**Journey Now**

*An Airline Reservation System*

Created by:

Tien Vo

Shah Mirza

Minal Dalwadi

Project Report

**Functionality Requirements**

Airline reservation systems have a very complex reservation system. This makes it the best type of project to simulate for aspiring application developers. When simulating the airline reservation system, there are several requirements that are necessary to include. These functionality requirements include a splash screen upon start of the application, the main menu asking the user to register (requires first name, last name, address, zip, state, username, password, email, SSN, and a security question for password recovery) or login (requires username and password), and a password retrieval using username with security question. Furthermore, the system should be designed to support two types of users: the administrator (who has more privileges in the application) and the customer. After logging into the application, a customer should be able to: search the database for flights depending on the criteria input (such as from city to city, date, time of flight, etc.), book a flight and add it to their account, delete a flight from their account, not book the same flight more than once, be warned of any date and time conflicts the flight may have, and be prevented from booking a flight if it is full (as it will be tracked in the application of how many passengers are booked for the flight currently). The admin will also be able to perform the above-mentioned activities, but they will also be able to add, update, or delete a flight. Finally, the booked reservation should still be visible in the user’s account upon logging back in after logging out, and there should always be an option on every screen to return to the Main Menu.

To successfully manage this project, our team collaborated on GitHub and tested out the codes on Eclipse. We assigned tasks based on where the strengths lie in the group. However, in the end, to ensure everyone was on the same page, meetings were held to discuss the activities each person completed. As one person explains how they went about creating the database or GUI of the airline reservation system, the other two would follow along and ask any question they may have. This guarantees that all members of the project team are ready to troubleshoot should the lead of that department not be available. The objective of this project is to accurately simulate an airline reservation system. A customer should be able to log into the system, create an account if need be, answer security questions if they forgot their password, look up flights to/from certain destinations, input the dates of travel, book a flight, delete a flight, and view their upcoming flights on their account.

There were many constraints our team encountered when completing the project. These mainly originated from the many connectors and subsequent databases that need to be utilized as well. The various external resources used to make this project run as efficiently as possible include: GitHub, JavaFX, and MySQL. Collaborating in GitHub was a difficult process due to the new conceptuality of pushing and pulling other team members’ code. In JavaFX is Scenebuilder, where our team built the GUI of the application. We also had some issues of the program not allowing us to code as needed. Finally, ensuring MySQL is properly connected, importing or exporting MySQL data from one desktop to another, and setting up the workspace in general took a lot of time. We overcame most of these constraints by conducting research on the issue and watching videos of other people who have overcome the same issue.

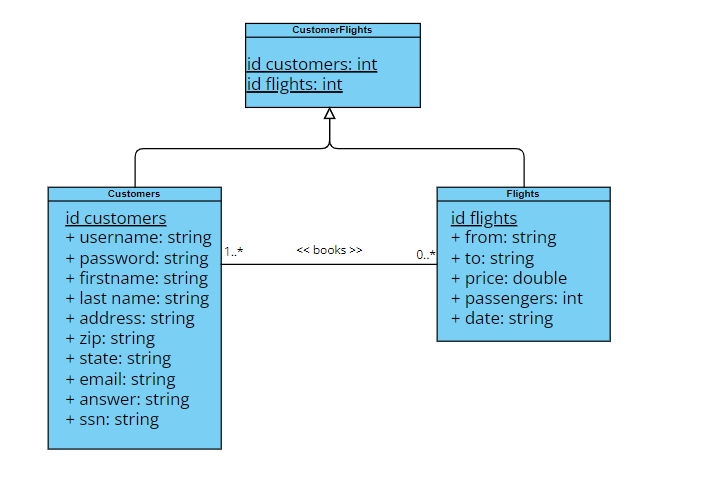
**Non-Functional Requirements**

In addition to the functional requirements needed to create the system, our team had to ensure there are non-functional requirements to make the system intuitive for users. Such requirements include efficiency, usability, organizational, and reliability requirements. Firstly, the efficiency requirements are determined from how easily a customer can navigate the system and book or cancel a flight from the list of flights provided. The usability provides the customer with tasks that can be performed quickly and efficiently (such as creating an account and navigating to their flight list). The organizational requirement of the system involves how the frontend and backend of the system is built. The front end (the GUI) is built with JavaFX and the backend (the data) connects Java with MySQL. Finally the reliability requirements are determined by how accurately the system reports the information back to the user.

**System Objectives**

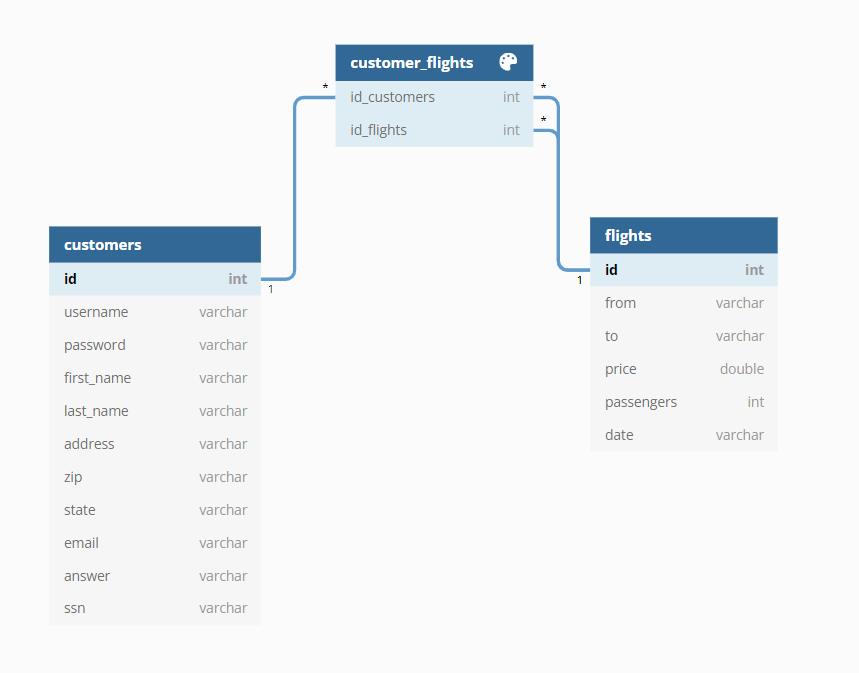
There are three main objectives of this airline. Firstly, by allowing customers to book flights online through this reservation system, airlines will be able to save money because this computerized system will eliminate the need to hire human workers to help customers book flights. By the same logic, customers/users will also save time with this airline reservation system. They can navigate the system due to the intuitive design. The system will also efficiently add users so they have their own account to manage and view their flight details.

**Class Diagram using UML**



This class diagram was constructed in accordance with the data compiled in MySQL and pulled into the Java program in Eclipse. The parent class CustomerFlights consists of the child classes Customers and Flights. This is represented by a generalization relationship (solid line with an arrow). The Customers class and Flights class have a primary key (“id customers” and “id flights” respectively) and several attributes to go along with them. There is also a relationship between the Customers and Flights class. A customer can book either no flights or many flights (as long as there are no time conflicts or if the flight has not already been booked by the customer), and a flight can be booked by one or many customers.

**Data Model**



This data model defines and orders the project data for use and analysis by certain business processes. The goal of data modeling is to produce high quality, consistent, and structured data for running business applications and achieving consistent results. This model was based on the class diagram shown above and the tables in the MySQL database. The id\_customers and id\_flights field are both int data types and primary keys.

**Flow and Functionality of Program**

After the completion of this project, customers will be able to open the application and perform the necessary actions to book one or more flight(s). Customers will be able to open the application, “Journey Now”, and see an aesthetically pleasing splash screen and be prompted to click a “Start your ” button that will lead them to the login screen. This is where the user can enter their username and password. If they do not remember their username or password, they can click the link to recover their username or recreate their password. However, if the user does not have an account with the application at all, they can create an account by entering their email, address, zip code, and state. They will be prompted to create a username and password, and then they will be required to choose a security question and enter an answer for that question in the event they will need to recover their username or password in the future. Finally, for additional security, the user will be asked to input their social security information.

After the customer is logged into the application, they will be presented with the main page. This page will have all the necessary functionalities to book and view the list of flights. The customer can choose the departing location, destination location, and the dates of travel. With this information, the system will provide a list of flights that match the needs of the customer. The customer can then select the flight they want to book to view its details (price, dates of travel, destination/departure locations) and book the flight if it is to their liking. A customer can book more than one flight as long as the flights do not have a time conflict overlap.

However, in the event that a customer needs to cancel/delete a flight, they can do this from their account. In the top right corner, the customer will see the “Your Trips” button which will take the page where they can view their list of booked flights and profile/account information. This is where a customer can delete a flight and edit their profile information if needed.

Finally, to log out, the customer can click the other button in the top right corner of the application asking if the user wants to log out.