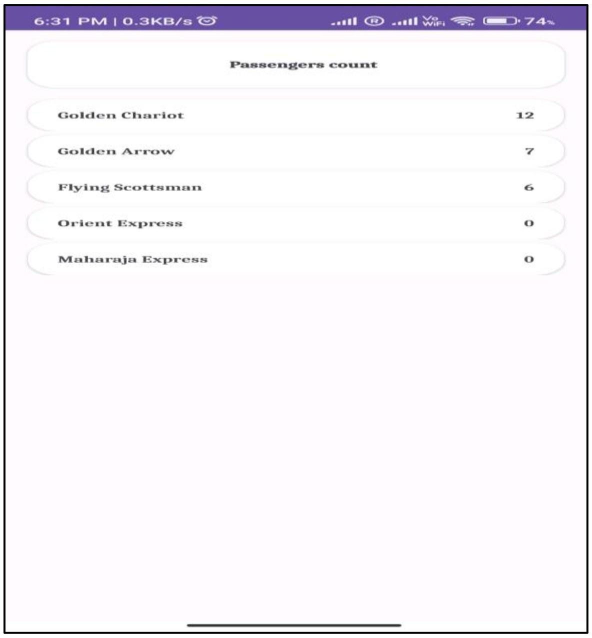
“List all the train name along with count of passengers it is carrying. “

Click on GetTrains button to retrieve all train names along with count of passengers it is carrying.

By clicking on the 4^{td} feature, it will redirect to the following page.

Query:

```
‟  db.rawQuery("SELECT t.train_name as TrainName,
COUNT(b.passenger_ssn) AS PassengerCount\n" +

              "FROM Train t \n" +

              "LEFT JOIN booked b ON t.train_number = b.train_number\n" +

              "GROUP BY t.train_name\n" +

              "HAVING COUNT(b.passenger_ssn) = 0 union all SELECT
t.train_name, COUNT(*) AS PassengerCount from train t INNER JOIN booked b
ON t.train_number = b.train_number group by t.train_name order by
PassengerCount desc", arrayOf()

       )
```



Passengers count

Golden Chariot	12
Golden Arrow	7
Flying Scottsman	6
Orient Express	0
Maharaja Express	0

5) Confirmed passenger list by Train name:

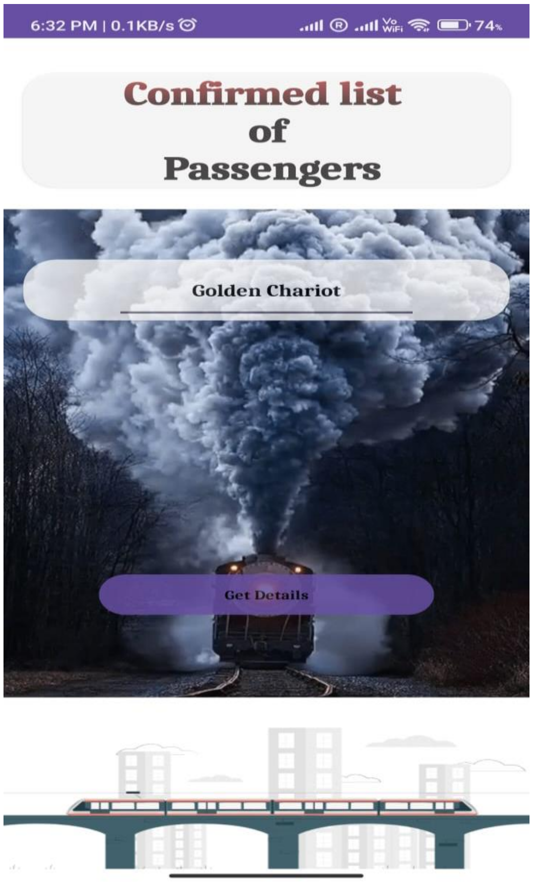
"Enter a train name and retrieve all the passengers with confirmed status travelling in that train. "

By clicking on the 5td feature, it will redirect to the following page

- Enter Train Name to retrieve all passengers with confirmed status travelling in that train.

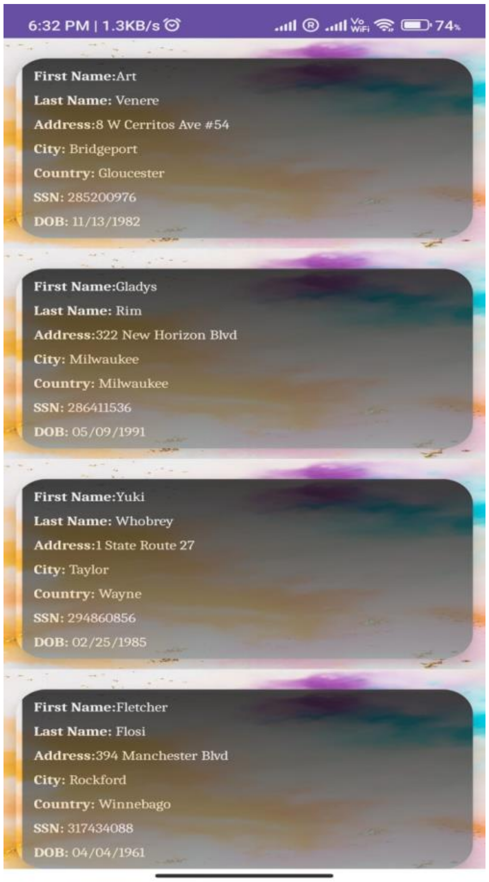
QUERY:

```
db.rawQuery("SELECT p.*\n" +
            "FROM Passengers p\n" +
            "JOIN booked b ON p.SSN = b.passenger_ssn\n" +
            "JOIN Train t ON t.train_number = b.train_number\n" +
            "WHERE t.train_name = ?\n" +
            "  AND b.status =?",arr)
var arr= arrayOf(trainName,"Booked")
```



Confirmed list of Passengers

Golden Chariot



First Name: Art
Last Name: Venere
Address:8 W Cerritos Ave #54
City: Bridgeport
Country: Gloucester
SSN: 285200976
DOB: 11/13/1982

First Name: Gladys
Last Name: Rim
Address:322 New Horizon Blvd
City: Milwaukee
Country: Milwaukee
SSN: 286411536
DOB: 05/09/1991

First Name: Yuki
Last Name: Whobrey
Address:1 State Route 27
City: Taylor
Country: Wayne
SSN: 294860856
DOB: 02/25/1985

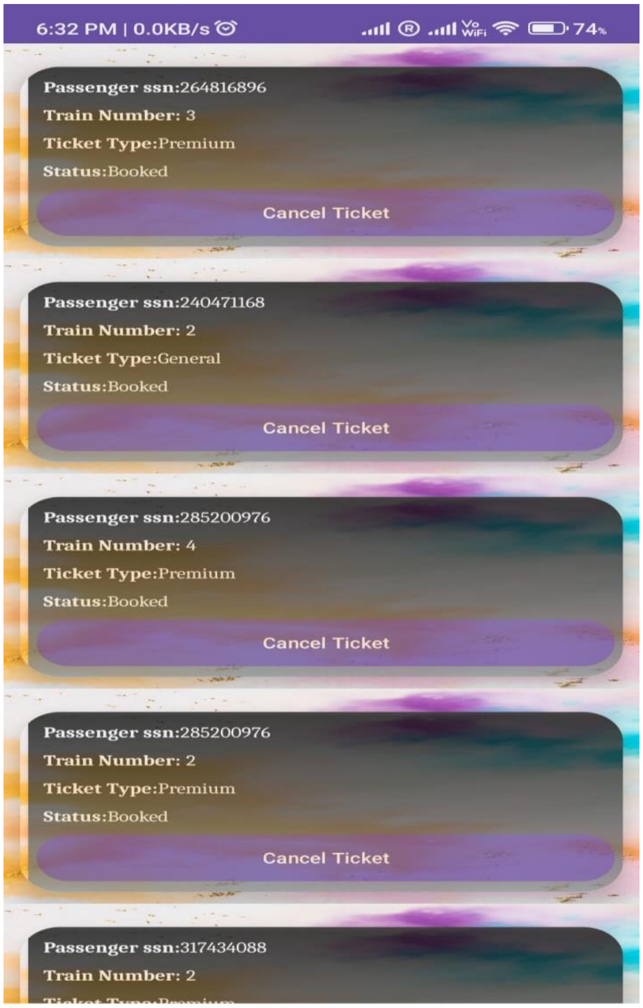
First Name: Fletcher
Last Name: Flosi
Address:394 Manchester Blvd
City: Rockford
Country: Winnebago
SSN: 317434088
DOB: 04/04/1961

6) Cancel Ticket:

“User Cancel a ticket (delete a record) and show that passenger in waiting list get ticket confirmed. “
By clicking on the 6td feature, it will redirect to the following page

QUERY:

```
db.execSQL("delete from booked where passenger_ssn=? and
train_number=?",arr)

db?.execSQL("create trigger updatelist after delete on booked for each row
begin update booked set status=\"Booked\" where passenger_ssn=(select
passenger_ssn from booked where status =\"WaitL\" limit 1); end;")
var arr= arrayOf(ssn,train_num)
```



6:32 PM | 0.0KB/s 74%

Passenger ssn:264816896
Train Number: 3
Ticket Type: Premium
Status: Booked
Cancel Ticket

Passenger ssn:240471168
Train Number: 2
Ticket Type: General
Status: Booked
Cancel Ticket

Passenger ssn:285200976
Train Number: 4
Ticket Type: Premium
Status: Booked
Cancel Ticket

Passenger ssn:285200976
Train Number: 2
Ticket Type: Premium
Status: Booked
Cancel Ticket

Passenger ssn:317434088
Train Number: 2

Contributions:

Hari Krishna Aravakantha

Did the base research for the project and created the GUI.

Worked on 3 query sql queries (train details by passenger name, list of travelling on given date, and passengers by age.)

Abhishek Saraf

Worked on 3 query SQL queries(Passenger count, passengers list by train name and cancel ticket)

Trial and error method for sql errors challenges.