Flight Reservation System

Alysha Velasquez

Regis University

Flight Reservation System

The purpose of the Flight Reservation System Project is to design and implement a reservation system that will support customer and management functions. For customers, the system will allow flexibility and convenience for booking and managing flight reservations online. The customer will be able to login to the system, search for flights, create and book tickets, as well as cancel their itinerary. For the reservation manager, the system will allow them to add, search, or cancel flight information, and generate inventory reports. The system will be developed in Java and will be constructed using the Eclipse IDE. Unit tests will be applied using JUnit, packaging using Ant, and the UI front will utilize SWING.

**Use Cases**

The following sections specify the various interactions the customer and reservation manager have with the system.

**Use Case Diagram**

**A close up of a map

Description automatically generated**

**Manager Interacting Use Cases**

The list below specifies the manager interacting use cases ordered by priority.

1. Add Items– Adding items (i.e. flights, airplanes) to the management system will be the top priority. The pre-condition assumes the manager already has an account and is logged into the system. In the main success scenario, the manager will have a page for managing flights and airplanes.
2. Search Items – Search for items will be the second top priority. The pre-condition assumes the manager already has an account and is logged into the system. In the main success scenario, the manager will have a page for managing flights and airplanes. The manager will be able to search for flights by flight ID and airplanes by airplane ID.
3. Register – The manager will be able to create a new account. In the main success scenario, the manager provides their name, email, and password.
4. Manager Login – assumes the manager has a login and password. In the main success scenario, the manager can log in to the system with their username and password.
5. Delete items - The pre-condition assumes the manager already has an account and is logged into the system. In the main success scenario, the manager will have a page for managing items (i.e. flights, airplanes). The manager will be able to delete a flight or airplane from the inventory.
6. Generate inventory report – Generating an inventory report is a medium priority and will be implemented if time permits. The pre-condition assumes the manager already has an account and is logged into the system. In the main success scenario, the manager will go to a page to generate an inventory report. After, the manager will be able to view a summary of all flights in the reservation system.

**Customer Interacting Use Cases**

The list below specifies the customer interacting use cases ordered by priority.

1. Customer login – Customer login will be the first top priority after all management functions are completed. In the main success scenario, the customer enters their username and password to sign in. If the password or username is incorrect, the user will receive an error message.
2. Customer Register – A customer can register by providing their name, address, username, password, and credit card information (optional). The criteria for the credit card information will include a 16-digit card number and a four-digit expiration date.
3. Search flight availability – Searching for flight availability will be implemented after implementing customer registration and login. The pre-condition for this use case requires the customer to be logged in to the system. In the main success scenario, the customer will be able to search for flights by providing the departure date, time, and location, entering the number of passengers, and specifying whether they would like a one-way or roundtrip flight. The search results will show a list of itinerary options with departure and arrival times, and cost information. The itinerary options may also include one or more flights from one or more airlines. Each itinerary will be limited up to two people.
4. Reserve flight(s) – Reserving flights use case will be implemented depending on time dependency. The pre-condition for this use case requires the customer to be logged in to the system and assumes the customer has already searched for flights. In the main success scenario, the customer has the option to reserve a flight after selecting an itinerary from the search results. The customer will enter their necessary personal information for everyone in their party. After, the customer will choose their preferred seat type (i.e., business class, economy class) and will review their reservation.
5. Payment processing – Payment processing will also be implemented depending on time. The pre-condition for this use case requires the customer to be logged into the system and assumes the customer is already reserved a flight. In the main success scenario, the customer selects the option to book a flight and will be prompted to provide their credit card payment information. If successful, the customer will receive a conformation message. If the credit card information is invalid, the customer will receive an error message.
6. Cancel ticket reservation – Canceling a ticket reservation is the last use case and will be implemented if time permits. The pre-condition for this use case requires the customers to have an account and logged to the system. In the main success scenario, the customers will have a page where they will be able to review their reserved flights. The customer will be able to select the particular flight they want to cancel and confirm they would like to cancel their itinerary.

**Fully Dressed Use Cases for Manager Interacting Use Cases (Ordered by Priority):**

1. **Add Items:**

Primary Actor – Admin/Reservation Manager

Stakeholders – The manager needs to be able to manage flights and airplanes.

Success Guarantee – The manager is able to add a flight or airplane.

Main Scenario (Basic Flow):

1. The manager presses the “add” button.
2. The manager is prompted to enter the flight details and selects the “save” button.
3. System saves new flight information.
4. **Search Flight:**

Primary Actor – Admin/Reservation Manager

Stakeholders – The manager needs to be able to manage flights and airplanes

Success Guarantee – The manager is able to search for a flight or airplane.

1. The manager presses the “Search” button.
2. The manager is prompted enter a flight ID or Airplane ID.
3. The manager clicks “search”.
4. The system queries flight or Airplanes by ID.
5. **Register:**

Primary Actor – Reservation Manager

Stakeholders – The manager needs to be able to register to the system.

Success Guarantee – The manager is able to register to the system.

Main Scenario:

1. The manager presses the “register” button
2. The manager is prompted to enter their name, email, and password.
3. The system verifies if the input is correct and saves the information to the system.
4. **Login:**

Primary Actor – Reservation Manager

Stakeholders – The manager needs to be able to login to the system.

Success Guarantee – The manager is able to login to the system.

Main Scenario:

1. The manager presses the “login” button
2. The manager is prompted to enter their email and password.
3. The system verifies if the input is correct and saves the input into the system.
4. **Delete Item:**
5. The manager presses the “delete” button.
6. The manager selects the specific flight or airplane they would like to delete and presses the “remove” button.
7. The system removes the flight or airplane from the list.
8. **Generate Inventory Report**

Primary Actor – Reservation Manager.

Stakeholders – The manager needs to be able to generate an inventory report.

Success Guarantee – The manager is able to generate an inventory report.

Main Scenario (Basic Flow):

1. On the homepage the manager presses “Inventory Report.”
2. After the manger is able to view the inventory information on flights.

**Fully Dressed Use Cases for Customer Interacting Use Cases:**

1. **Login**

Primary Actor – Customer.

Stakeholders – The customer needs to be able to login to the system.

Success Guarantee – The customer successfully logs into the system.

Main Scenario (Basic Flow):

1. The customer has the option to press the “Login” or “Register” button.
2. If the customer chooses “Login,” they will be prompted to enter their email and password.
3. The system will verify if the input is correct. If correct, the customer will be directed to their homepage.
4. If student presses the “Register” button, they will be prompted to enter basic personal information (i.e., email, password, address, credit card number).
5. The customer presses “Submit” and will be directed back to login with their email and password.

Extension:

1. If the email and/or password is incorrect, the system will prompt an error message and will tell the user to try again or to a register an account.
2. **Search for flights**

Primary Actor – Customer

Stakeholders – The customer needs to be able to search for flights that match their preferences.

Success Guarantee – The customer is able to search flights that meets their needs.

Preconditions – Customer needs to be logged into the system.

Main Scenario (Basic Flow):

1. Upon logging in, the customer will be presented a form where they can search for flights by departure, arrival date, time and location, number of passengers, one-way or round trip. The customer provides their flight information based on these options.
2. The system will process the information.
3. A list of flight options is presented with departure/arrival and cost information.

**3. Reserve ticket**

Primary Actor – Customer

Stakeholders – The customer needs to be able to reserve a flight.

Preconditions – Customer needs to be logged into the system and have already searched for flights.

Main Scenario (Basic Flow):

* 1. The user selects an itinerary they would like to purchase by pressing the “Book” button.
  2. The itinerary will be place in a shopping cart in the system.
  3. During the checkout process, the customer will be promoted to pick between economy or business class. The customer selects their seat preference and selects “Next.”
  4. After the customer is prompted a form requesting basic personal information on themselves and any passengers that will be joining them. The customer provides the requested information and selects “Next.”
  5. The customer reviews their order before proceeding to the payment process.

**4. Payment**

Primary Actor – Customer

Stakeholders – The customer needs to be able to pay for a flight itinerary.

Preconditions – Customer needs to be logged into the system and have already reserved a flight.

Main Scenario (Basic Flow):

1. After reviewing their order, the customer selects “Pay” to initiate the payment process.
2. A form will be promoted to provide their credit card information or select a credit that is already saved to their profile (optional).
3. After providing the requested information, the customer selects “Confirm” to process the payment.
4. The system will verify if the credit card information is valid. If valid, then they will receive a confirmation message.

Extension:

1. If the customer the credit card information is incorrect, the user will be prompted to try again or use another credit card for payment.

**5. Cancel flight**

Primary Actor – Customer

Stakeholders – The customer needs to be able to cancel a flight reservation.

Preconditions – Customer needs to be logged into the system.

1. On the customer’s home profile page, the customer has the option to select “Manage Reservations.” After selecting this option, the customer will be able to see a list of flights they have reserved.
2. The customer selects the specific flight they would like to cancel and selects “Cancel Reservation.”
3. The customer receives a confirmation message confirming their reservation has been canceled.

**Summary**

This document details the requirements and use cases for the Flight Reservation System. The project will prioritize management functions. Afterwards, the project will prioritize the customer functions.