Software Requirements and Design Document

for

Airport Management System

**Prepared by Affan Amer, Mobeen Hasan, Aman Asif**

**NextGen Solutions**

**26/11/2024**

**Table of Contents**

**Table of Contents ii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 Product Scope 1

1.3 Title 1

1.4 Objectives 1

1.5 Problem Statement 1

**2.** **Overall Description 1**

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 List of Use Cases 2

2.4 Extended Use Cases 2

2.5 Use Case Diagram 2

**3.** **Other Nonfunctional Requirements 2**

3.1 Performance Requirements 2

3.2 Safety Requirements 2

3.3 Security Requirements 2

3.4 Software Quality Attributes 2

3.5 Business Rules 3

3.6 Operating Environment 3

3.7 User Interfaces 3

**4.** **Domain Model 3**

**5.** **System Sequence Diagram 3**

**6.** **Sequence Diagram 3**

**7.** **Class Diagram 4**

**8.** **Package Diagram 4**

**9.** **Deployment Diagram 4**

# Introduction

## Purpose

*The Airport Management system is designed to streamline airport operations, particularly flight booking and management for passengers and administrative functions for the airport staff. This airport management system covers the functionalities for both passengers and administrators, providing a unified platform for booking, scheduling, and airline management.*

## Product Scope

*The Airport Management System (AMS) is a JavaFX-based Desktop software application integrated with an SQL Server database to manage airport operations effectively.*

## Title

*Airport Management System (AMS)*

*This project aims to provide an efficient solution for managing airport operations, including passenger flight bookings, boarding pass generation, and administrative functions such as airline and flight management, passenger oversight, and revenue reporting. It bridges the gap between passengers and airport administration for smooth, automated operations.*

## Objectives

*The primary objectives of the Airport Management System are:*

1. *To provide passengers with an intuitive and user-friendly interface for managing their flight bookings.*
2. *To reduce manual tasks for administrators by automating flight scheduling, airline management, and reporting.*
3. *To enhance operational efficiency by maintaining accurate records of passengers, bookings, and revenues.*
4. *To improve customer satisfaction by enabling smooth and fast airport processes.*

## Problem Statement

*Airports face significant challenges in managing manual operations, which can lead to inefficiencies, errors, and reduced customer satisfaction. The absence of a centralized system results in delays in passenger services and administrative tasks.*

# Overall Description

## Product Perspective

*The Airport Management System (AMS) is a new, self-contained software product designed to automate and modernize airport operations. It does not depend on any pre-existing systems but can serve as a foundation for integrating additional modules or extending functionalities in the future.*

*System Overview:*

*The AMS operates as an independent system consisting of two user categories:*

1. *Passengers: Who interact with the system through the user-friendly JavaFX interface to perform actions like booking flights, generating boarding passes, and viewing flight details.*
2. *Administrators: Who utilize admin tools to manage airlines, flights, and passenger records and generate revenue reports.*

*The system is built on a client-server architecture:*

* *Client-Side: Developed using JavaFX for an intuitive graphical interface.*
* *Server-Side: Uses SQL Server to store and manage all data.*

## Product Functions

*The Airport Management System (AMS) supports a wide range of functionalities organized for two primary user roles:*

*Passenger Functions:*

* *Sign Up and Log In: New passengers can register and log into their accounts.*
* *Search Flights: Browse available flights based on date, destination, and airline.*
* *Book Flights: Reserve seats on selected flights.*
* *View Flights: Review booked flights, including details like time and seat number.*
* *Generate Boarding Pass: Create a digital boarding pass for booked flights.*
* *View Boarding Pass: Access and print boarding passes when needed.*

*Admin Functions:*

* *Manage Airlines: Add, view, or remove airlines.*
* *Manage Passengers: View passenger details, delete accounts, or suspend users as needed.*
* *Manage Flights: Add new flights, reschedule existing ones, or cancel flights.*
* *Generate and View Revenue Reports: Track and analyze revenue generated from bookings.*

## List of Use Cases

1. Search Flight

2. Book Flight

3. View Flights

4. Manage Airlines

5. Reschedule Flights

6. Schedule Flights

7. Manage Users

8. Cancel flight

9. Create Boarding Pass

10. View Boarding Pass

11. Add Flight

12. View Revenue Report

13. Create Revenue Report

## Extended Use Cases

**Use case name:** Search Flight

**Scope:** User level goal

**Primary Actor:** Passenger

**Stakeholders and Interests:**

Passenger – Wants to find available flights according to their requirements and needs.

Airport Management System – Wants to ensure that all the flight data is available.

**Preconditions:**

Flight data is available in the airport management system.

The passenger has an active connection with the system to view the flight data.

**Success guarantee (postcondition):**

A list of flights matching the passenger's search criteria is displayed.

**Main Success Scenario:**

Actor Action (or Intention)

1. Passenger selects the option to search for flights.

3. Passenger enters the search criteria and submits the request.

5. Passenger views the flight details.

System Responsibility

2. The system prompts the passenger to enter search criteria (e.g., destination, date, time, airline, price).

4. The system searches the flight database and displays the matching flights.

**Extensions:**

2a. Passenger enters invalid or incomplete search criteria:

2a1. The system notifies the passenger to correct or complete the input.

4a. No flights are found with the matching criteria:

4a1. The system notifies the passenger that no flights were found with the entered criteria and offers the passenger to modify their seaUse case name: Search Flight

Scope: User level goal

Primary Actor: Passenger

Stakeholders and Interests:

Passenger – Wants to find available flights according to their requirements and needs.

Airport Management System – Wants to ensure that all the flight data is available.

Preconditions:

Flight data is available in the airport management system.

The passenger has an active connection with the system to view the flight data.

Success guarantee (postcondition):

A list of flights matching the passenger's search criteria is displayed.

Main Success Scenario:

Actor Action (or Intention)

1. Passenger selects the option to search for flights.

3. Passenger enters the search criteria and submits the request.

5. Passenger views the flight details.

System Responsibility

2. The system prompts the passenger to enter search criteria (e.g., destination, date, time, airline, price).

4. The system searches the flight database and displays the matching flights.

Extensions:

2a. Passenger enters invalid or incomplete search criteria:

2a1. The system notifies the passenger to correct or complete the input.

4a. No flights are found with the matching criteria:

4a1. The system notifies the passenger that no flights were found with the entered criteria and offers the passenger to modify their search criteria.

rch criteria.

**Special Requirements:**

Search functionality should be accessible on both desktop and mobile devices.

Filters should be available in the system to allow to filter out flights.

**Technology and data variations list:**

Passenger may use a mobile/desktop or the web browser to search for a flight.

User may enter the search criteria either using a keyboard or a touch screen.

**Use case name:** Book Flight

**Scope:** User level goal

**Primary Actor:** Passenger

**Stakeholders and Interests:**

Passenger – Wants to book a flight based on their requirements.

Airport Management System – Wants to ensure that the flight is booked successfully.

Admin – Track and store the cost of the booked ticket.

Payment Vendors – Provide their payment services to the app.

**Preconditions:**

Passenger is logged in the system.

Passenger has successfully selected a flight to book.

The selected flight has available seats.

**Success guarantee (postcondition):**

The flight is successfully booked, and the customer is sent a confirmation. The system adds the flight details and ticket price to the database. Admin adds the cost of the flight to the corresponding airline.

**Main Success Scenario:**

Actor Action (or Intention)

1. Passenger selects a flight to book.

3.  Passenger selects a seat and proceeds to payment option.

5. Passenger receives a booking confirmation with their flight details.

System Responsibility

2. The system displays available seats alongside their prices.

4. The system process the payment, confirms the booking and updates its database.

**Extensions:**

2a. The flight is fully booked during the transaction:

2a1. The system notifies the passenger that the flight has been fully booked and offers them other flights to choose from.

4a. The payment fails to go through

4a1. The passenger is notified with an error message and is prompted to choose another payment option.

**Special Requirements:**

The payment should be secure and use TLS certificate if using the web version of the app.

Passengers should receive a confirmation message within 2 minutes of booking a flight.

**Technology and data variations list:**

Passenger may use a mobile/desktop or the web browser to book for a flight.

Passenger may use a credit card, debit card or a digital wallet to book a flight.

The booking page should include a QR code for quick payment.

**Use case name:** View Flights

**Scope:** User level goal

**Primary Actor:** Passenger

**Stakeholders and Interests:**

Passenger- Wants to view all of his current/ previous flights

**Preconditions:**

Passenger is logged in

**Success guarantee (postcondition):**

All of passenger flights are displayed in table form on the screen.

**Main Success Scenario**

Actor Action (or Intention)

1. Passenger logs in

2. Passenger clicks MyFlights button

System Responsibility

3. The system uses passengers ID and list all his flights using the database

**Extensions:**

No flights were found for the user in the database.

**Use case name:** Manage Airlines

**Scope:** User level goal

**Primary Actor:** Admin

**Stakeholders and Interests:**

Admin – Should be able to add/view/update/ delete airlines from the system.

Airline – Wants to successfully register/update with the airport management system.

**Preconditions:**

Admin is logged in the system.

Airline to be managed has its data available.

**Success guarantee (postcondition):**

Airlines are successfully registered or updated in the system and compliant with regulations.

**Main Success Scenario:**

Actor Action (or Intention)

1. Admin selects the manage airlines option from the system.

3.  Passenger selects an airline to register or update.

5. Admin enters the required details into the system.

System Responsibility

2. The system displays all the available airlines alongside the CRUD operations.

4. The system presents the interface to enter details to register an airline or details to manage the airline.

6.  The system confirms the changes and updates the airline’s data.

**Extensions:**

3a. Admin tries to register an airline that is already registered.

3a1. The system notifies the admin that the airline selected is already registered with the system.

5a. The admin enters incorrect/inadequate details

5a1. The admin is notified to enter the complete and correct details.

**Special Requirements:**

Only authorized admins should be able to manage the airlines.

**Technology and data variations list:**

Admin may use a web based or desktop/mobile app to manage the airlines.

Admin should verify themselves using a fingerprint scanner each time they try to push a change to the system. 

**Use case Name:** Reschedule flight

**Scope**: User Level Goal

**Primary Actor:** Admin, Airline representative

**Stakeholders and Interests:**

            Passenger – Must be aware of rescheduling of their flight

Airport Management System – The updating of flights is important to avoid misunderstandings and correcting the schedule

Airplane staff – Are affected as they need to know about timing changes

Airline – Need the rescheduling

**Preconditions:**

          Flight needs to take place somewhere in the future.

          Flight data is present in the airport management system.

          There has been some kind of issue leading to a reschedule.

**Postconditions:**

    The date and time of the flight is reassigned and updated within the database. A notice is sent to the      flight passengers.

**Main Success Scenario**

**Actor Action                                                                                           System Responsibility**

1. Airline representative selects flight reschedule

2. Displays all of the airline’s upcoming flights.

3. Airline representative selects a flight and enters

the new date, time and plane number.

  4. Sends a reschedule ticket to admin.

5. Admin views ticket and clicks on reschedule flight.

6. System displays the flights that are to occur.

7. Admin enters the flight number.

8. System fetches and displays the information relevant to the flight and prompts to enter information.

8. Admin enters new date, time and necessary

flight details.

9. The flight information is updated                       accordingly in the database.

10. An automated confirmation is sent to all passengers of the flight and the airline representative.

**Extensions:**

9a. The runway has traffic at the given time, or no planes are available from the airline.

      9a1. The system states the unavailability and asks for a new date/time.

      9a2. The admin requests a new date and time from the airline representative and tries again.

9b. The weather conditions are harsh and unacceptable.

     9b1. The system states the weather conditions and asks for a new date/time.

     9b2. The admin requests a new date and time from the airline representative and tries again.

**Special Requirements:**

1. Authorization is granted

2. Coordination with the airline

3. Gate and runway allocation

4. The weather conditions are suitable

5. The new date is not too far apart from the current

**Technology and data variations list:**

1. Cloud storage

2. SQL database

3. Keyboard and mouse

4. Authentication System (fingerprint scanner) for admins

**Use case Name:** Manage User

**Scope**: User Level Goal

**Primary Actor:** Admin

**Stakeholders and Interests:**

            Admin- Want to manage users (add new users, delete them or suspend/unsuspend them)

**Preconditions:**

         Admin is logged in

         User exists

**Postconditions:**

        User is added/removed/suspended successfully

**Main Success Scenario**

**Actor Action                                                                                           System Responsibility**

1. Admin clicks View Users

2. System displays all of the users

3. Admin then selects 1 of the options. To either add,

Remove or suspend/unsuspend user.

4. Admin then enters user details

4. System confirms user management

**Extensions:**

User not found.

Incorrect/Incomplete user information entered

**Use case Name:** Cancel flight

**Scope**: User Level Goal

**Primary Actor:** Passenger

**Stakeholders and Interests:**

            Passenger – Needs to cancel their flight for an unknown reason.

Airport Management System – Data validity is important, so needs to remove the booking for the passenger and allocate empty seat(s).

Airline – Needs to know about a new vacant spot in their flight.

**Preconditions:**

Cancellation is requested within the cancellation duration

          Flight needs to take place somewhere in the future

          Flight data is present in the airport management system.

          Passenger must be booked for the flight

**Postconditions:**

     The booking for the passenger is cancelled, and they as well as the airline are notified. Passenger is    offered a refund if within the specified refund duration.

**Main Success Scenario**

**Actor Action                                                                                           System Responsibility**

1. Passenger selects the cancel flight option.

2. System displays the passenger’s upcoming flights.

3. Passenger chooses the desired flight.

4. System prompts a confirmation and bank details in case of an approved reimbursement, then forwards the request to an admin.

5. Passenger enters the information and confirms.

6. Admin verifies the date and time of the flight and

Checks that it’s within the refund period.

7. The passenger’s flight booking is erased from the database.

8. (if applicable) The refund is processed and the passenger is reimbursed the charged sum.

9. An automated confirmation is sent to the passenger and the airline.

**Extensions:**

6a. The refund duration has expired.

      2a1. The admin notifies the passenger that the refund duration has expired and that they can still

       cancel their flight.

     2a2. The passenger goes ahead and cancels the flight.

     2a2. Admin provides a 20% discount on the next flight to the passenger and then proceeds to cancel.

8a. The refund could not be processed due to incorrect bank details provided.

    8a1. The passenger is contacted directly and asked to send correct details for the refund.

**Special Requirements:**

1. Authorization is granted

2. There is sufficient cash in the refund card to successfully pay for the refund.

3. Passenger’s credit card has not reached full capacity in case of an approved refund.

**Technology and data variations list:**

1. Passenger may use a mobile/desktop or the web browser to cancel the flight.

2. Auto payment system

3. Authentication System (fingerprint scanner) for admins

**Use case Name:** Schedule flight

**Scope**: User Level Goal

**Primary Actor:** Admin

**Stakeholders and Interests:**

            Passenger/Other – Want their flight to be scheduled on time.

Admin- Want efficient scheduling of flights.

Airline – Check there flights schedule.

**Preconditions:**

The system has all the necessary data of the flight.

          Admin is logged in.

**Postconditions:**

     Flights are successfully scheduled, and delivered to passengers and airlines.

**Main Success Scenario**

**Actor Action                                                                                           System Responsibility**

1. Admin Clicks New Flight

2. System prompts to enter the flight number.

3. Admin enters the flight number.

4. System confirms that a flight with that number exists and shows its details

 5.Admin verifies the details and enters scheduling

information.

                                                                                                                  6.Systems schedules flight

**Extensions:**

4a. The flight does not exist.

* The flight does not have necessary equipment to schedule flight

**Use Case Name**

Create Boarding Pass

**Scope**

Airport Management system

**Level**

User goal level

**Primary Actor**

Passenger

**Stakeholder and Interests**

Passenger - Wants a boarding pass to enter the airport

System- Checks if a boarding pass is created and also to a valid booking

**Preconditions**

Passenger has booked an existing flight

**Post Conditions**

Boarding pass is generated.

**Main Success Scenario**

|  |  |
| --- | --- |
| Actor action | System |
| 1. Passenger logs in |  |
| 2. Passenger clicks the create boarding pass button |  |
|  | 3. System opens the myflightpanel and prompts to enter flightID |
| 4. Passenger enters FlightID |  |
|  | 5. System Verifies if the flight exists |
| 6. Admin enters the code in the box | 6.Once verified it generates a boarding pass |

**Extensions**

Wrong FlightID inserted

**Use Case Name**

View Boarding Pass

**Scope**

Airport Management system

**Level**

User goal level

**Primary Actor**

Passenger

**Stakeholder and Interests**

Passenger - Quick access to boarding information

**Preconditions**

Passengers have checked-in

**Post Conditions**

Passenger viewed boarding pass

**Main Success Scenario**

|  |  |
| --- | --- |
| Actor action | System |
| 1 .Passenger clicks “My Flights” |  |
|  | 2. System displays list of Passenger flights |
| 3 .Passenger selects the flight |  |
|  | 4. System displays boarding pass and other information |

**Extensions**

Flight cannot be found

System is down

**Use Case Name**

Add Flight

**Scope**

Airport Management system

**Level**

User goal level

**Primary Actor**

Admin

**Stakeholder and Interests**

Admin- Wants to add new flight so passengers can book it.

**Preconditions**

Admin is logged in

**Post Conditions**

Successful addition of flight

**Main Success Scenario**

|  |  |
| --- | --- |
| Actor action | System |
| 1 .Admin Clicks AddFlight button |  |
|  | 2. System prompts to enters all of the required information |
| 3 .Admin enters the information |  |
|  | 4. System verifies the information(Such as if the airline of the flight exists) |
|  | 5. System adds the flight |

**Extensions**

Incorrect/Incomplete information is given

**Use Case Name**

View Revenue Report

**Scope**

Airport Management system

**Level**

User goal level

**Primary Actor**

Admin

**Stakeholder and Interests**

Admin- Wants to view the revenue

**Preconditions**

Admin is logged in

**Post Conditions**

Successful display of revenue of all flights

**Main Success Scenario**

|  |  |
| --- | --- |
| Actor action | System |
| 1 .Admin Clicks Revenue Button |  |
|  | 2. System displays all of the flights with their revenue |

**Extensions**

Fails to retrieve flights

**Use Case Name**

Create Revenue Report

**Scope**

Airport Management system

**Level**

User goal level

**Primary Actor**

Admin

**Stakeholder and Interests**

Admin- Wants to Generate a revenue report, to be able to send or print it.

**Preconditions**

Admin is logged in

**Post Conditions**

Successful creation of revenue report of all flights

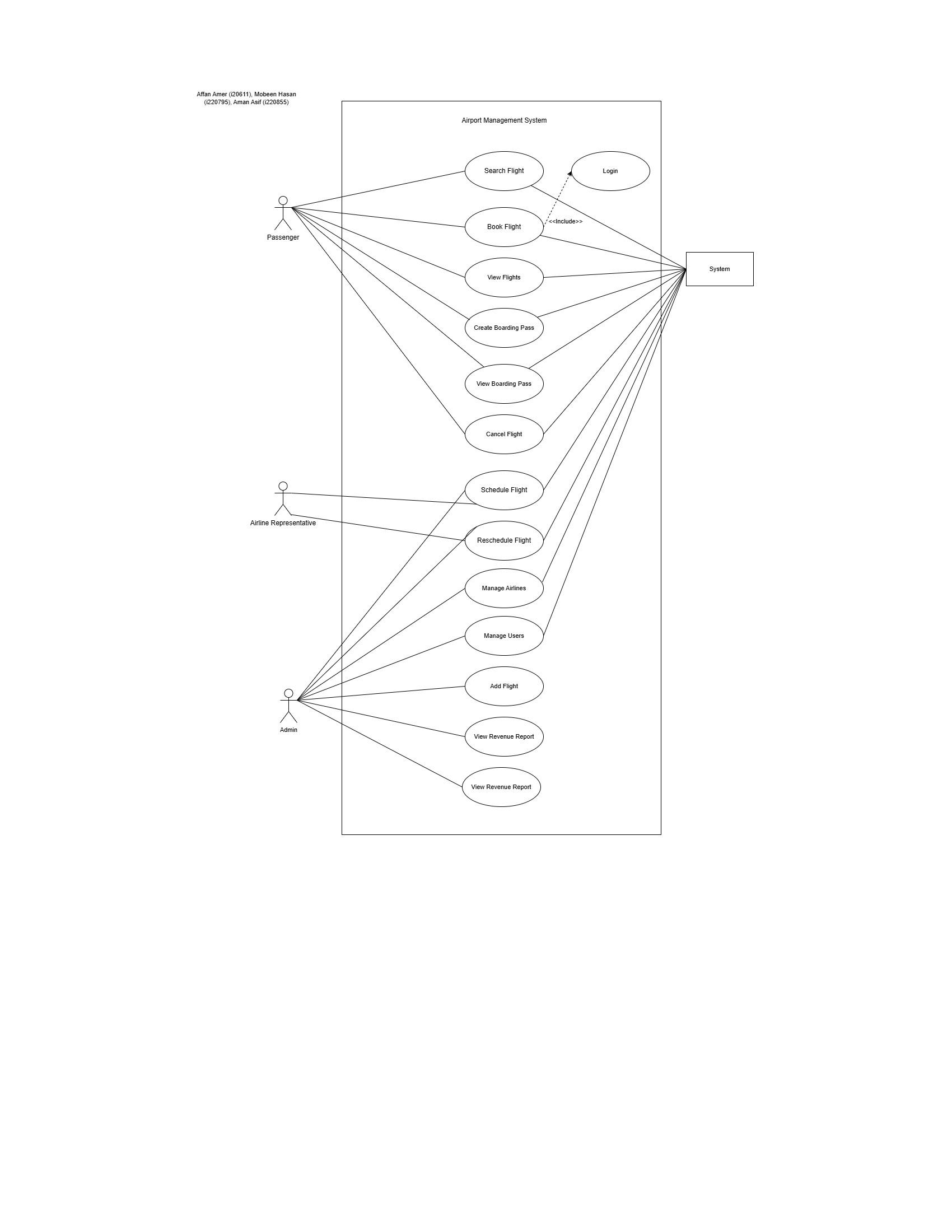
**Main Success Scenario**

|  |  |
| --- | --- |
| Actor action | System |
| 1 .Admin Clicks Revenue Button |  |
|  | 2. System displays all of the flights with their revenue |
| 3. Admin clicks button to generate revenue |  |
|  | 4. System generate a report |

**Extensions**

Fails to retrieve flights

## Use Case Diagram



# Other Nonfunctional Requirements

## Performance Requirements

* *The system must handle logins from at least 100 users without degradation in performance.*
* *Search functionality for flights should retrieve results in under 2 seconds in an average case.*
* *The generation of revenue reports should complete within 5 seconds for an average search case.*

***Rationale:***

*These requirements ensure a smooth user experience and operational efficiency.*

## Safety Requirements

* *Any operation that involves deletion (e.g., deleting a passenger or airline) must prompt the user for confirmation to prevent accidental data loss.*
* *User data should be saved inside a database that doesn’t have any unauthorized access.*
* *Only an authorized admin should be able to handle sensitive data such as user info.*
* *Access control mechanisms must ensure that only authorized users perform critical actions (e.g., rescheduling flights or generating revenue reports).*

## Security Requirements

*Authentication:*

* *Passengers must register with a valid username and password to access the system.*

*Authorization:*

* *Different user roles (Passenger, Admin) must have distinct permissions.*
* *Admin operations should be restricted to authorized personnel only.*

## Software Quality Attributes

* ***Usability:*** *The user interface must be intuitive and require minimal training for passengers and admins.*
* ***Correctness:*** *The system must ensure accurate flight schedules, bookings, and revenue calculations.*
* ***Maintainability:*** *The codebase must follow clean coding principles to make maintenance and updates straightforward.*
* ***Reliability:*** *Data consistency must be guaranteed even in the event of unexpected errors or server failures.*

## Business Rules

* ***Passenger Accounts****: Passengers can book, view, and cancel their own flights but cannot modify flight schedules or airline details.*
* ***Admin Roles****: Admins can manage all aspects of the system, including suspending passengers or rescheduling flights.*
* ***Revenue Reports:*** *Only admins can generate and view revenue reports.*
* ***Booking Restrictions****: A flight booking must be made before departure.*
* ***Passenger Suspension:*** *Suspended passengers cannot login*

## Operating Environment

***Database:***

* *SQL Server 2019 or newer.*

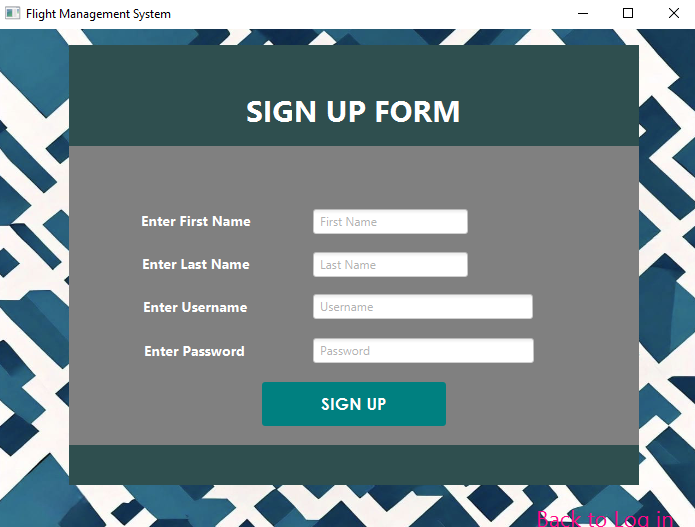
***Other Components:***

* *Java JDK 17 for the client application.*
* *JavaFX for the GUI.*

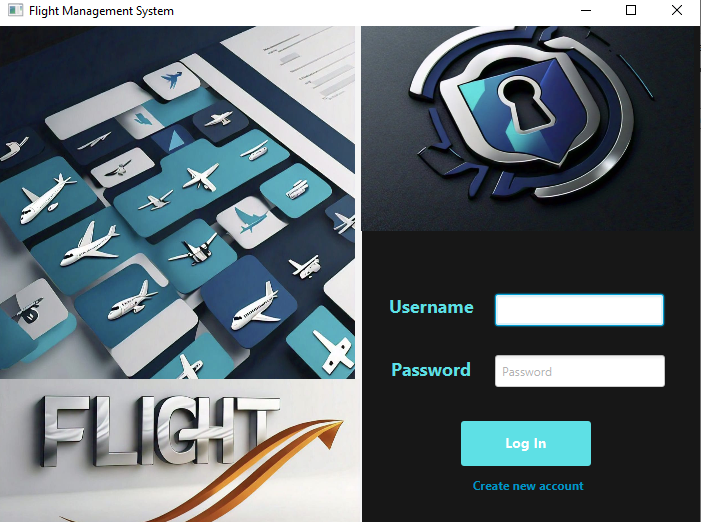
## User Interfaces

***Passenger Interface:***

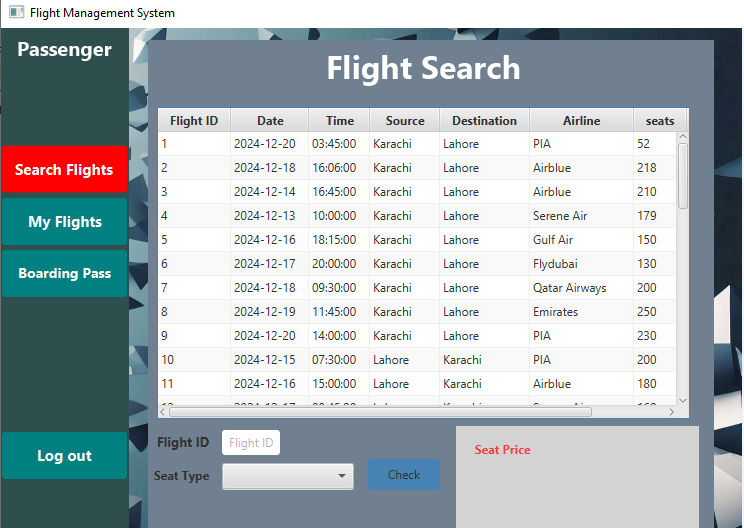
***Signup Page:*** *Simple page with fields for first name, last name, username, and password.*

**

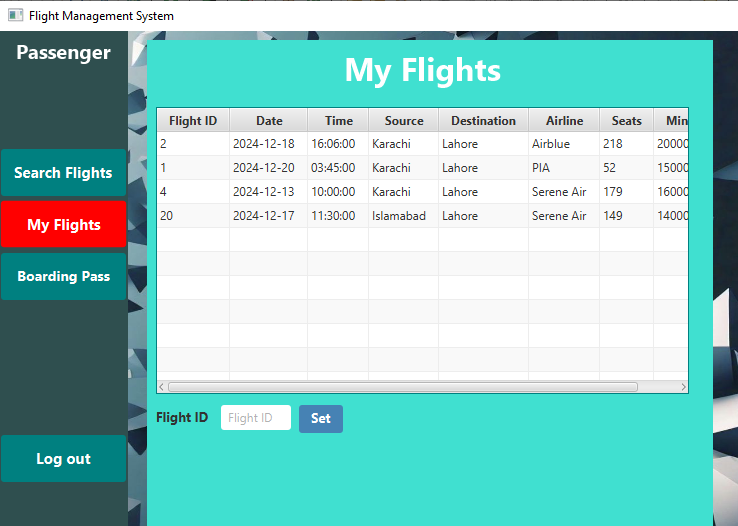
***Login Page:*** *Clean and straightforward, with fields for username and password.*

**

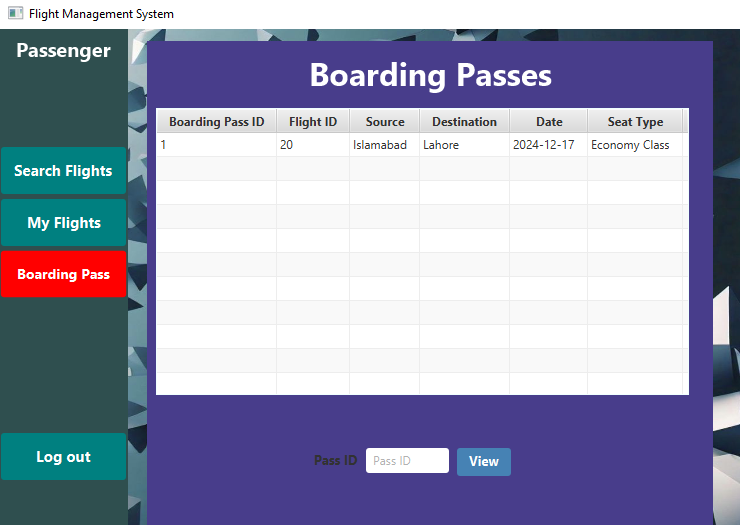
***Flight Search Page:*** *Fields for entering flight id or searching by seat type.*

**

***My Flight Page:*** *Displays information about the flights belonging to the logged in passenger.*

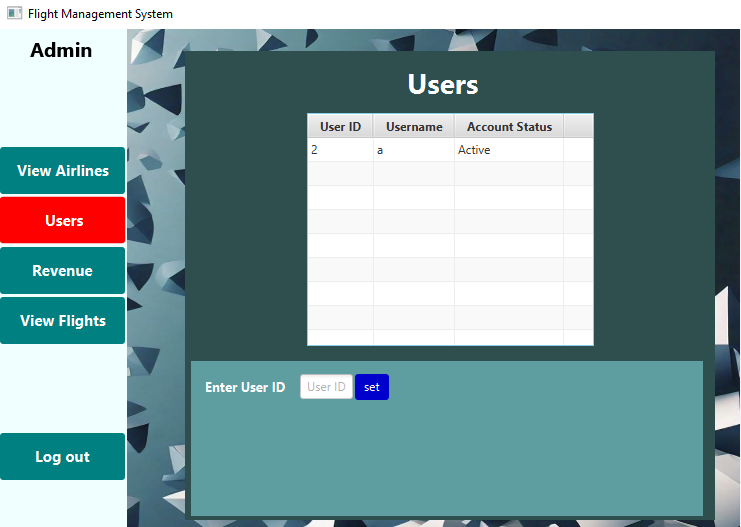
**

***Boarding Pass Page****: Includes boarding pass ID, flight ID, source, destination, date, and seat type. Includes a field to view boarding pass by ID and also create a boarding pass.*

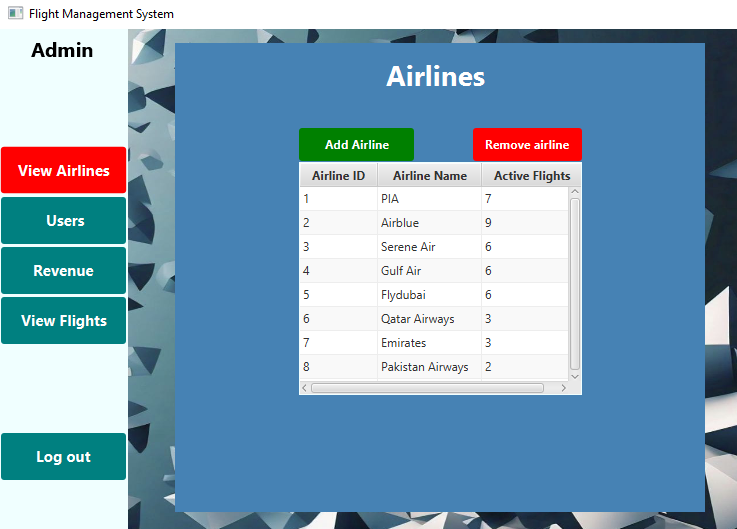
**

***Admin Interface:***

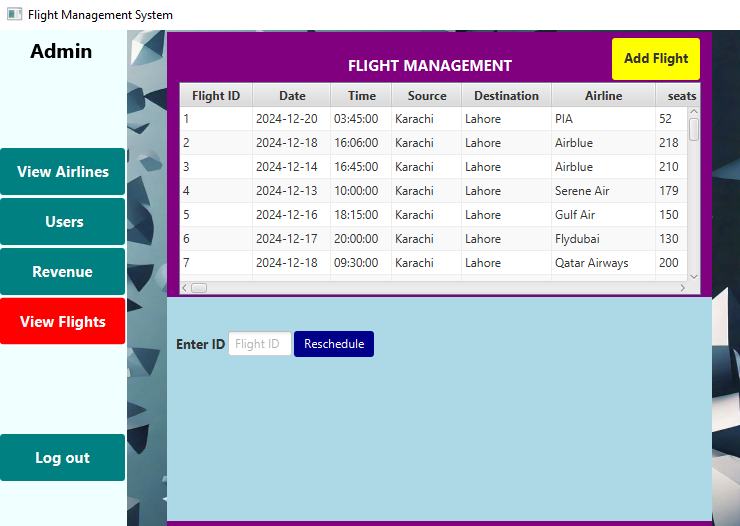
***User Dashboard****: User dashboard for managing users.*

**

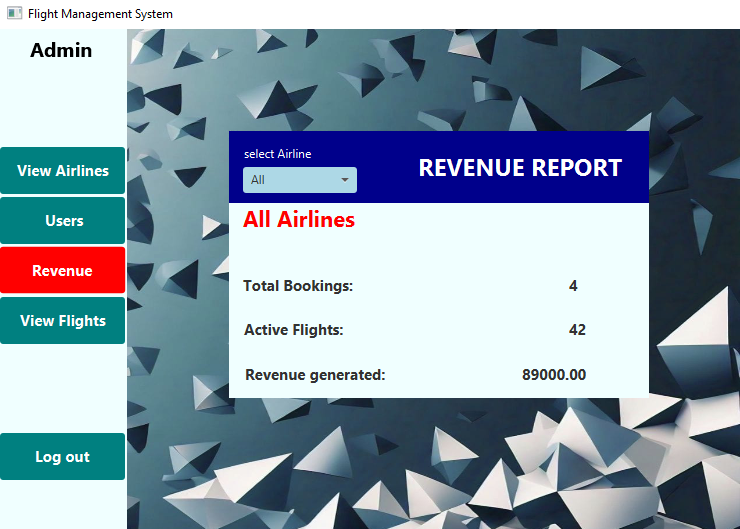
***Airline Management Page****: Page for managing airlines.*

**

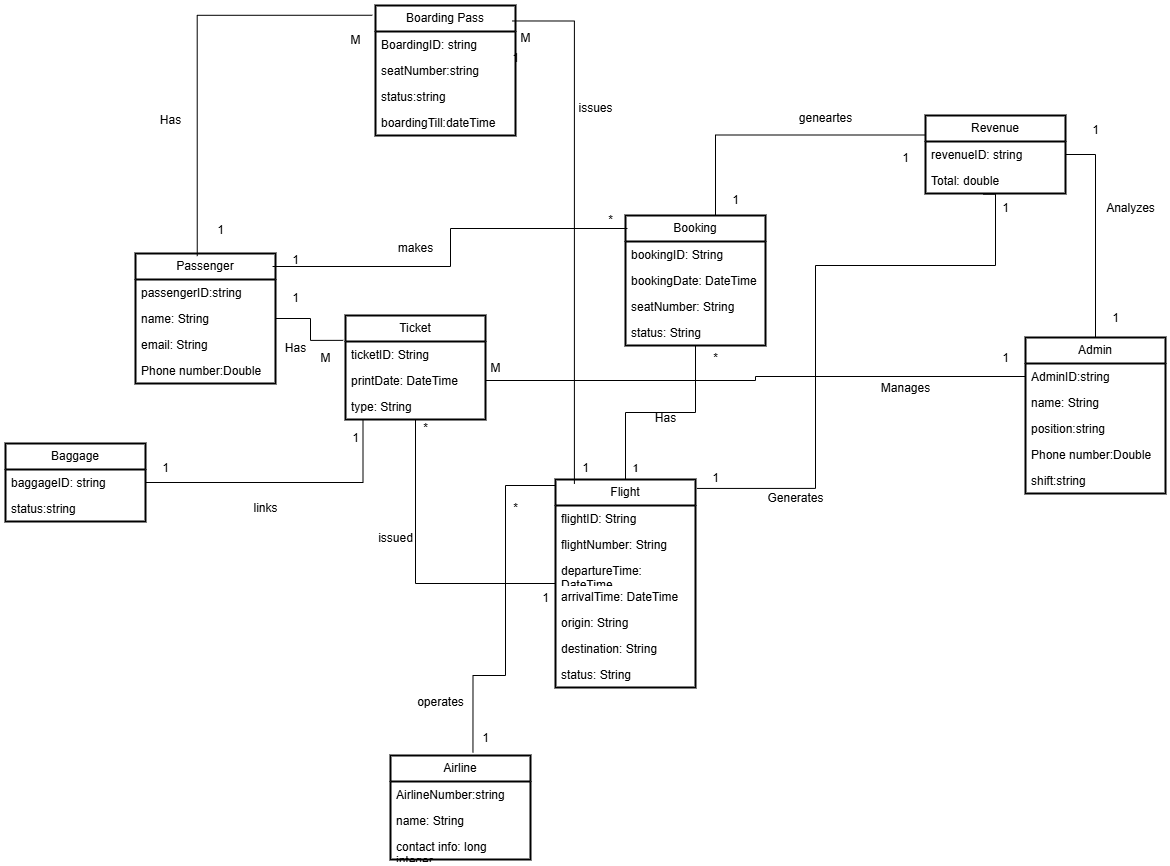
***Flight Management Page****: A page for managing all the flights and the operations associated with those flights.*

**

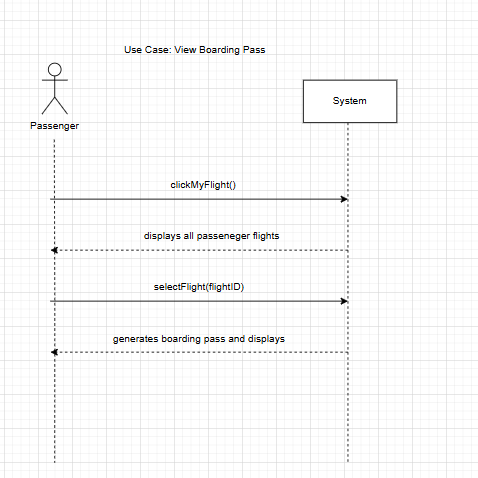
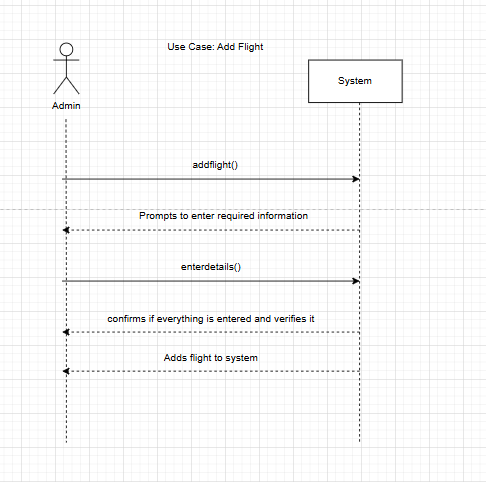
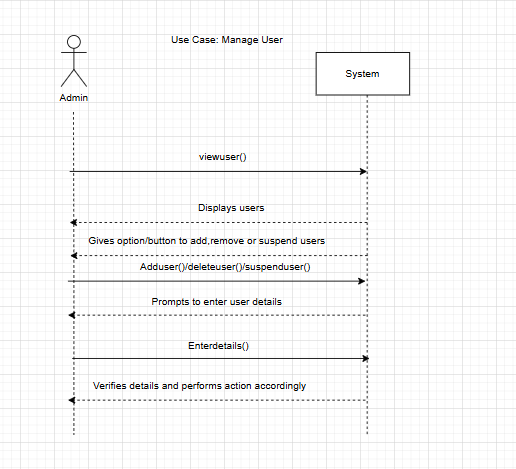
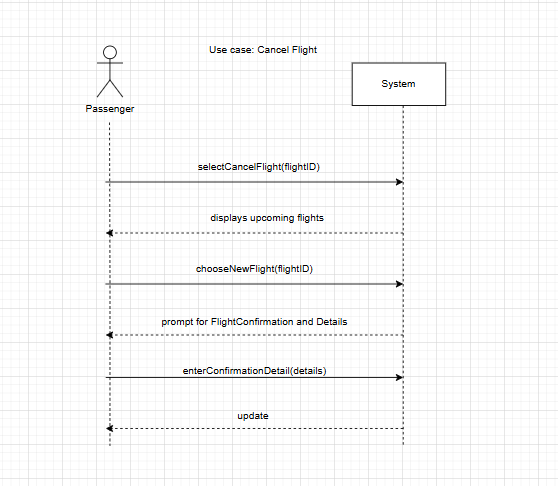
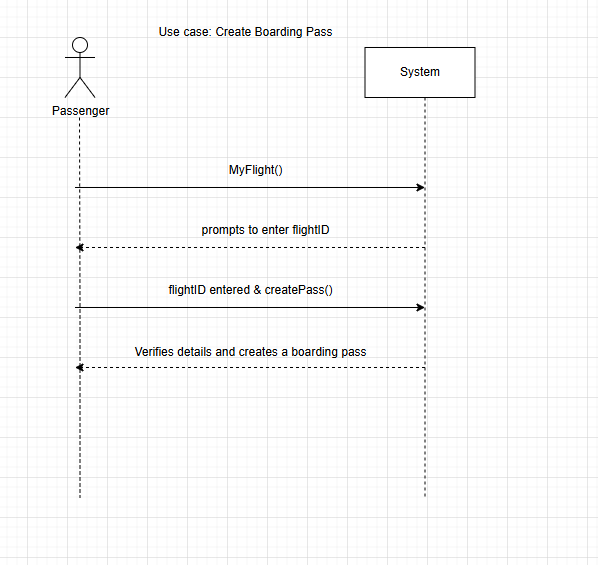
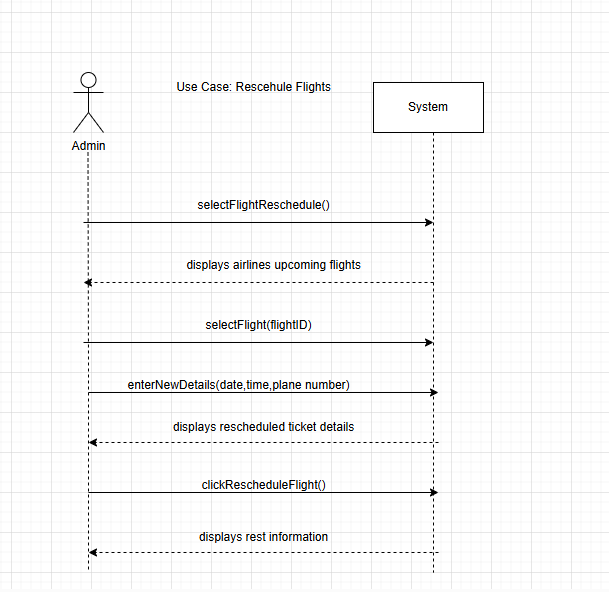
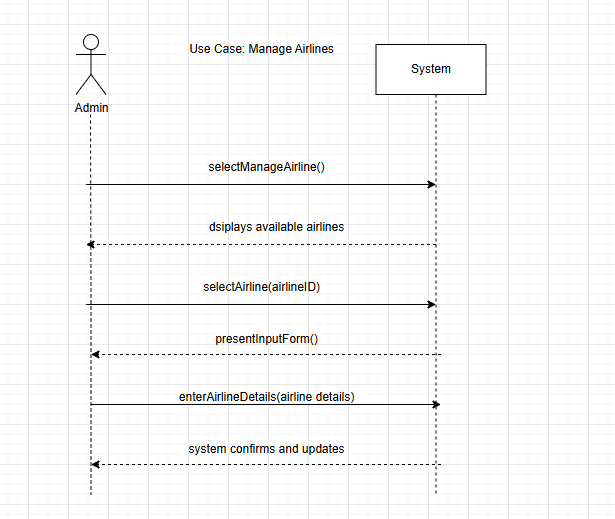
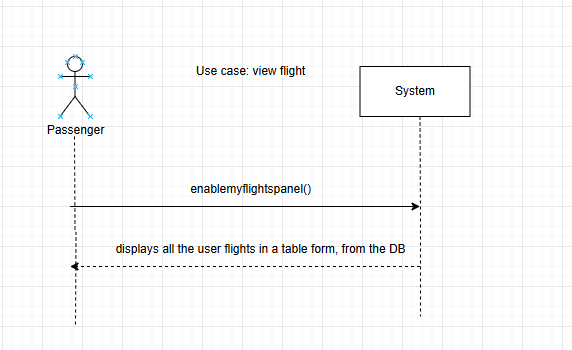
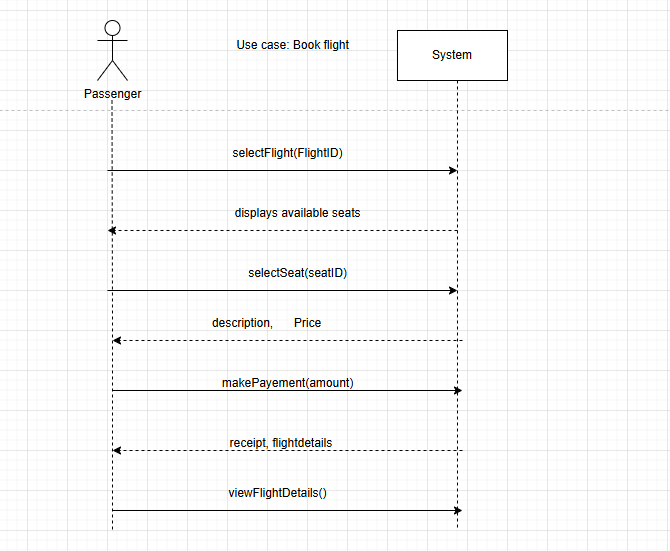
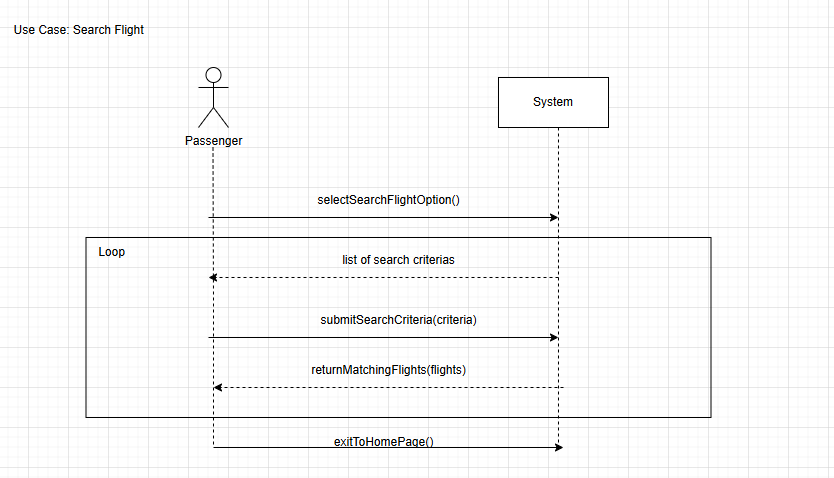
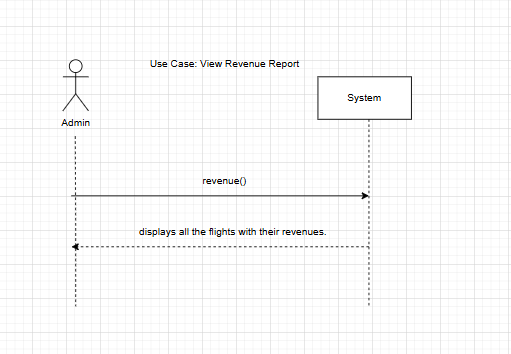
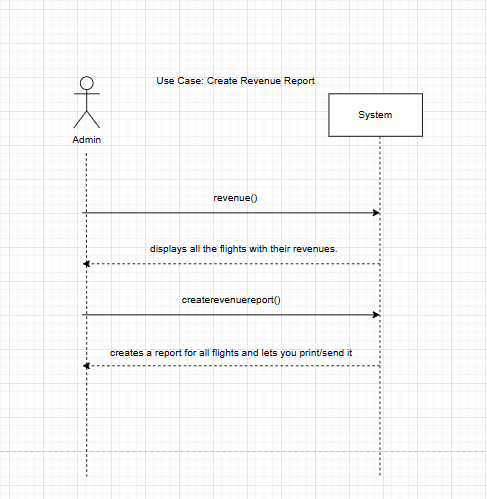
***Revenue Reports Page****: A page for generating and displaying the revenue report*

**

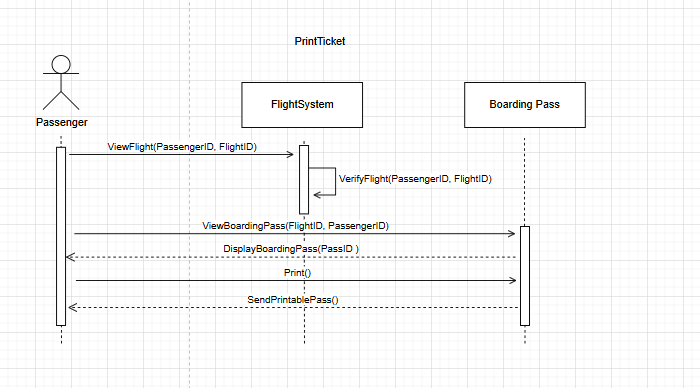
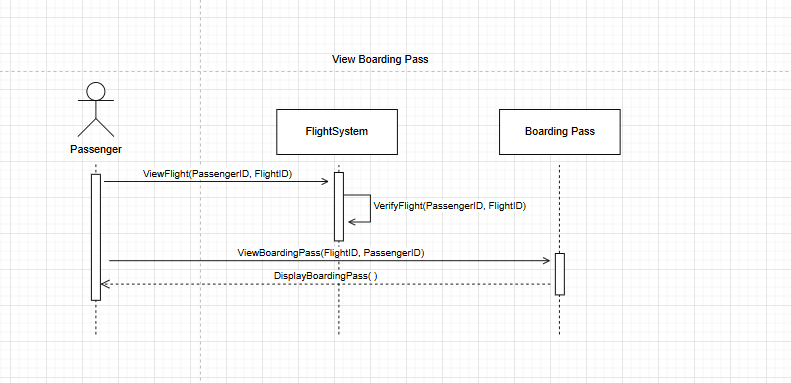
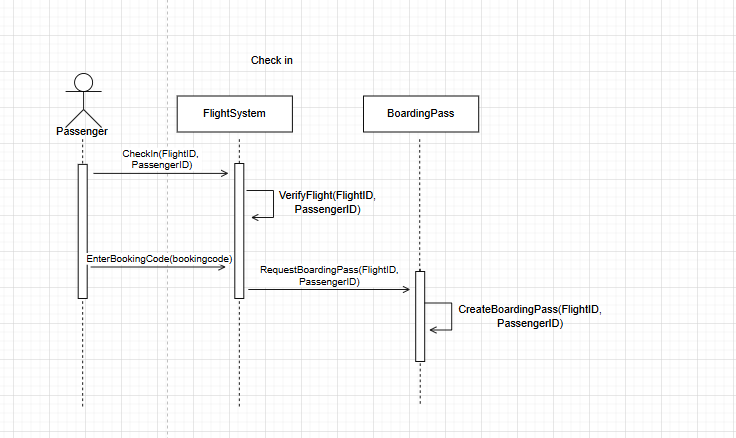
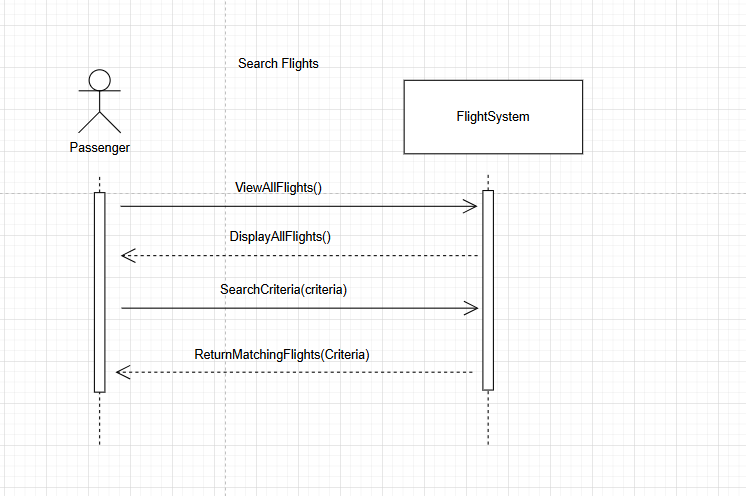
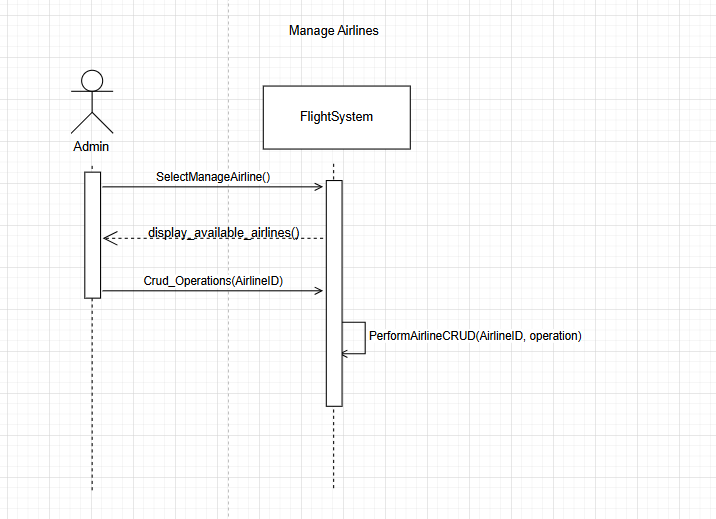
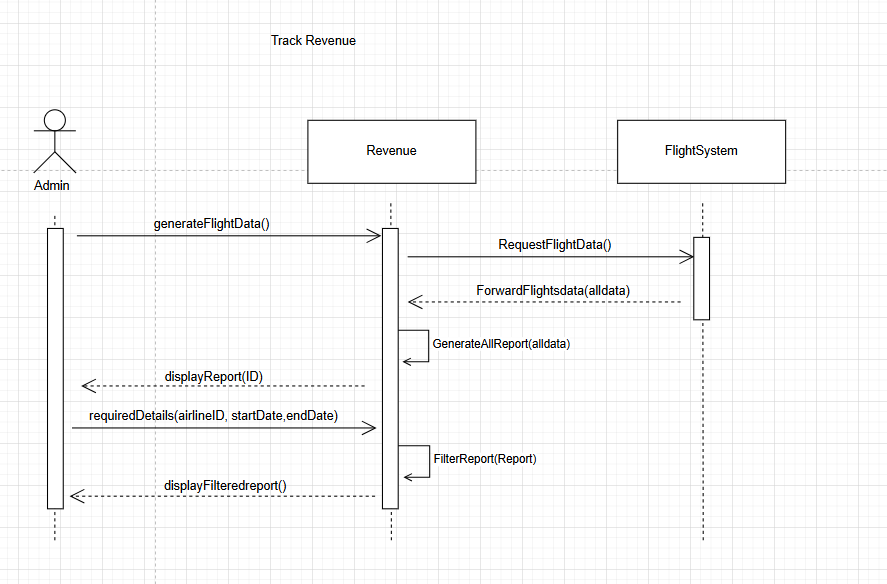
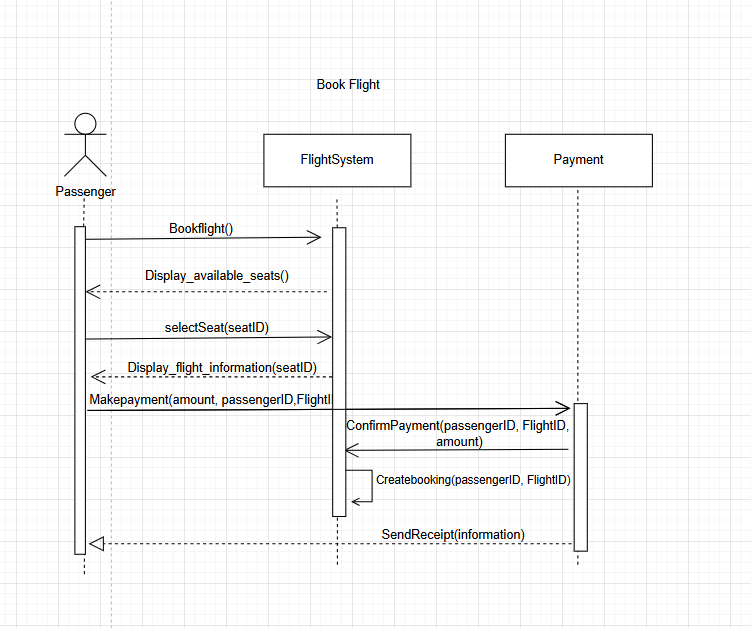
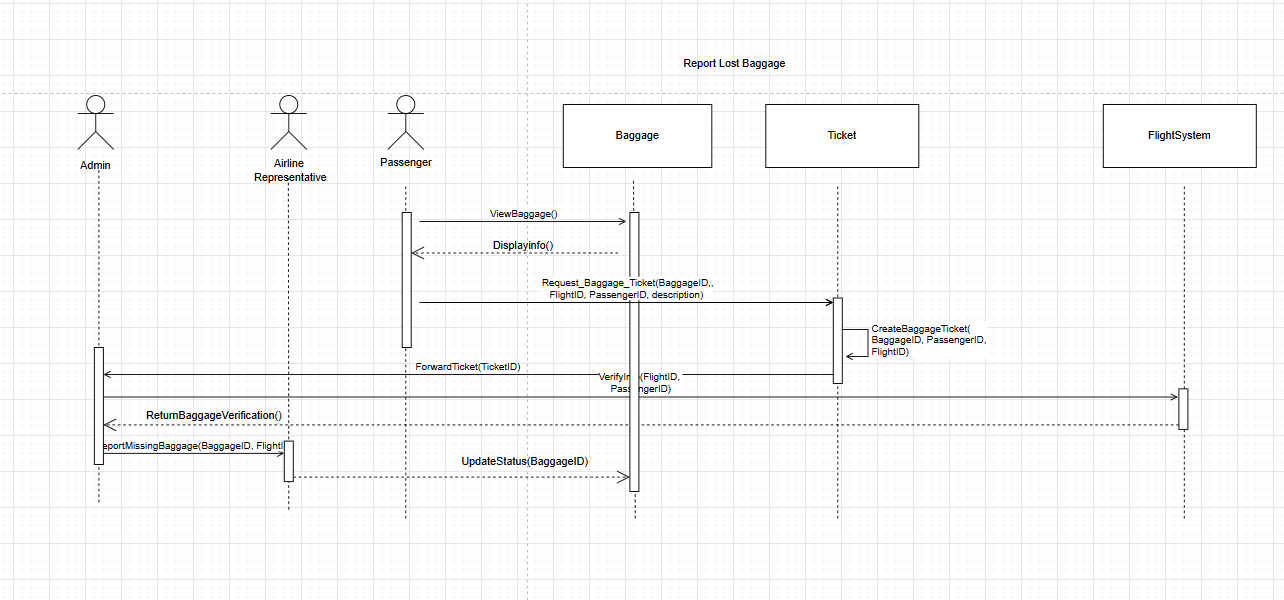
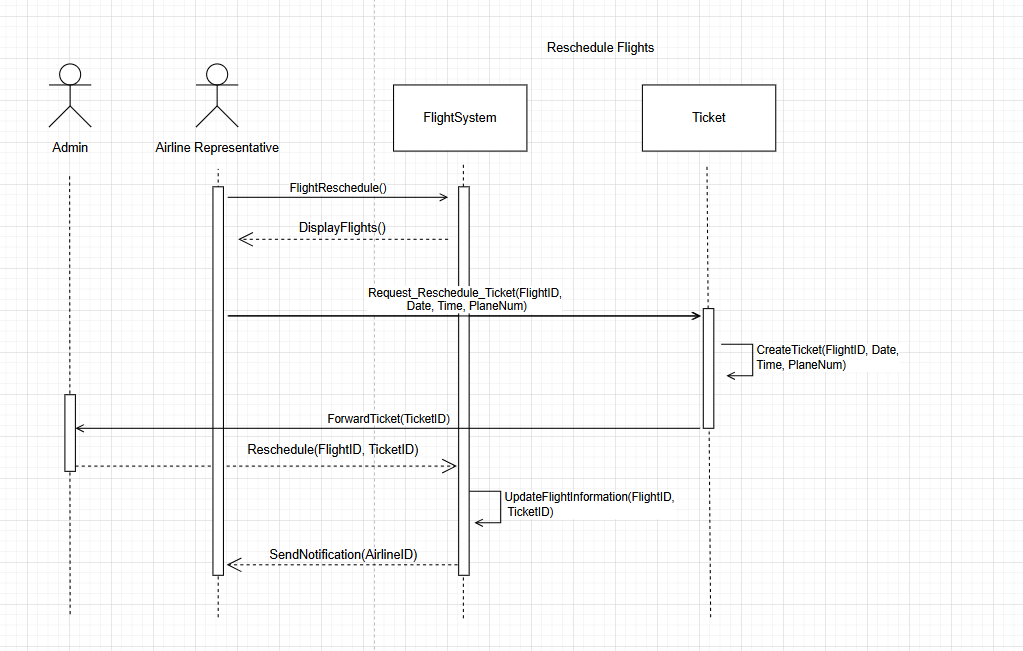
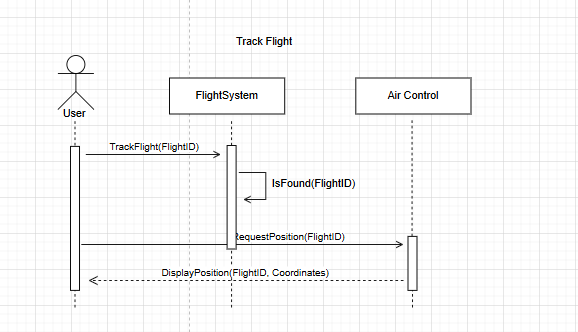
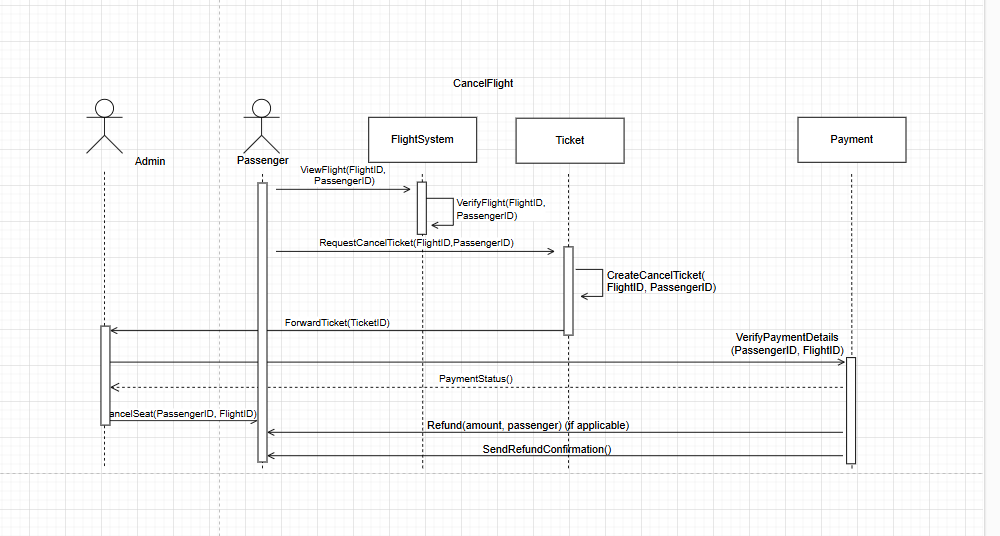
# Domain Model



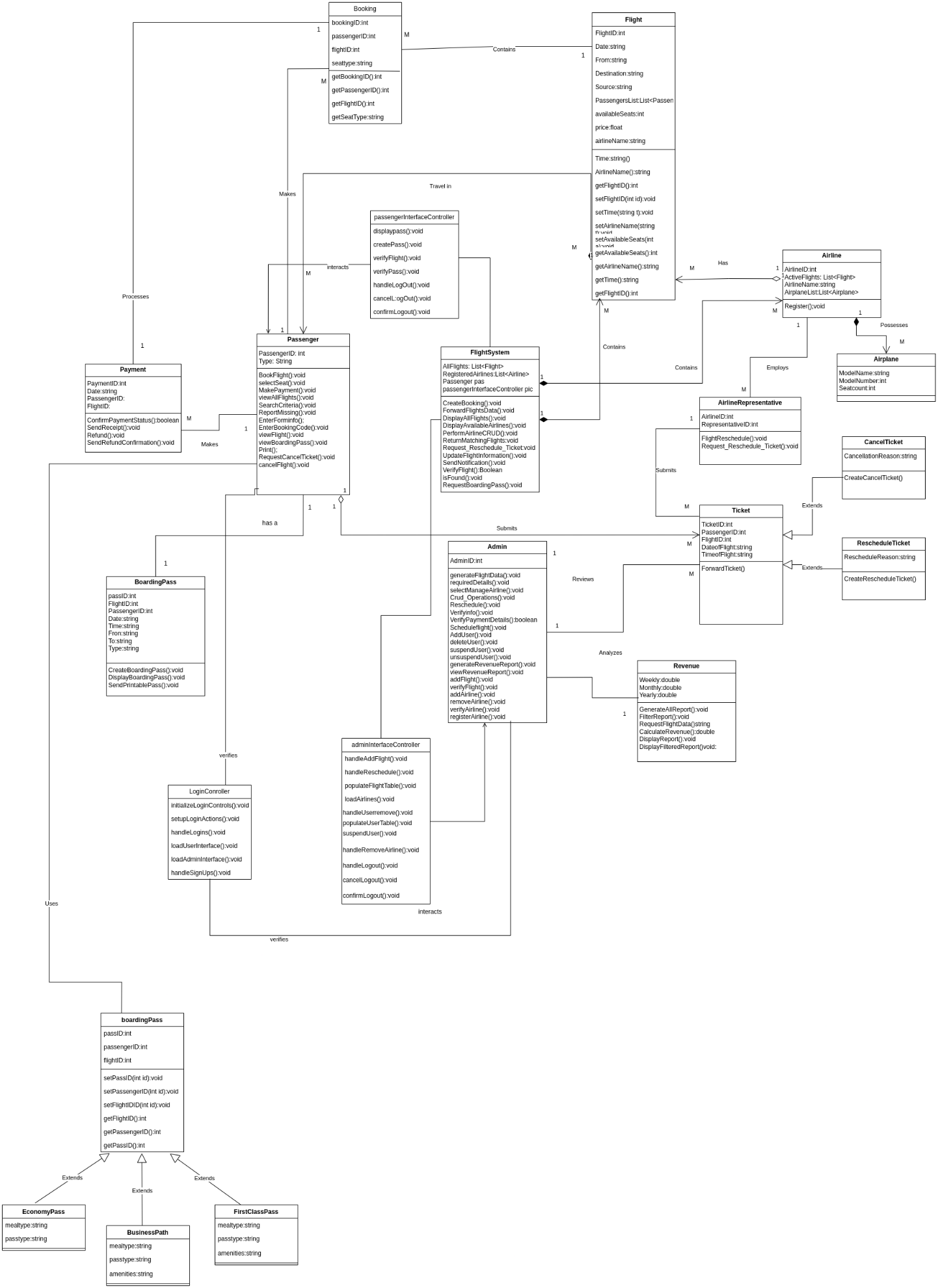
# System Sequence Diagram



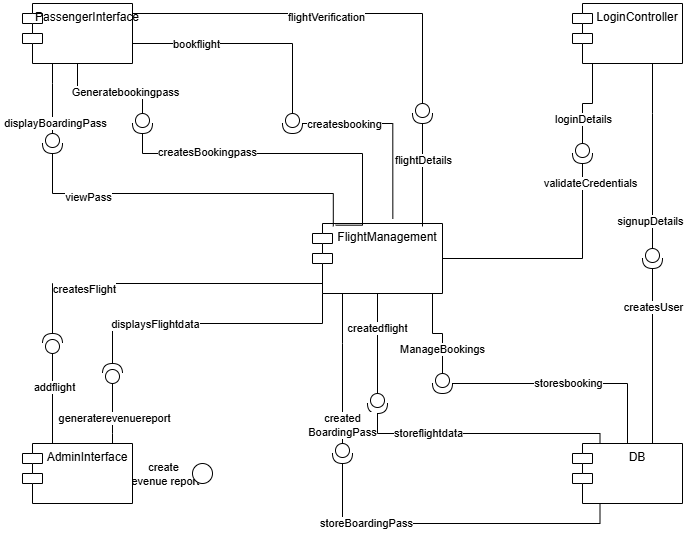
# Sequence Diagram



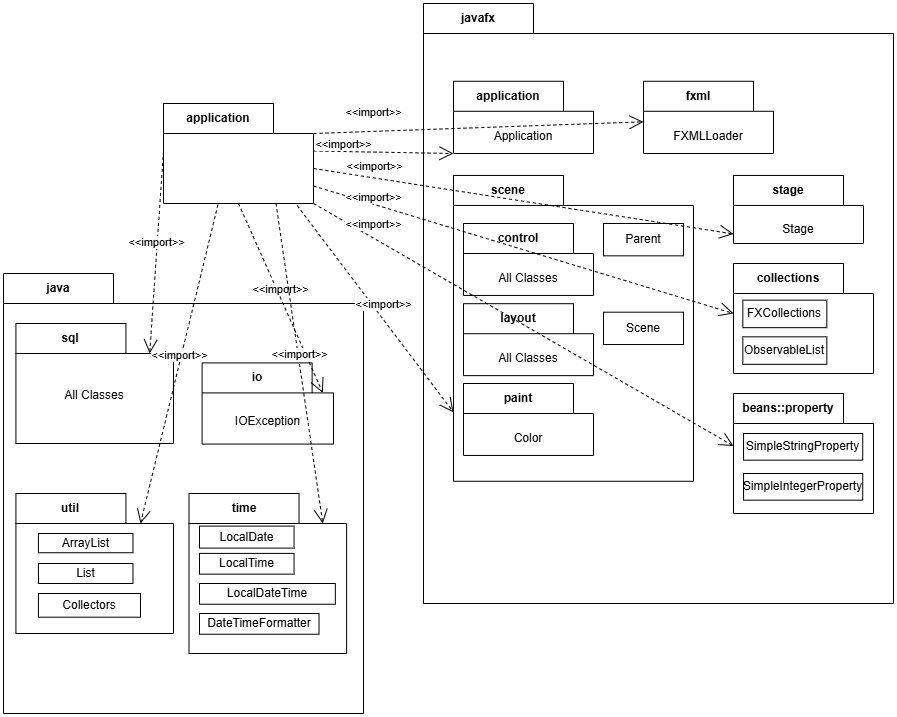
# Class Diagram



# Component Diagram



# Package Diagram



# Deployment Diagram

