Airline Ticket Reservation System

Mini Project Report -Database Lab (DSE 2260)

Department of Data Science & Computer Applications



B. Tech Data Science

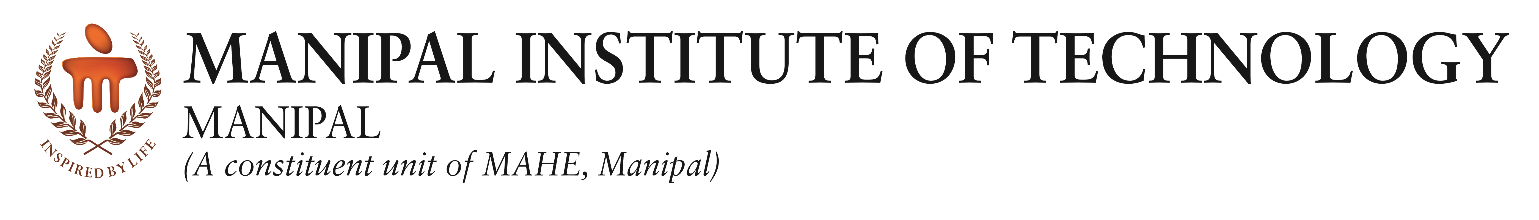
4th Semester – Batch: B2

Submitted By

|  |  |
| --- | --- |
| Preetham Kasyap | 210968190 |
| Aakash Gangurde | 210968192 |
| Parush Oberai | 210968194 |
| Deepesh Adwani | 210968198 |

**Mentored By**

|  |  |
| --- | --- |
| Vinayak M | Archana H/ Dr. Natesh |
| Assistant Professor-Senior | Assistant Professor-Senior |
| DSCA, MIT | DSCA, MIT |



Date:

**CERTIFICATE**

This is to certify that the Aakash Gangurde (210968192), Preetham Kasyap (210968190), Parush Oberai (210968194), Deepesh Adwani (210968198), have successfully executed a mini project titled “Airline Ticket Reservation system” rightly brining fore the competencies and skill sets they have gained during the course- Database Lab (DSE 2262 & DSE), thereby resulting in the culmination of this project.

|  |  |
| --- | --- |
| **Vinayak M** | **Archana H / Dr. Natesh** |
| **Assistant Professor-Senior** | **Assistant Professor-Senior** |
| **DSCA, MIT** | **DSCA, MIT** |

*ABSTRACT*

The airline reservation database project aims to develop a computerized system to facilitate the booking and management of flights for customers. The system will store information about flight schedules, seat availability and customer data. It will provide customers with up-to-date information on flights, allowing them to easily search and select their preferred options, book, and pay for their flights online, and manage their bookings. The system will also provide service to issue ticket to customers.

This project adopts tools such as SQL/SQL+ and JAVA for back-end and front-end respectively. The development of the project begins with the development of schema tables and based on the features that this system would provide, we would define the database system requirements, attributes and constraints. Taking the flight\_details as the parent table and the customer\_details and seats as the child table the schema has been created. The PNR and Flight\_no are the primary keys in flight\_details and customer\_details tables respectively. Seats and customer\_details table contain foreign keys which refer to the primary key present in the flights details table.

**Contents**

**1. Introduction**

**2. Synopsis 2.1 Objectives**

**2.2 Tables**

**3.SRS and Details**

**4. Functionalities**

**5. Detailed Design**

**5.1 ER Diagram**

**5.2 Schema Diagram**

**5.3 Data Dictionary**

**5.4 Relational Model Implementation**

**5.5 Queries**

**6. Implementation Functional Requirements**

**7. Result**

**8. Conclusion and Future Work**

***Chapter 1:***

1.INTRODUCTION

An airline reservation system is a computerized application that facilitates the booking process for customers to reserve seats on an airline. The system typically stores information about flight schedules, seat availability, pricing, and customer information. The system can also handle the and issue tickets to customers. Overall, the airline reservation database project aims to provide a valuable solution to the airline industry, enhancing the customer experience and improving efficiency in flight booking and reservation processes.

***Chapter 2:* Synopsis**

The airline reservation system is a database management system designed to help airlines manage their reservation process efficiently. It is a computerized system that enables customers to book and purchase tickets, as well as make changes to their bookings. The database schema for the airline reservation system includes tables for airlines, flights, seats, passengers and reservations. Each table has its own set of attributes that describe the entities and their relationships. For example, the flights table may have attributes such as flight number, departure time, arrival time, and destination airport.

2.1) Objective:

The project's objective is to provide booking interface to the customers which provides features such as: -

* To provide customers with up-to-date information on flight schedules, seat availability, view their past flight history, and pricing.
* To enable customers to book flights through this interface.
* To provide a database for both customers and administrators.
* To provide customers with the ability to manage their bookings, including making changes or cancellations.
* To maintain accurate and reliable data.
* To provide an interface for users to change their flight.

**2.2) Proposed system**

The airline reservation system is designed to manage the reservations made by users. It can also modify or cancel the flights made by the user and then finally generate an itinerary which can show all the required information which a user need in order to fly and get to their respective flights.

**2.2) Tables:**

* Flight
* Booking
* Seats
* Login

***Chapter 3:******SRS (System Requirement and Specification) and details***

In the context of a Database System (DBS), an SRS (Software Requirements Specification) is a document that describes the functional and non-functional requirements of the database system.

These are the following software which we used for our airline reservation database system :

1. SQL Plus client
2. VSCode
3. OJDBC (Oracle Java Database Connector)

The application starts with a login prompt where it asks the user about the username and the password, after entering the username and the password if the credentials exist then booking flight prompt will be displayed.

* While creating a booking by a user an **INSERT** query will run which will enter the entered details in the table.
* While changing the flight details an **UPDATE** query will run which will change the record details and update the record of the user/passenger details.
* While cancelling a flight a **DELETE** query will run in order to delete the record for the flight.
* While editing the passenger details an **UPDATE** query will run which can be used to change the seat details or the names/personal details.

***Chapter 4: Functionalities***

Some of functionalities provided by this system are:

* Ticket cancellation and reservation system
* Flight changing privileges.
* Seat changing facility.
* Check-in system
* Viewing current itinerary and past flight history.
* Ticket booking system.

***Chapter 5: Detailed Design***

**5.1) ER DIAGRAM**

|  |
| --- |
| **seats** |

|  |
| --- |
| **Booking** |

flight

|  |
| --- |
| **Login** |

**Seat allocation**

Booking\_passengers

Flight\_allocation

|  |
| --- |
| Login  **Username**  Password |

|  |
| --- |
| Booking  Name  ***Contact***  Age  Departure  Arrival  Travel\_date  ***PNR***  acc |

|  |
| --- |
| Seats  Name  Seat\_no  ***Contact***  ***PNR***  Price  Flight\_no  acc |

**5.2) SCHEMA DIAGRAM**

|  |
| --- |
| Flight  **Flight\_Number**  Seat\_available  Cost\_per\_seat |

**Schema: -**

**Login** (**Username,** password)

**seats** (Name**,** Seat\_no, **contact, PNR**,price,flight\_no,acc)

**Flights** (**Flight\_Number,** seat\_available, cost\_per\_seat)

**Booking** (name**,Contact**,**PNR** age,departure,arrival,travel\_date,acc)

**5.3) Data Dictionaries**

**1)Seat**

|  |  |  |  |
| --- | --- | --- | --- |
| column | Data type | constraint | Constraint\_name |
| Name | Varchar2(20) | - |  |
| Seat\_no | Varchar2(10) | Unique |  |
| Flight\_no | Varchar2(10) | References flight | Fk\_fno |
| PNR | Number(3) | Comp primary key | - |
| Contact | Number(10) | Comp primary key | - |
| Price | Number(6) | - | - |
| acc | Varchar(10) | References login | fk\_acc |

**2)Login**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data Type | constraint | Constraint name |
| Username | Varchar2(200) | PRIMARY KEY | Pk\_login |
| Password | Varchar2(200) | - | - |

**3)Booking**

|  |  |  |  |
| --- | --- | --- | --- |
| column | Data type | constraint | Constraint name |
| Name | Varchar2(20) | - |  |
| Contact | Number(10) | Comp primary key | - |
| age | Number(3) | Between 0-110 |  |
| arrival | Varchar2(200) | - | - |
| departure | Varchar2(200) | - | - |
| PNR | Number(3) | Comp primary key | - |
| Travel\_date | date | Format(‘DD-MM-YYYY’) |  |
| acc | Varchar(100) | References login | Fk\_acc |

**4)flights**

|  |  |  |  |
| --- | --- | --- | --- |
| Column | Data type | constraint | Constraint name |
| Flight\_number | Varchar2(100) | PRIMARY KEY | Pk\_flights |
| Seat\_available | Number(3) | - | - |
| Cost\_per\_seat | Number(5) | - | - |

**5.4) Relational model implementation and Queries**

**LOGIN TABLE**

create table login(username varchar(100) primary key, password varchar(100));

**FLIGHT TABLE**

create table flight(flight\_number varchar(100) primary key,seat\_available number(3),cost\_per\_seat number(6));

**BOOKING TABLE**

create table booking(name varchar(20), age number(2),contact number(10),departure varchar(30),arrival varchar(30),travel\_date date,pnr number(3),primary key(contact,pnr));

**SEATS TABLE**

create table seats(name varchar(20), contact number(10), flight\_no varchar(10) references flight(flight\_number), seat\_no varchar(10) unique,price number(6),pnr number(3),primary key(contact,pnr));

**LAST FLIGHT SELECTION:-**

select \* from booking where pnr=(select max(pnr) from booking where acc='user\_account');

**ALL DETAILS OF LAST FLIGHT SELECTION:-**

select \* from booking,seats where booking.pnr=(select max(pnr) from booking where acc='admin') and booking.contact=seats.contact and booking.pnr = seats.pnr;

INSERTING INTO BOOKING TABLE

s1.executeQuery("insert into booking values ('"+name[temp]+"',"+age[temp]+","+contact[temp]+",'"+des+"','"+arr+"',to\_date('"+travel\_date+"','DD-MM-YYYY'),"+i+",'"+uname+"')");

INSERTING INTO SEATS TABLE AND UPDATING THE VALUE OF AVAILABLE SEATS

st1.executeQuery("insert into seats values('"+name[temp]+"',"+contact[temp]+",'"+flight+"','"+s1+"',"+price+","+i+",'"+uname+"')");

st1.executeQuery("update flight set seat\_available = seat\_available-1 where flight\_number = '"+flight+"'");

INSERTING FLIGHT INFO IN THE FLIGHT TABLES TO PRE POPULATE IT:

insert into flight(flight\_number) values('6E 1021');

insert into flight(flight\_number) values('GA 3212');

insert into flight(flight\_number) values('NS 2112');

insert into flight(flight\_number) values('JH 8898');

insert into flight(flight\_number) values('ZN 2901');

insert into flight(flight\_number) values('RA 3333');

insert into flight(flight\_number) values('PY 7139');

insert into flight(flight\_number) values('LN 5459');

insert into flight(flight\_number) values('WD 2192');

insert into flight(flight\_number) values('QX 4839');

update flight set seat\_available=150, cost\_per\_seat = 5400 where flight\_number = '6E 1021';

update flight set seat\_available=175, cost\_per\_seat = 7400 where flight\_number = 'GA 3212';

update flight set seat\_available=275, cost\_per\_seat = 3400 where flight\_number = 'NS 2112';

update flight set seat\_available=100, cost\_per\_seat = 4700 where flight\_number = 'JH 8898';

update flight set seat\_available=122, cost\_per\_seat = 7700 where flight\_number = 'ZN 2901';

update flight set seat\_available=322, cost\_per\_seat = 17700 where flight\_number = 'RA 3333';

update flight set seat\_available=250, cost\_per\_seat = 9700 where flight\_number = 'PY 7139';

update flight set seat\_available=166, cost\_per\_seat = 6700 where flight\_number = 'LN 5459';

update flight set seat\_available=196, cost\_per\_seat = 4700 where flight\_number = 'WD 2192';

update flight set seat\_available=276, cost\_per\_seat = 4382 where flight\_number = 'QX 4839';

IMPORTANT QUERIES:

s1.executeQuery("update booking set contact = "+con2+" where contact = "+con1+" and acc = '"+uname+"'"+" and pnr = (select max(pnr) from booking where acc='"+uname+"')");

s1.executeQuery("update seats set contact = "+con2+" where contact = "+con1+" and acc = '"+uname+"'"+" and pnr = (select max(pnr) from booking where acc='"+uname+"')");

s1.executeQuery("update seats set seat\_no = '"+seat+"' where contact = "+con+" and acc='"+uname+"'"+

                        "and pnr = (select max(pnr) from booking where acc='"+uname+"')");

ResultSet r1 = s1.executeQuery("select \* from booking,seats where booking.pnr=(select max(pnr) from booking where acc='"+uname+') and booking.contact=seats.contact and booking.pnr = seats.pnr");

System.out.println("name\tcontact\t\tflight number\tseat number\tdeparture\tarrival");

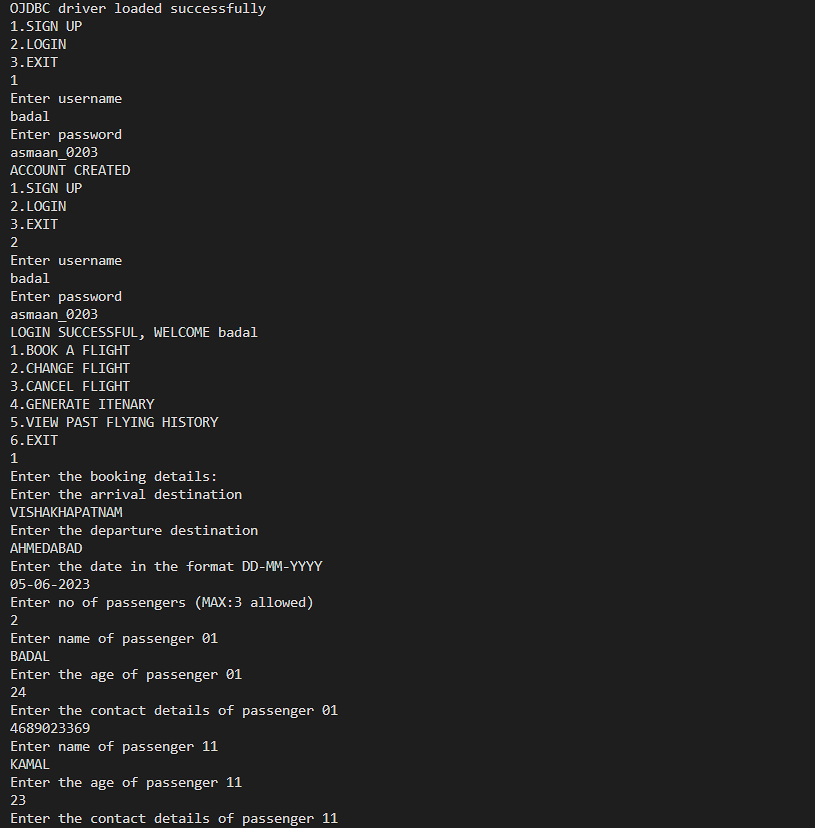
***Chapter 6: Implementation and functional requirements***

**To run the program the requirements are:**

* The working body has been coded in java language using **VSCode** so the needed software is **JDK** and VSCode
* The connection between java and SQL is made with **OJDBC** (Oracle Java DataBase Connector).
* The database is stored using Oracle SQL Plus 21c Express edition.
* The **OJDBC jar** file is also needed to implement the connection and ResultSet execution.
* The function which are made outside main functions but called in the main function require a pre populated of flight details so a pre populated table is required.

***Chapter 7: Result***

The result is that after logging into the interface a user enters his details along with the passenger details. This in turn generates an itinerary with the required details which are necessary to fly with an airline.

***3***

***Text

Description automatically generated***

***\*The currently made reservation can also be changed or cancelled.***

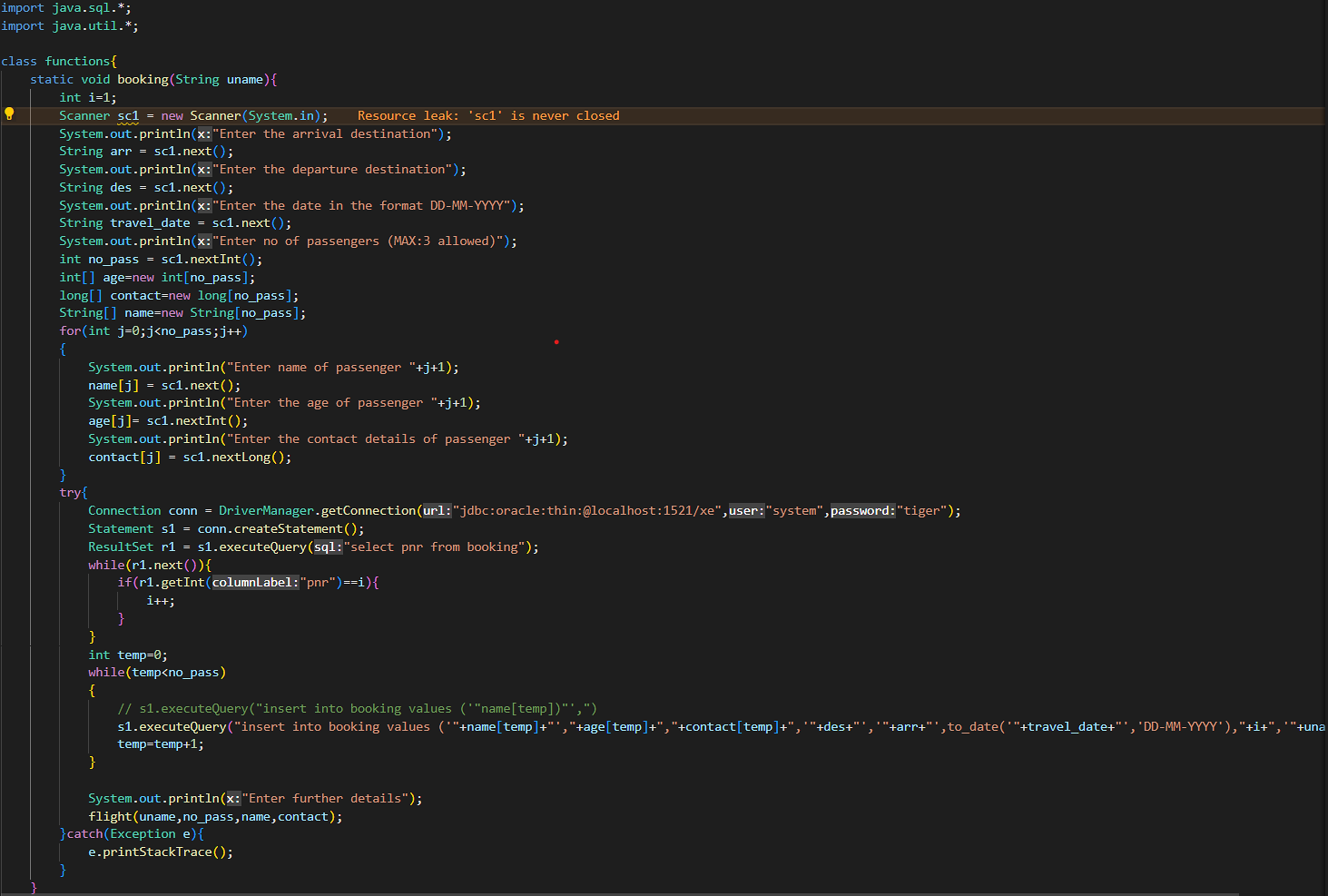
***Chapter 8: Conclusion and future work***

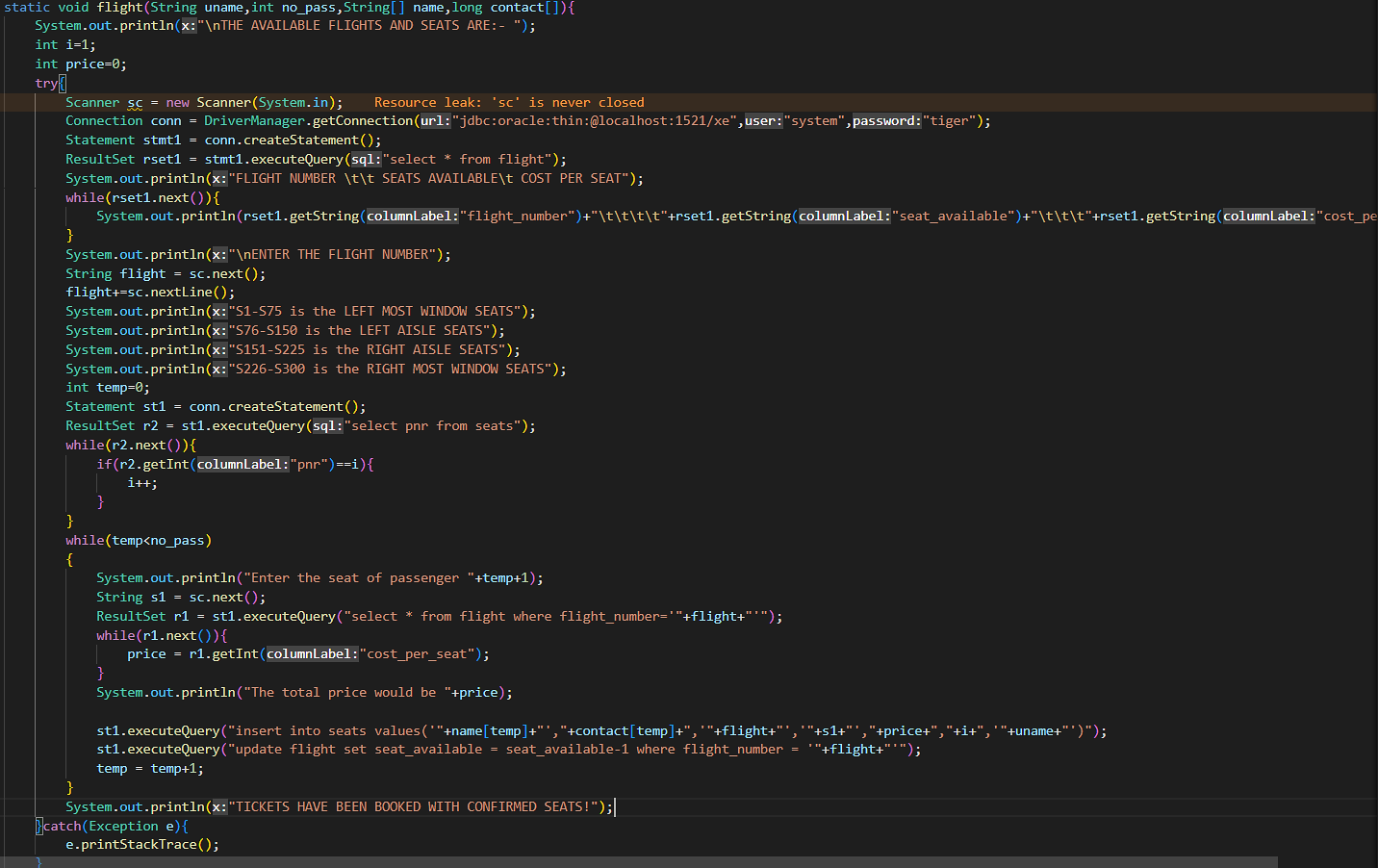
The airline reservation system has successfully made an interface with which a user can book flights and the details can be saved in a database for further work on how to make the interface more efficient by reducing some of the constraints and columns and it can also make changes in the itinerary and cancel flights that is in terms of SQL it can perform update and delete queries as needed.

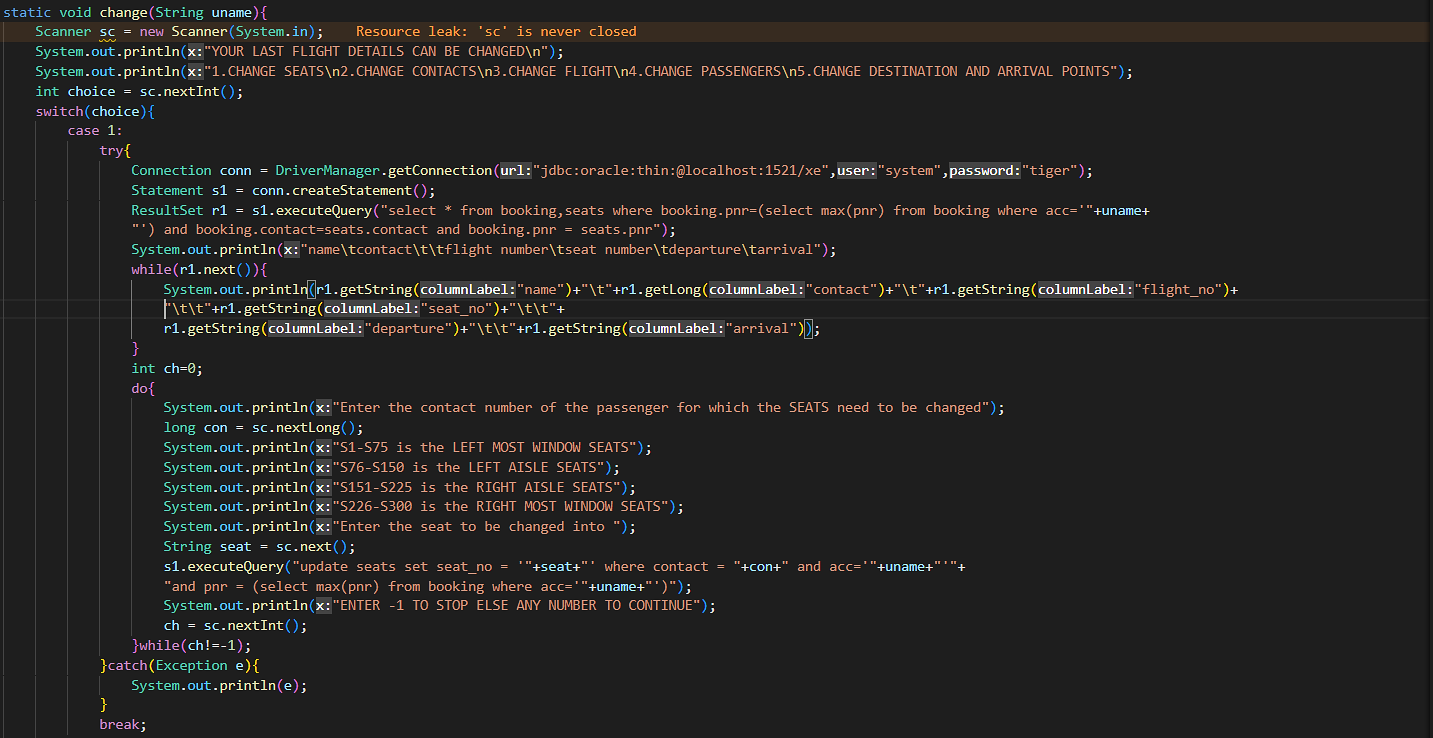
The future work in accordance with this project can be a more precise workflow where every detail of a passenger is mapped with table which ensures that the passenger has really made a reservation to prevent other people to access his/her itinerary. This project also has a lot of potential where a well-designed GUI can be implemented using JAVA SWINGS to make the user interaction better and with GUIs the payment platform can also be made seamless. With the inclusion of payment platform, we can encrypt the complete data which can make it even more secure.

***CODE SNIPPETS***

**FUNCTIONS:**

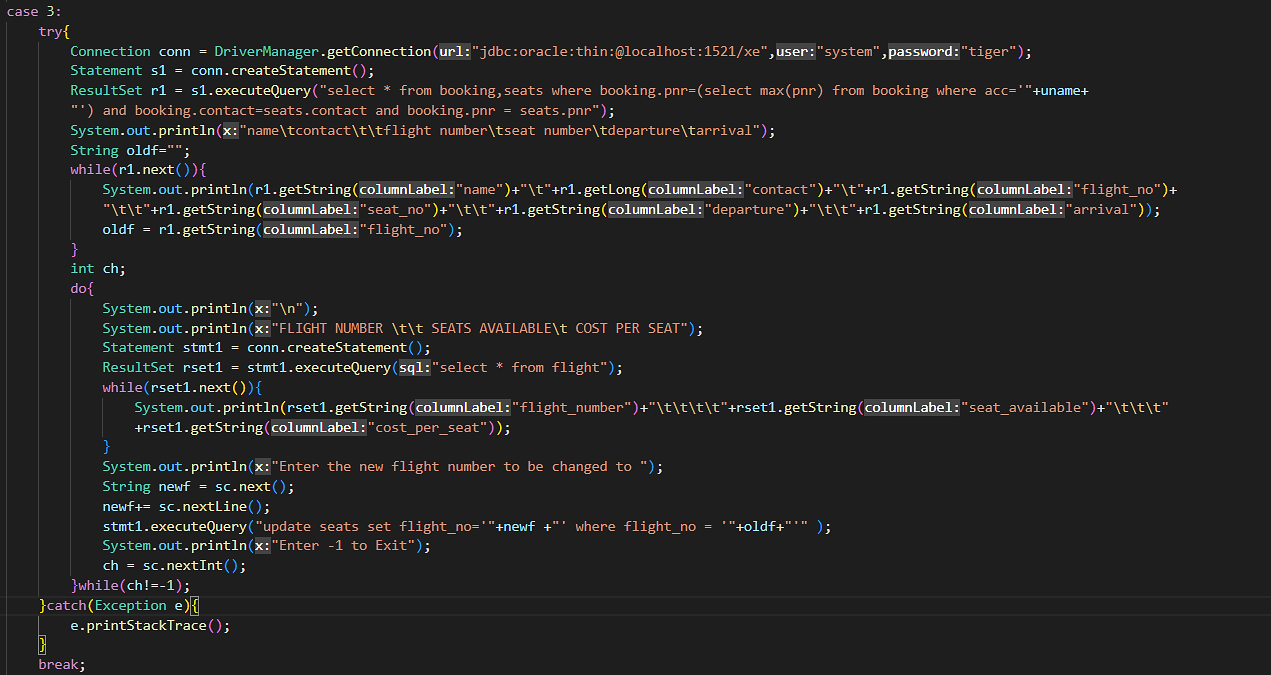






Text

Description automatically generated



Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

A picture containing text

Description automatically generated

**MAIN FUNCTION:**

Text

Description automatically generated

Text

Description automatically generated



**Contribution of members:-**

1. Preetam kasyap: - Contributed in report generation. Contributed in change function made in java to execute some of the update queries. Contributed in sql queries for inserting and displaying entries along with update queries. Contributed in designing table cardinalities.
2. Aakash Gangurde: - Contributed in menu driven interface and OJDBC working and execution. Made functions for queries and functions to display using ResultSet and update queries to run using change function. Contributed in schema diagram and data dictionaries. Contributed in report generation. Contributed in making flight function which executes seat booking process and inserting seat details into the seat table.
3. Parush Oberai:- Contributed in java for menu driven program and made functions for inserting queries. Contributed in ER model, established relations between tables. Contributed in booking function which inserts user details into the booking using PNR number. Contributed in table cardinality.
4. Deepesh adwani:- Contributed in java functions to run sql queries and in planning the vital role of keys in tables. Also contributed in cancel function along with itinerary generation and displaying past flight history using a function.

GRP MEMBERS: -

Preetham Kasyap 210968190

Aakash Gangurde 210968192

Parush Oberai 210968194

Deepesh Adwani 210968198