**Project Report**

**Design Railway Reservation**

**System Database and GUI Application**

**By:**

**HOAN LE and ADEWALE OBALANLEGE**

**Honor code:**

I pledge, in 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.

**ER Diagram for the RRS Database:**

The railway reservation system database includes information about passengers, trains, train status, and booking status. The ER diagram below provides the structure and relations of all the entities in the system.

Entity type Passenger contains passenger’s name (first names and last names), SSN as primary key, DOB, age (derived from DOB), address (street, city, and county), and phone number. Entity type Train contains the train name, train number as primary key, source, destination, days of availability (Monday to Sunday), and the cost of premium ticket and general ticket. Lastly, entity type Train Status contains the train date, train name as primary key because these are unique values, total number of seat available and occupied for premium and general seat.

Passenger can check the status of Train Status and book a Train or multiple Trains based on the sources, destinations, and available seats. Since there are maximum 5 trains and a train can have many passengers or no passenger, (0, 5) is specified for Passenger in Book, and (1, n) is specified for Train in Book. A passenger can book many trains, therefore, can check status of many trains, and many passengers can also look at the status of the same train. So, Passenger and Train Status are many-to-many relation to Check Status relationship. A Train has a Train Status specified by 1-to-1 relation. Booking date, type of ticket, and status of reservation are also recorded when a passenger books a train.

Figure 1: The ER Diagram of the railway reservation system.

1

1

N

(1, n)

(0,5)

M

Passenger

Check Status

Train Status

Book

Train

Has

**GUI Task:**

1. Task 1(input the passenger's last name, first name, and retrieve all the trains they booked):

In this task, the passenger’s first name and last name was provided, the Query 1 button will submit the command to the database and retrieve the passenger’s train name and train number that they are booked.

Graphical user interface, table

Description automatically generated

Running Query 1 for passenger James Butt, the result is train\_name: Golden Arrow, and train number 3. This passenger only booked 1 ticket and 1 row returned as a result.

Graphical user interface, table

Description automatically generated

Running Query 1 for passenger Art Venere, there are 2 results: train\_name: Golden Arrow, and train number 3, and train\_name: Flying Scotsman, and train number 2. So, this passenger booked 2 tickets, 2 rows returned as a result.

1. Task 2 (Input the day and retrieve list of passengers with confirmed tickets on that day):

In this query, any day of the week (Monday to Sunday) will be the input. The Query2 button will submit the command to the database to retrieve the list of all passengers that booked on that day.

Graphical user interface

Description automatically generated

The result shows that James Butt has ticket booked on Monday, and he is the only one booked ticket on Monday (1 row return).

Graphical user interface, table

Description automatically generated

The result for Friday shows that there are 5 passengers booked ticket on that day, their first name and last name was retrieved as in order.

1. Task 3 (input passenger's range of age, and retrieve train, passenger information for passengers between that range):

Query 3 button will submit the command to calculate all the passengers’ age and retrieve only the information of the passenger that has the age in the range given. In the figure below, the range of the age input is 50-60, and there is only one passenger who has age in that range. The function calculates the age is 54-year-old. The name, address, train name, train number, ticket type and reservation status of the passenger is also retrieved.

Graphical user interface, table

Description automatically generated

1. Task 4 (list all train names along with the count of passengers it is carrying):

Query 4 button will submit a command to count all the passengers that has booked reservation status, group them by the train name, and display results as train name and count of passengers it is carrying.

The figure below show that there are 5 passengers booked Flying Scotsman train, 1 passenger booked Golden Arrow train, and 12 passengers booked Golden Chariot train.

Graphical user interface, table

Description automatically generated

1. Task 5 (input train name, and retrieve passengers with confirmed status in that train):

Query 5 button will submit the command to retrieve all the passengers that have reservation status booked, and the train name they booked matching the train name provided in the input box.

Graphical user interface

Description automatically generated

In the figure above, the train Golden Arrow has 1 passenger booked: James Butt, and the SSN of the passenger is 264816896.

Graphical user interface, table

Description automatically generated

In the figure above, the train Flying Scotsman has 5 passengers booked, their names and SSN are retrieved from the database. More passengers can be seen by scrolling down the result box.

1. Task 6 (user cancel a ticket, and show that passenger in waiting list get ticket confirmed):

The Query 6 button will submit a command to delete a passenger that booked a train. The function finds the matching passenger from the database in the Booked table that has matching SSN, train number, ticket type and reservation status, the delete record, and output the message ‘Cancel success!’.

Another function retrieves the next passenger in the waitlist with the same train number and ticket type, update their record to booked status, and display their name with the message ‘has Booked!’

In the figure below, the passenger with the SSN number 264816896 was deleted from the record, and passenger Minna Amigon was in the waitlist for train number 3, Premium ticket been updated to booked status.

Graphical user interface

Description automatically generated

In the figure below, the passenger with the SSN number 24071168 was deleted from the record, and passenger Abel Maclead was in the waitlist for train number 2, General ticket been updated to booked status.

Graphical user interface, application

Description automatically generated

![A picture containing calendar

Description automatically generated]() A picture containing calendar

Description automatically generated A picture containing calendar

Description automatically generated

From left to right, the first figure shows the original database. the second figure shows the 1st passenger (SSN 26481696, train number 3, and Premium ticket) was removed from the Booked table, and the passenger with SSN 256558303 with train number 3, Premium ticket in the Waitlist was updated to Booked. The third figure shows that the passenger with SSN 240471168, train number 2, General ticket was removed from the Booked table, and passenger with SSN 302548590, train number 2, General ticket, and in the waitlist was updated to Booked.

**Conclusion**

The Railway Reservation System allows users to easily query passengers’ information, train and booking status, including ticket types and available seats. The user-friendly GUI window provides a convenient tool for a quick query with just one click. This combination of the database management system and other programing languages such as Pythons provides the ideal platform for storing and updating record, data reporting, and customer services.

**Team Contribution:**

Hoan Le: Create ER diagram, write sql queries, test GUI functions, write report.

Adewale Obalanlege: Create database schema, write sql queries, create GUI, write report.