*Johanna Blumenthal*

*MSSE 670 | Assignment 1*

*Flight Reservation System*

Project Specification

**Table of Contents**

**Section Page**

Project Description 2

Context Diagram 3

Stakeholders 3

Use Case Diagram 4

Use Case Descriptions 4

**Project Description**

The Flight Reservation System (“FRS”) is essentially an electronic airline booking system that allows customers to search through flights from various airlines based upon the customer’s desired departure/arrival times and locations.

The system has two main users: the customer, who will be booking airline flights and the reservation manager, who is in charge of managing the flights.

The customer must be able to create an account, login to the account, search for available flights, select flights to create an itinerary and then be able to manage that itinerary (i.e. confirm it through payment and/or cancel it).

The reservation manager must be able to login and manage the flights on the system (i.e. add, update, delete). The manager should also be able to create a report of all the flights and reservations.

Since the system will be implemented in Java, we will need to determine what classes we will need. The requirements have specifics about attributes for flights and customers.

Class Flight {

Enum airline;

int flightNumber;

String departureCity;

Date departueTime;

String arrivalCity;

Date arrivalTime;

double priceBusiness;

double priceEconomy;

}

Class Customer {

String name;

List<Address> address;

String username;

String password;

String email;

List<Card> card;

}

In addition to what is stated in the description, the following items seem important:

* ADA compliance as the system is an airline reservation system and could be considered a public accommodation (this means the user interface should include features for those who are low-vision or blind)
* It probably needs to interface with a payment processing system to process the credit card payments from the customers
* The system is likely to be implemented using web and mobile to make it available to customers more conveniently
* We may need to consider the traffic that the system will get from customers

**Context Diagram**

![Diagram

Description automatically generated]()

Both types of users will have a login screen that allows them to login to the system. Customer users will login and be taken to the customer user interface, which will give two major options: 1) search for and book new flights or 2) view booked flights. The reservation manager will have two options once logging in: 1) generate a report or 2) manage flights.

**Stakeholders**

**Customer:** The customer should be able to register, search for available flights, reserve, book and cancel flights.

**Reservation Manager:** The reservation manager needs to be able to login, add a new flight, edit an existing flight or delete the flight (including the prices for seats). The reservation manage should also be able to get a report of all flights and bookings.

**Owner:** We were not told anything about the owner of the FRS, but I imagine there is one, who needs the system to work well and to be compliant with all laws and regulations.

**Developers:** Those who are making the system…it will be Java developers (for this class me).

**Airlines:** The airlines are the ones who are creating the real flights in the world that will be displayed in the FRS…the FRS could impact them and their ability to function. They are likely partners to our FRS.

**Payment Processor:** I assume we will use a payment processor to charge customers and will need to interact with the payment processor and systems they have.

**Use Case Diagram**

![Diagram

Description automatically generated]()

Use Case Descriptions

|  |  |
| --- | --- |
| **Use case name:** | Register |
| **Use case ID:** | 001 |
| **Primary actors:** | Customer |
| **Secondary actors:** |  |
| **Brief description:** | Customer Registers to use the system |
| **Preconditions:** | Each user should be unique; need to validate credit card |
| **Flow of events:** | 1. The customer access the system and selects to register 2. The customer enters in his name, username, password, email, address, credit card information |
| **Post-conditions:** | New User is created in the system and can be accessed by logging in |
| **Priority:** | 3 |
| **Alternative flows and exceptions:** |  |
| **Non-behavioral requirements:** | 1. Security measure to protect PII 2. Meet PCI (maybe using third party) |
| **Assumptions:** | Want to make sure users have a valid credit card when registering;  Want each user to be unique so we can identify |
| **Issues:** |  |
| **Source:** |  |

|  |  |
| --- | --- |
| **Use case name:** | LogIn |
| **Use case ID:** | 002 |
| **Primary actors:** | Customer, Reservation Manager |
| **Secondary actors:** |  |
| **Brief description:** | Customer and Reservation manager logs in using username and password and is navigated to the appropriate part of the system |
| **Preconditions:** | Customer must be registered, Reservation Manager must be authorized |
| **Flow of events:** | 1. The user enters username and password 2. The system validates the credentials against existing users |
| **Post-conditions:** | User is navigated to the appropriate screen based on type of user |
| **Priority:** | 4 |
| **Alternative flows and exceptions:** |  |
| **Non-behavioral requirements:** | Must validate credentials to make site secure |
| **Assumptions:** | Assume that manger has user creates when hired? |
| **Issues:** |  |
| **Source:** | Flight Reservation System |

|  |  |
| --- | --- |
| **Use case name:** | Search Flights |
| **Use case ID:** | 003 |
| **Primary actors:** | Customer |
| **Secondary actors:** |  |
| **Brief description:** | Customer can search available flights |
| **Preconditions:** | User wants to see only selected flights, all criteria must be selected (need departure and arrival destination and day) |
| **Flow of events:** | 1. The customer picks departure and arrival location, departure and arrival day plus passengers 2. The system returns flights meeting the criteria |
| **Post-conditions:** |  |
| **Priority:** | 1 |
| **Alternative flows and exceptions:** |  |
| **Non-behavioral requirements:** |  |
| **Assumptions:** | Flights keep track of available seats |
| **Issues:** |  |
| **Source:** | Flight Reservation System |

|  |  |
| --- | --- |
| **Use case name:** | Manage Itinerary |
| **Use case ID:** | 004 |
| **Primary actors:** | Customer |
| **Secondary actors:** |  |
| **Brief description:** | Customer creates a new itinerary |
| **Preconditions:** | To proceed to this step the user must be logged in and have a valid credit card |
| **Flow of events:** | 1. The customer chooses flight(s) for itinerary 2. The itinerary is marked as reserved 3. Email sent to user 4. The customer confirms and the credit card is charged 5. The itinerary is marked as booked 6. Email sent to user |
| **Post-conditions:** | The itinerary appears with its status on the customers itineraries |
| **Priority:** | 2 |
| **Alternative flows and exceptions:** | Alternative 1 (don’t finish booking)   1. The customer chooses flight(s) for itinerary 2. The itinerary is marked as reserved 3. If the flight is not booked in x time from reserving it is canceled 4. Notification sent to user   Alternative 2 (cancel reservation/boking)   1. The customer chooses flight(s) for itinerary 2. The itinerary is marked as reserved 3. The user cancels the reservation/booking 4. Email sent to user |
| **Non-behavioral requirements:** |  |
| **Assumptions:** | Flights have a way of keeping track of available seats from other systems booking? |
| **Issues:** |  |
| **Source:** | Flight Reservation System |

|  |  |
| --- | --- |
| **Use case name:** | Manage Flights |
| **Use case ID:** | 005 |
| **Primary actors:** | Reservation Manager |
| **Secondary actors:** |  |
| **Brief description:** | Reservation manager can create ,update, delete flights |
| **Preconditions:** | Reservation Manager needs to be logged in; flight needs to be unique |
| **Flow of events:** | 1. The manager navigates to user screen to add flight 2. Manager adds the flight details |
| **Post-conditions:** | Flight available in the system to search |
| **Priority:** |  |
| **Alternative flows and exceptions:** | Alternative 1 (update flight)   1. The manager navigates to user screen to a particular flight to update 2. Manager updates the flight details   Alternative 2 (delete flight)   1. The manager navigates to user screen to a particular flight 2. Manager deleted the flight details |
| **Non-behavioral requirements:** |  |
| **Assumptions:** | Reservation manager has a reliable source of flights to input |
| **Issues:** |  |
| **Source:** | Flight Reservation System |

|  |  |
| --- | --- |
| **Use case name:** | Generate Flight Report |
| **Use case ID:** | 006 |
| **Primary actors:** | Reservation Manager |
| **Secondary actors:** |  |
| **Brief description:** | Reservation Manager generates report |
| **Preconditions:** | Reservation manager must be logged in, there must be flights in the system |
| **Flow of events:** | 1. The manager selects report interface 2. The manager selects criteria for report |
| **Post-conditions:** | Report is generated |
| **Priority:** |  |
| **Alternative flows and exceptions:** |  |
| **Non-behavioral requirements:** |  |
| **Assumptions:** |  |
| **Issues:** |  |
| **Source:** | Flight Reservation System |