ENSF 480
Flight Reservation System

Final Project

Names: Carson May (30139961), Aarsh Shah, Alessandro Baldassarre, Naina Gupta
Course Name: Principles of Software Design
Lecture Section: L02
Course Code: ENSF 480
Submission Date and Time: Dec 12, 2023

Video:

https://yuja.ucalgary.ca/V/Video?v=973324&node=5165788&a=176225656&autoplay=1

Part 1

System Description:

The Flight Reservation Web Application is an integrated platform designed to cater to the needs of various stakeholders in the airline industry, including passengers, tourism agents, airline agents, and system administrators. This web-based system provides an interface for users to browse and book flights to a multitude of destinations, select preferred seating arrangements, and manage travel details with ease. Registered users enjoy additional benefits, such as membership perks and promotional offers. Airline agents can efficiently manage passenger lists and flight attendants, while system administrators are equipped with robust tools for comprehensive management of flight schedules, crew assignments, and aircraft availability. Additionally, the application handles payment processing and issues electronic tickets and receipts directly to the user's email. The database serves as a central repository for storing all user information and flight details, ensuring data integrity and supporting the backend processes. The system aims to enhance the user experience, streamline operational efficiencies for airline staff, and provide a secure and reliable channel for managing flight reservations and travel accommodations.

Actors: Use cases

- **Admin**: manage flights and schedules, manage user accounts, handle payments/refunds, generate reports
- **Passenger**: Search and book flights, view and manage reservations, check-in, view flight status
- **Customer support**: Assist customers with reservations, handle questions/issues, provide information
- **Notification service**: Send customers booking confirmations, send flight updates
- **Payment system**: Process payments, handle payment-related issues
- **Airline agents:** Manage the list of passengers on flights.
- **Registered user:** Register for membership, apply for a company credit card, receive monthly promotional news, receive a free companion ticket annually, access airport lounges with a discount
- **Database:** store information for all users in the system and information regarding flights.

Use Case Scenarios:

Passenger:

Scenario 1
Title: Booking a Flight
Actors: Passenger, Flight Reservation System
Preconditions: The passenger has access to the internet and the reservation system.
Flow of Events:

1. The passenger logs into the system.
2. The passenger selects the "Book a Flight" option.
3. The system displays a list of available flights.
4. The passenger selects a desired flight.
5. The system displays seat choices and their respective prices.
6. The passenger selects a desired seat and seat-type, and is given the option to select ticket cancellation insurance.
7. The system updates the total cost and prompts for payment.
8. The passenger enters credit card details and confirms payment.
9. The system processes the payment and books the seat.
10. The notification service sends a confirmation email/notification with the ticket to the passenger.
11. The passenger logs out.

Scenario 2
Title: Cancel a Flight Reservation
Actors: Passenger, Flight Reservation System
Preconditions: The passenger has an existing booking.
Flow of Events:

1. The passenger logs into the system.
2. The passenger selects the "Manage My Booking" option.
3. The system displays the passenger's current bookings.
4. The passenger selects a booking to view or change.
5. The system displays booking details, including the option to cancel the flight.
6. The passenger selects the cancellation option.
7. The system asks for confirmation and informs about the cancellation policy.
8. The passenger confirms the cancellation.
9. The system processes the cancellation and sends an email confirmation along with a receipt for any refund.
10. The passenger logs out.

Scenario 3

Title: Registering as a user
Actors: User, User Registration System
Preconditions: The user has access to the internet and the user registration system.
Flow of Events:

1. The user accesses the user registration system.
2. The user selects the "Register for User" option.
3. The system prompts the user to enter personal details, such as name, address, email, and phone number.
4. The user enters the required personal details and proceeds to the next step.
5. The system processes the request and registers the user in the database.
6. The system displays a confirmation message to the user.
7. The system displays the profile page that displays all of the user's information
8. The user logs out.

Admin:

Scenario 1:
Title: Adding a New Flight Destination
Actors: Administrator, Flight Destination Management System
Preconditions: The administrator has administrative access to the flight destination management system.
Flow of Events:

1. The administrator logs into the flight management system.
2. The administrator selects the “Manage Destinations” option.
3. The system presents the current list of destinations.
4. The administrator selects the "Add New Destination" button.
5. The system prompts the administrator to enter details of the new destination, including airport name, city, and country.
6. The administrator enters all required information and submits it.
7. The system validates the entered details and adds the new destination to the database.
8. The system confirms the addition to the administrator.
9. The administrator logs out.

Scenario 2: Admin Creating Flight
Title: Creating a New Flight Schedule
Actors: Administrator, Flight Scheduling System
Preconditions: The administrator has administrative access to the flight scheduling system.
Flow of Events:

1. The administrator logs into the flight management system.
2. The administrator selects the “Manage Flights” option.
3. The system presents a form to enter flight details, including, origin, destination, departure time, and arrival time.
4. The administrator enters the necessary details for the new flight.
5. The system validates the information.
6. Upon validation, the system adds the flight to the schedule.
7. The system displays a confirmation message to the administrator.
8. The administrator logs out.

Scenario 3: Admin Adding Aircraft
Title: Registering a New Aircraft
Actors: Administrator, Aircraft Management System
Preconditions: The administrator has administrative access to the aircraft management system.
Flow of Events:

1. The administrator logs into the flight management system.
2. The administrator selects the "Add New Aircraft" option.

3. The system presents a form to choose an aircraft type from the airline's fleet.
4. The administrator fills out the form with the aircraft's specifications.
5. The system validates the data and registers the aircraft.
6. The system confirms the successful registration to the administrator.
7. The administrator logs out.

Scenario 4: Admin Adding Crew Member
Title: Adding a Crew Member
Actors: Administrator, Crew Management System
Preconditions: The administrator has access to the crew management system with the necessary rights to add new crew members.
Flow of Events:

1. The administrator logs into the flight management system.
2. The administrator selects the option to add a new crew member.
3. The system prompts the administrator to choose one of the crew members in the system.
4. The administrator chooses the crew member.
5. The system validates and registers the new crew member.
6. The system acknowledges the addition of the new crew member to the administrator.
7. The administrator logs out.

Scenario 5:
Title: Viewing All Registered Users
Actors: Administrator, Airline User Management System
Preconditions: The administrator has access to the airline's user management system and possesses the necessary permissions to view registered user information.
Flow of Events:

1. The administrator logs into the airline's flight management system using their credentials.
2. Upon successful login, the administrator navigates to the "User Management" section of the system.
3. The administrator selects the "View Registered Users" option from the menu.
4. The system displays a table of all the users with information such as user name, personal information, and membership status.
5. After completing the necessary tasks, the administrator logs out of the system.

Scenario 6:
Title: Modifying Flight Information
Actors: Administrator, Flight Management System
Preconditions: The administrator has authenticated access to the flight management system with permission to modify flight details.
Flow of Events:

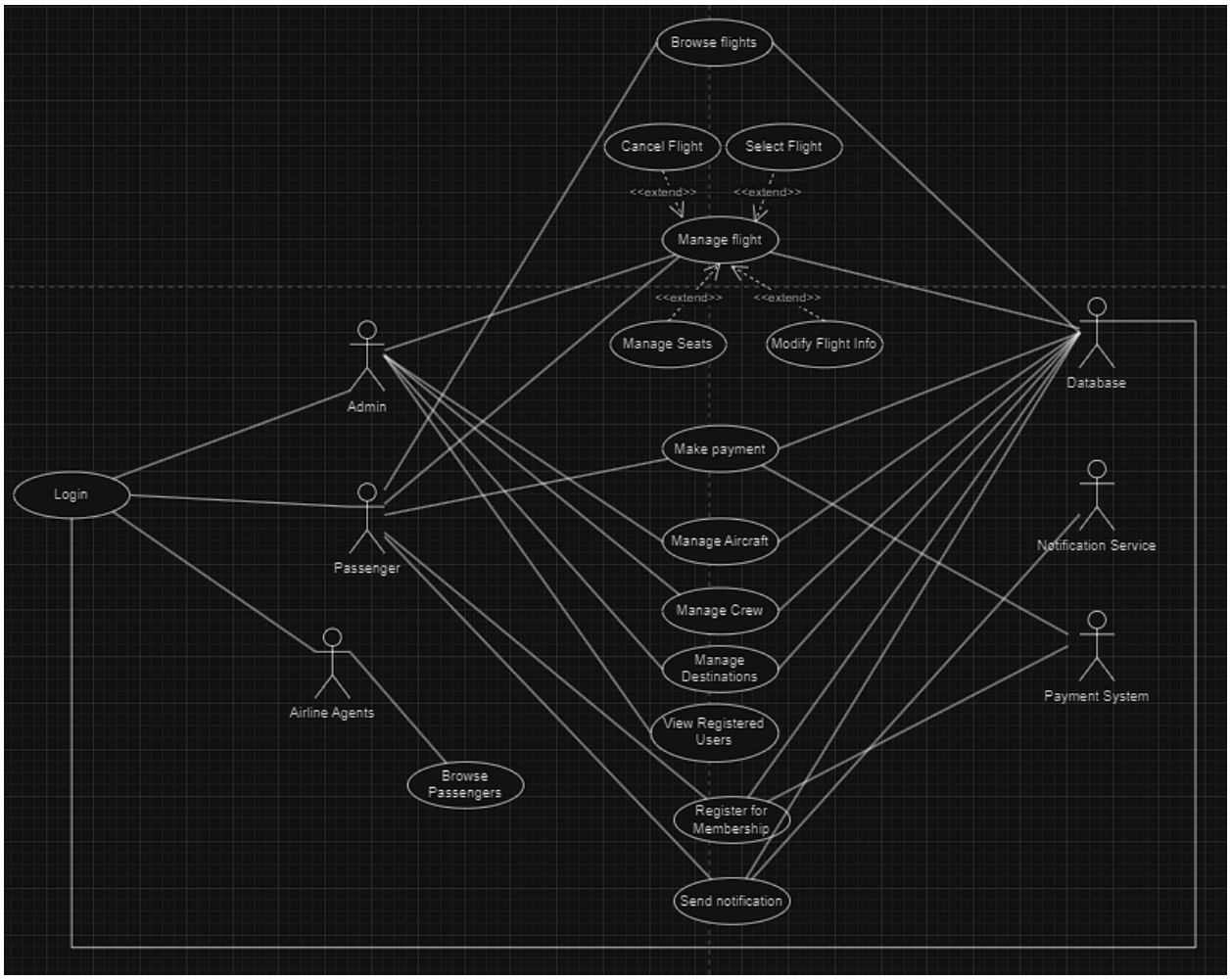
1. The administrator logs into the flight management system using their credentials.
2. After successful login, the administrator navigates to the "Flight Management" section of the system.
3. The administrator selects a flight ID and clicks the “Change Flight" option.
4. The system retrieves and displays the flight details based on the provided criteria.
5. The system presents editable fields containing the flight's current information, including departure and arrival times, aircraft, price, etc.
6. The administrator makes the necessary changes to the flight information, which could include updating times or modifying the aircraft.
7. After making the changes, the administrator reviews the updated details for accuracy and submits the changes.
8. Upon confirmation, the system updates the flight information in the database.
9. The administrator logs out of the flight management system.

Airline Agent

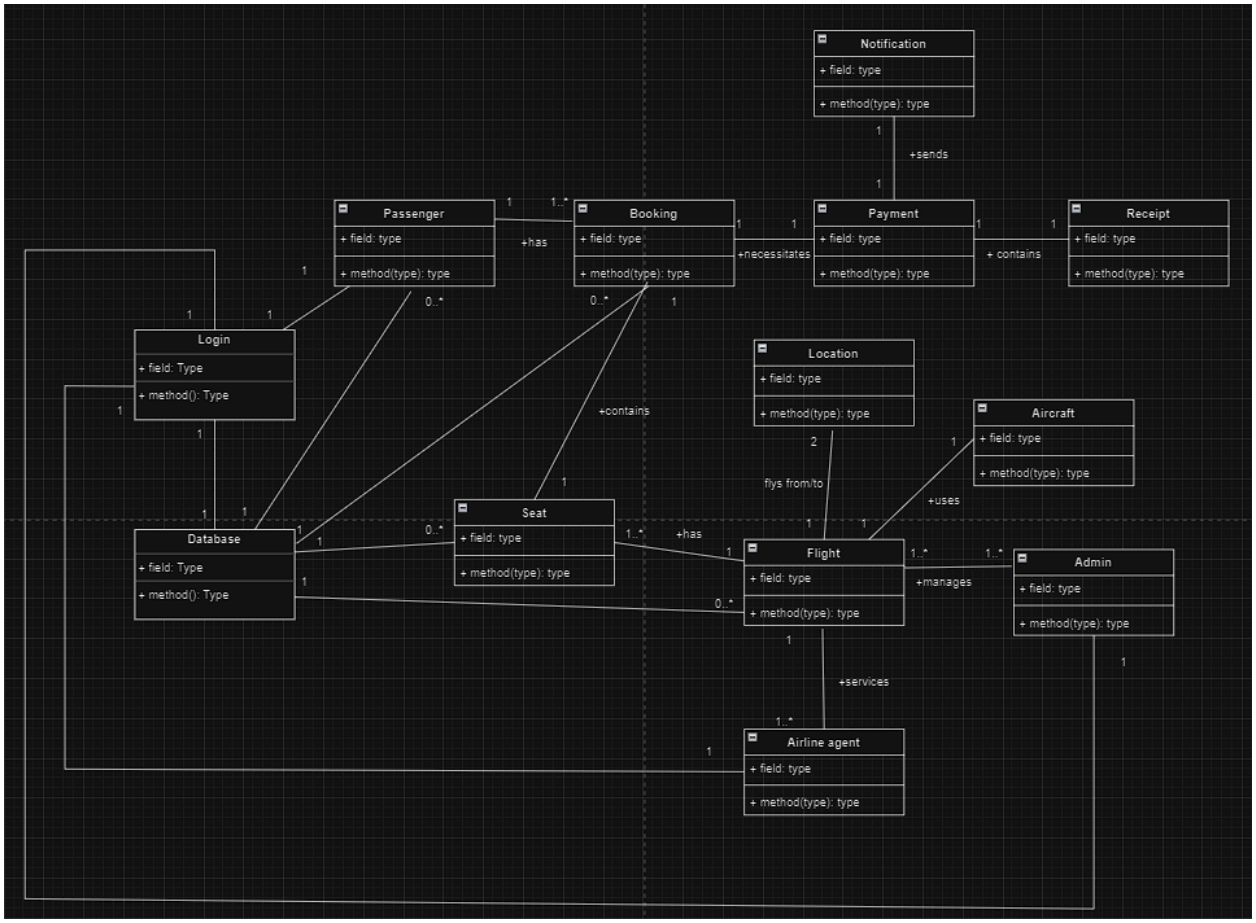
Scenario 1: Airline Agent Browsing Passengers on a Flight
Title: Browsing Passengers on a Flight
Actors: Airline Agent, Passenger Manifest System
Preconditions: The airline agent has access to the passenger manifest system.
Flow of Events:

1. The airline agent logs into the passenger manifest system.
2. The system retrieves and displays all the flights they are assigned to.
3. The airline agent will click a specific flight and a passenger list will show up with any details.
4. After performing the necessary actions, the airline agent logs out.

Use Case:



Conceptual Model:

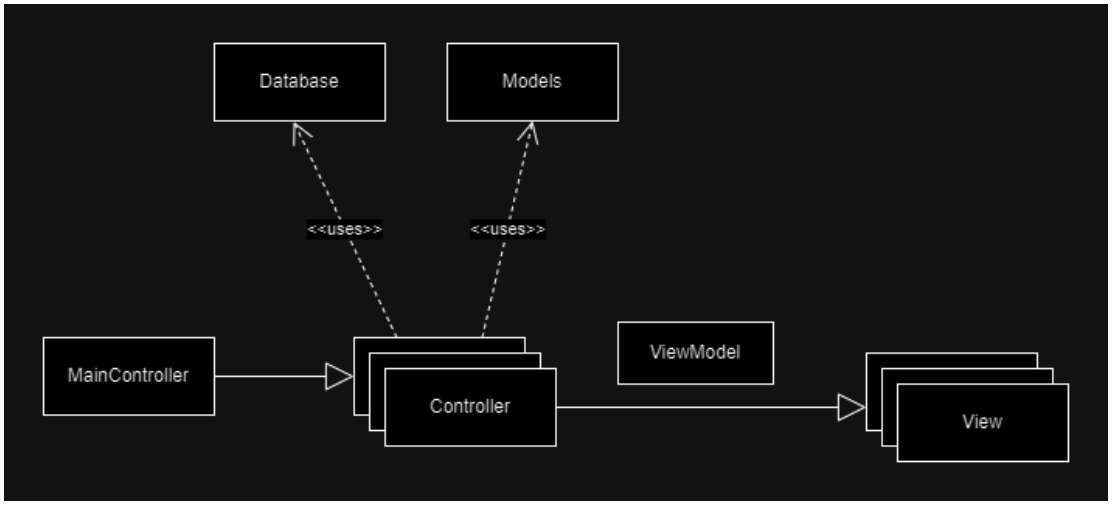


Part 2

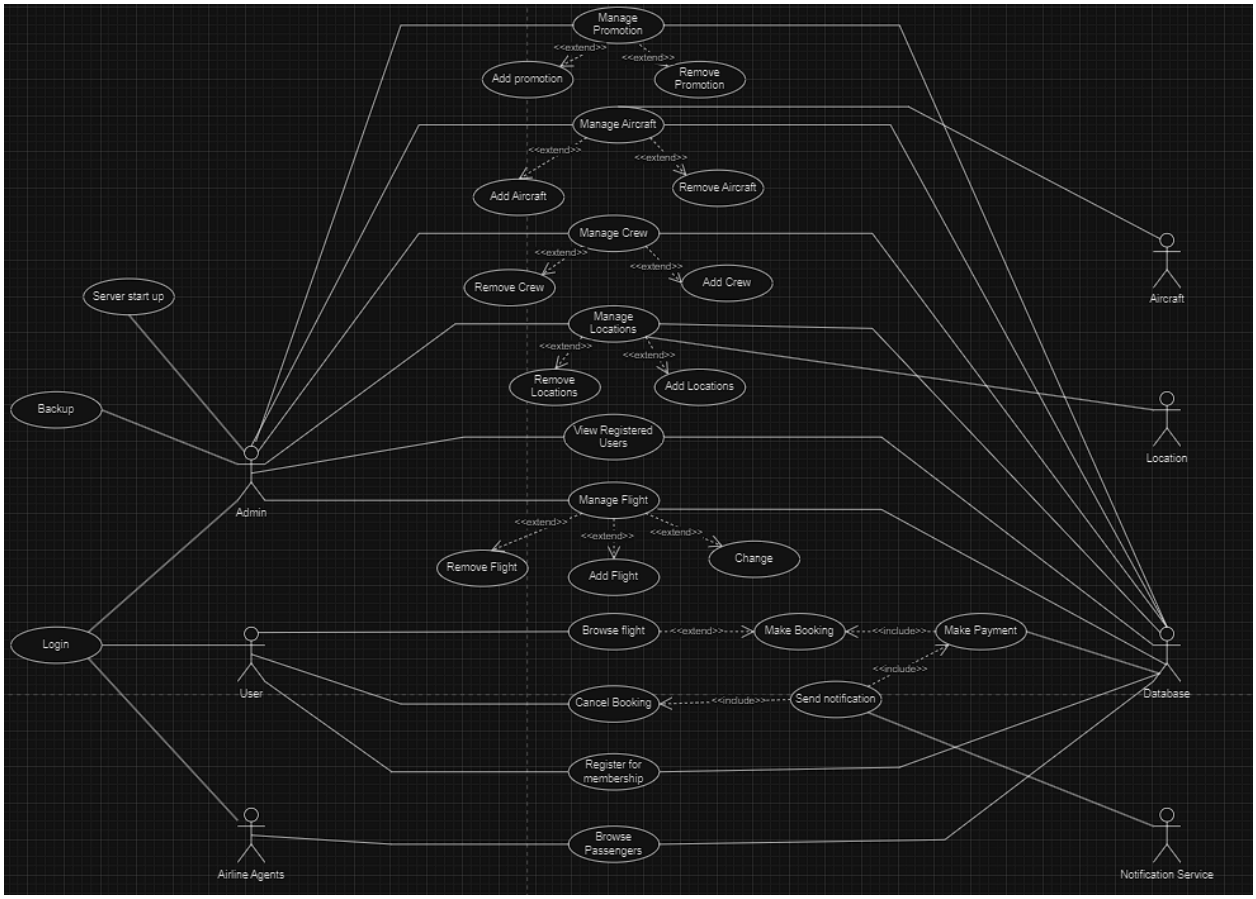
System Architecture:

We opted for a full Java application that utilizes the MVC software architectural pattern to create a GUI. The root of the application starts in App.java which creates the Singelton ‘MainController.’ This controller handles all the navigation, creating the main ‘JFrame’ which holds all the ‘JPanels’ that are classified as the views. The MainController creates an instance of all the subsequent controllers. Each of these controllers handles their own view and view model. The controller will pass any dynamic data via a view model parameter to its view which the view will use to display to the user. All the controllers use Java models or ‘entities’ to model the database tables so that there is an organized way to store the data. The controllers are responsible for pulling data from the MySQL database and handling any logic from the views. The view is the Java GUI which displays the UI such as buttons, the navigation menu, tables, etc. Below is a diagram and a more high-level overview of the system architecture.

- MVC Pattern (Decorator Pattern):
    - Views: GUI Java application Model: Updates the GUI
    - Controller: Controls the data manipulation via the DB class
    - Models: models the database tables
- The Database Class Object (Singleton Pattern)

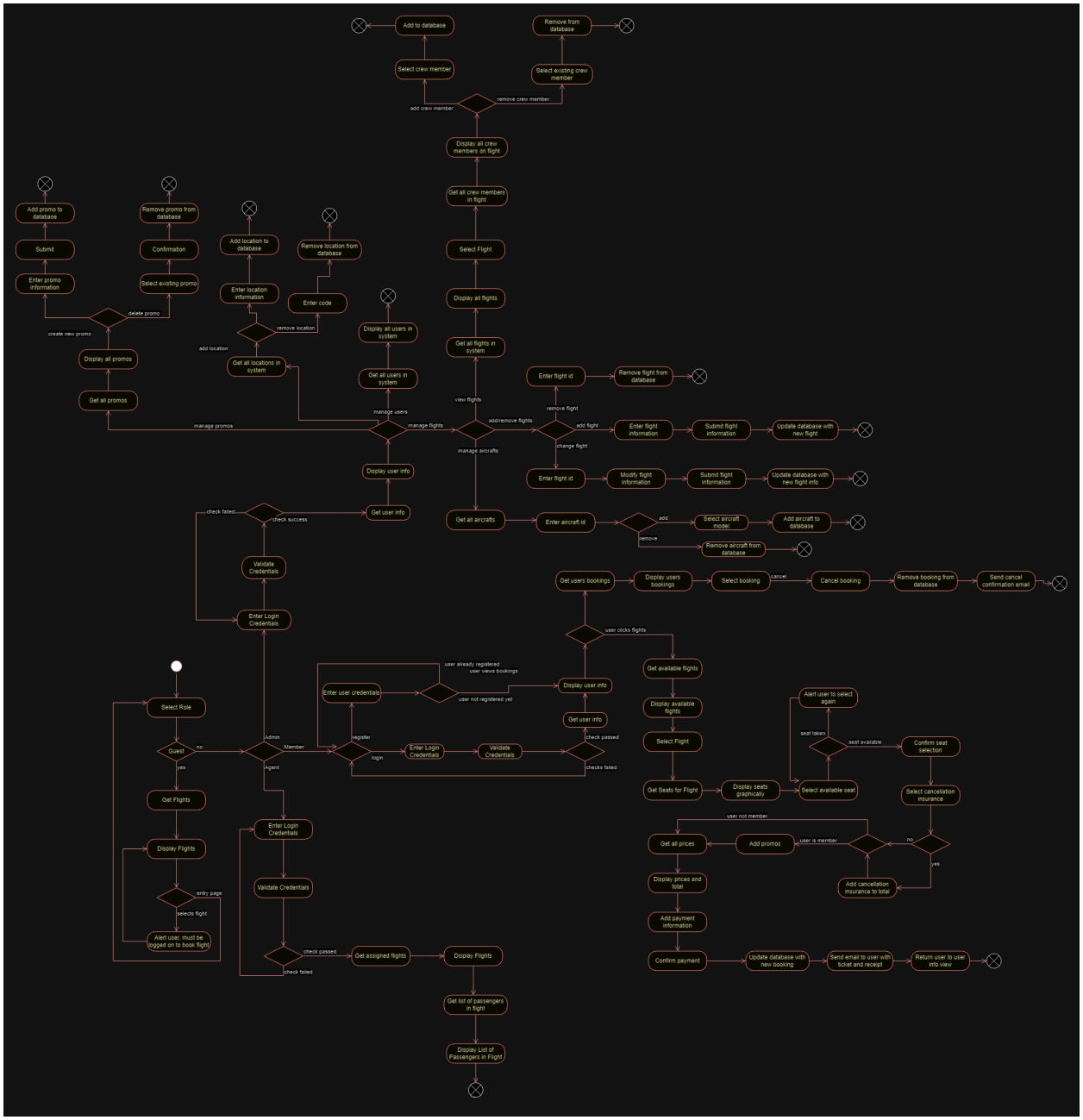


Updated Use Case



(**See attached for high resolution version**)

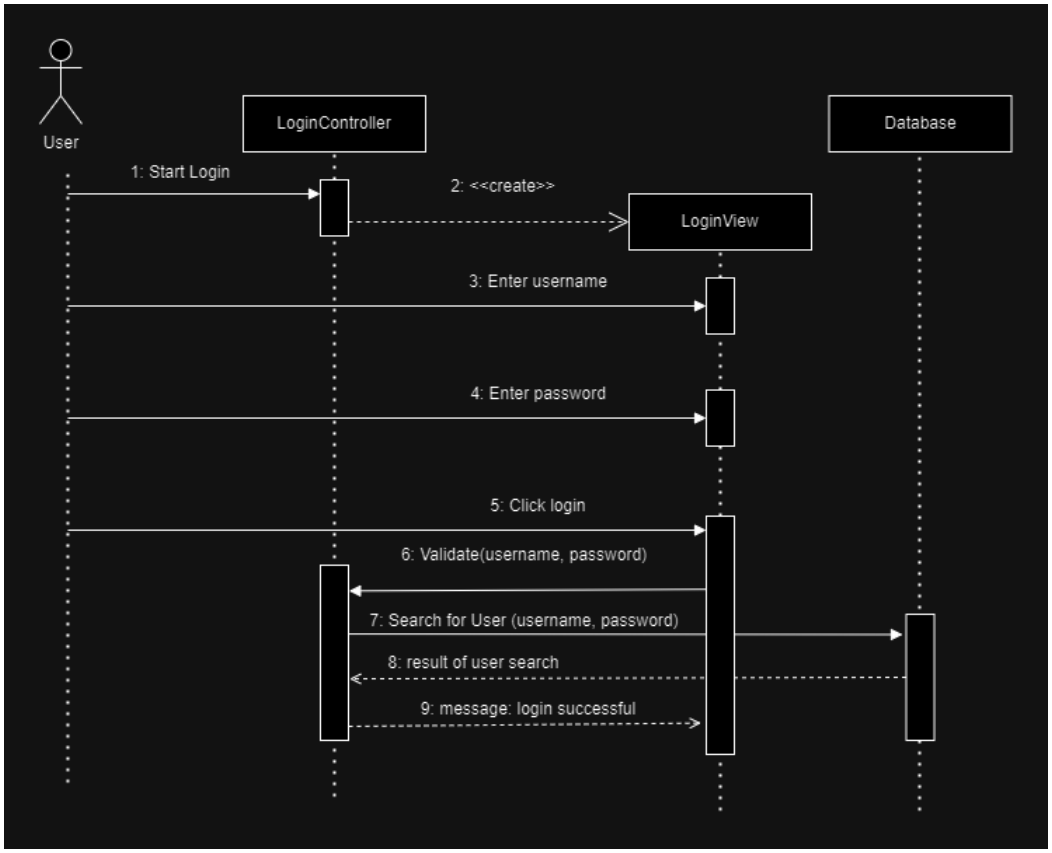
Activity Diagram



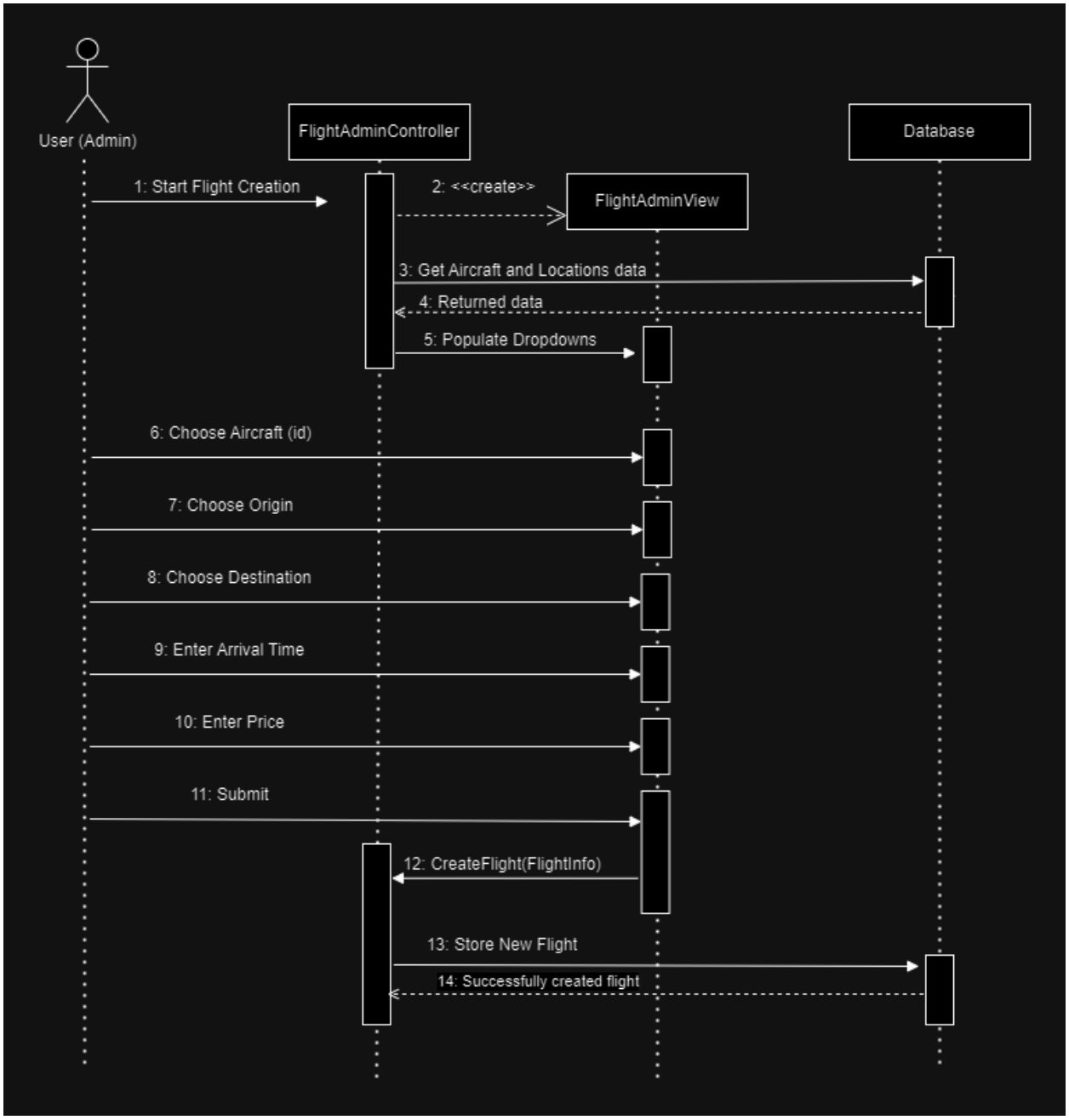
(**See attached for high resolution version**)

Sequence Diagrams:

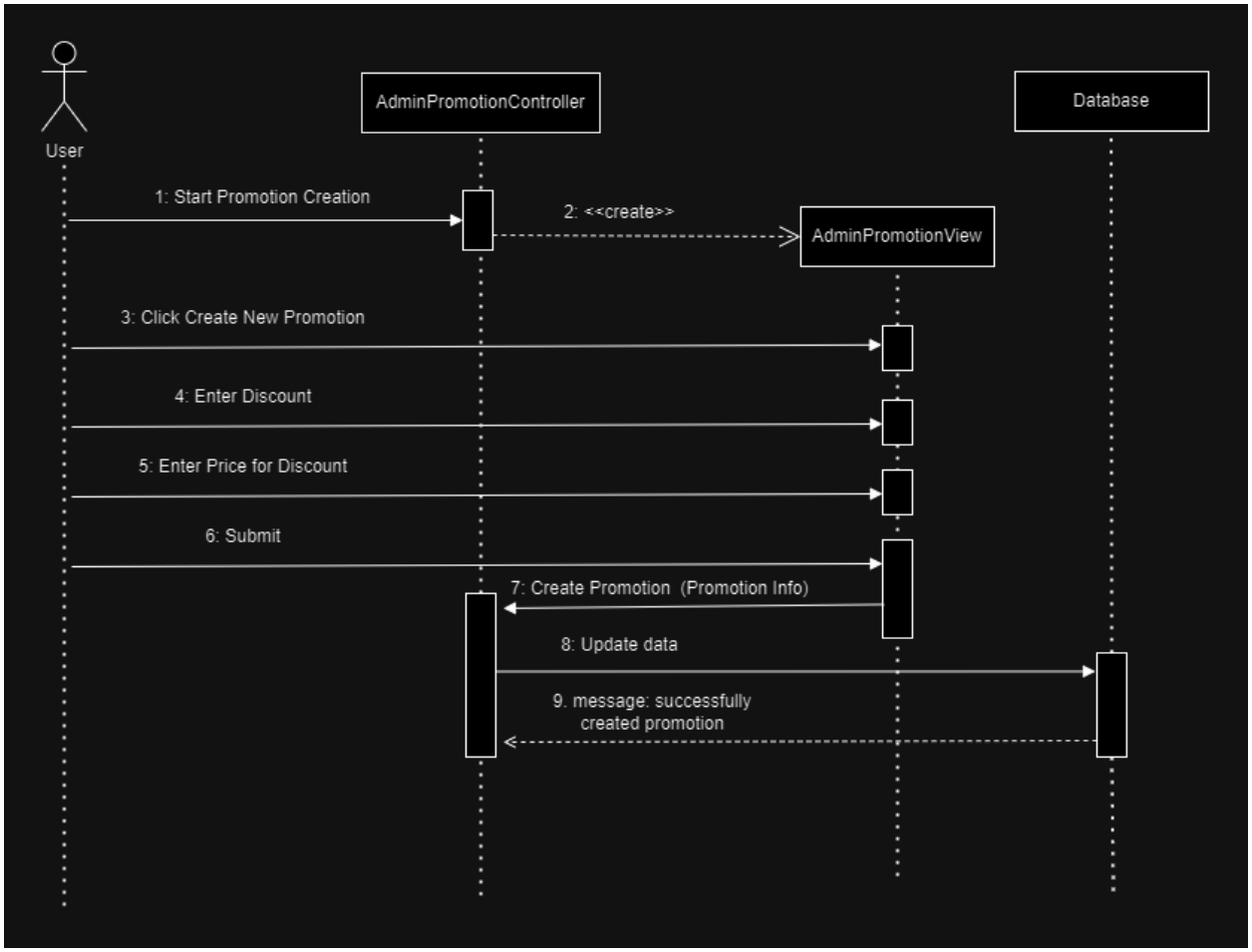
Login Sequence



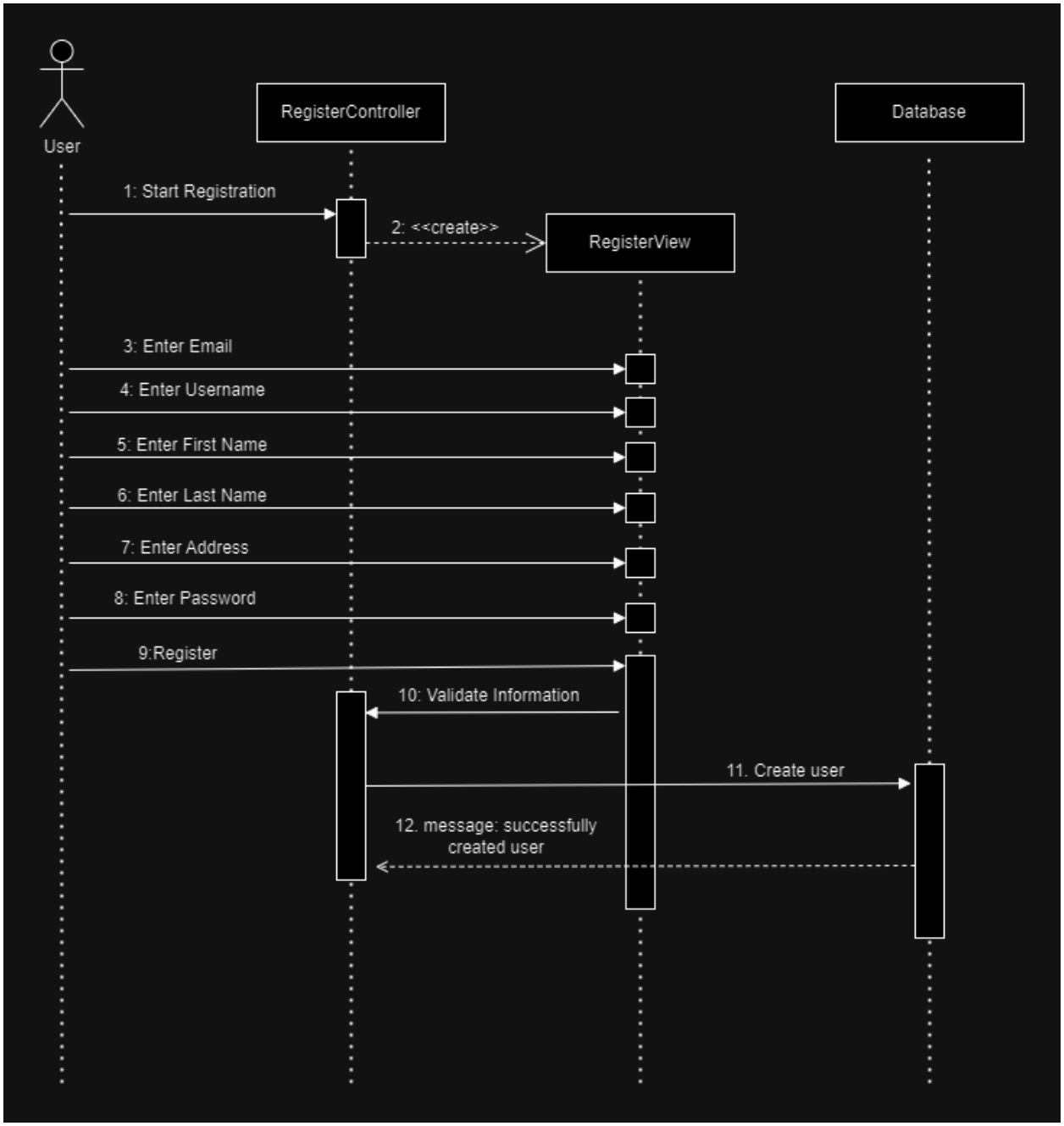
Create Flight Sequence:



Create Promotion Sequence:

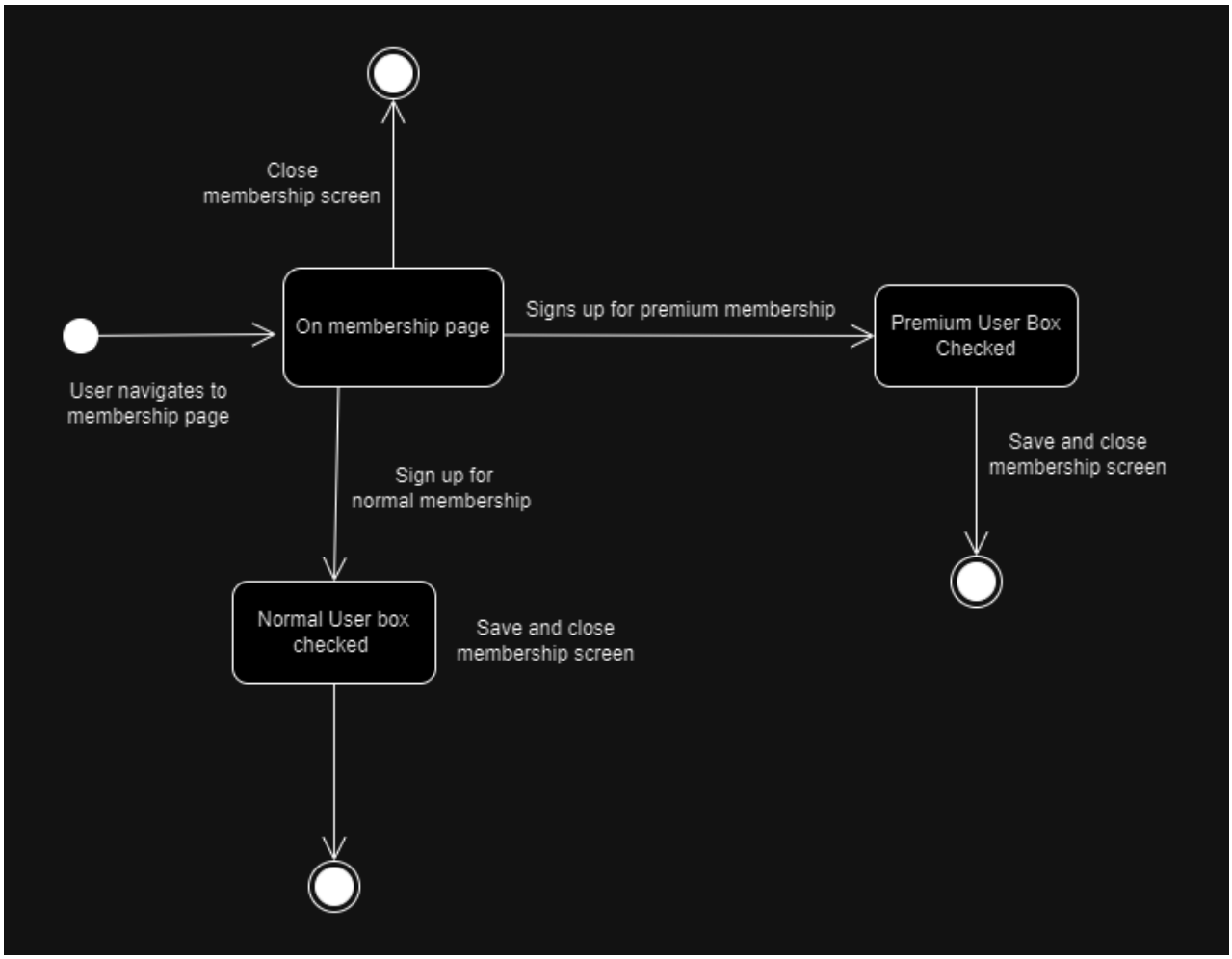


Registration Sequence:

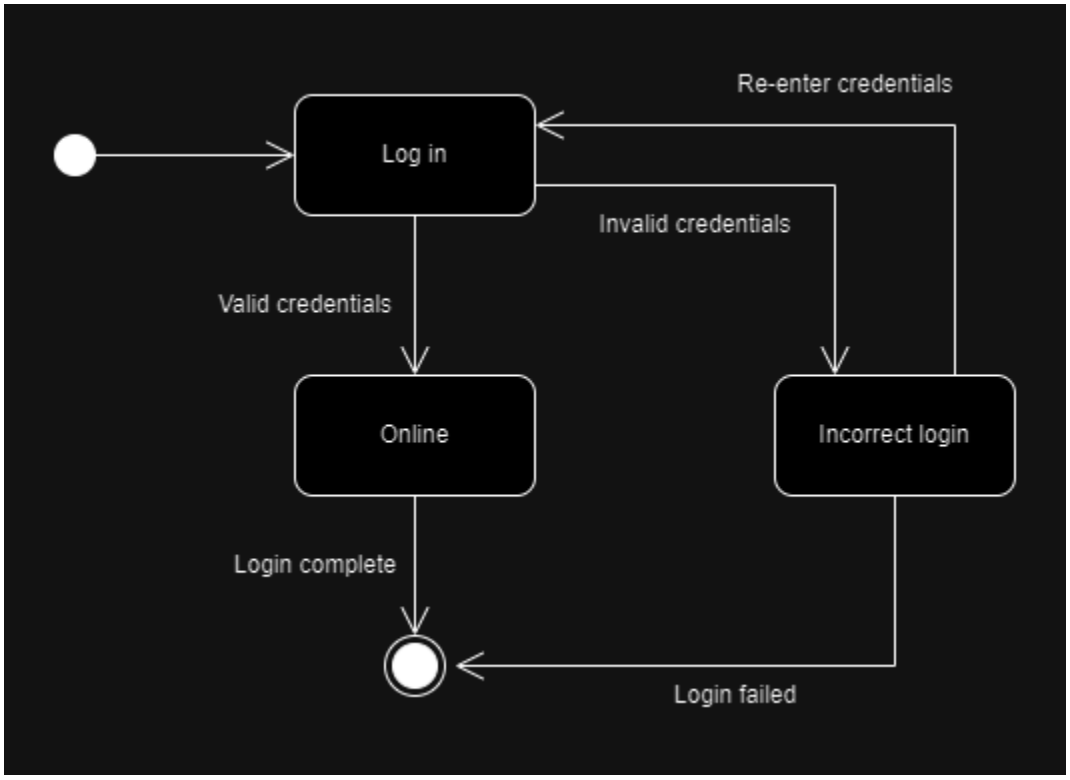


State Transition

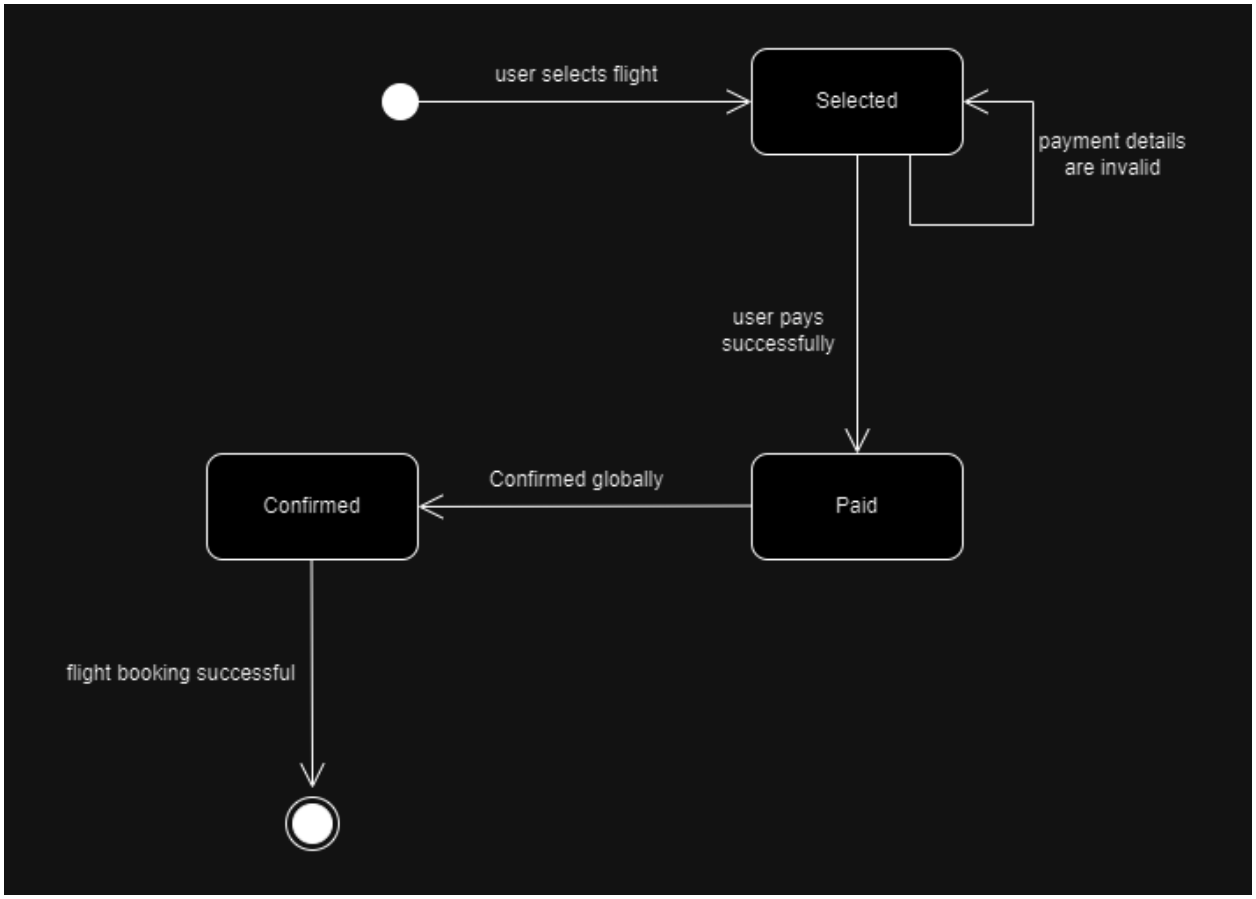
Premium Membership:



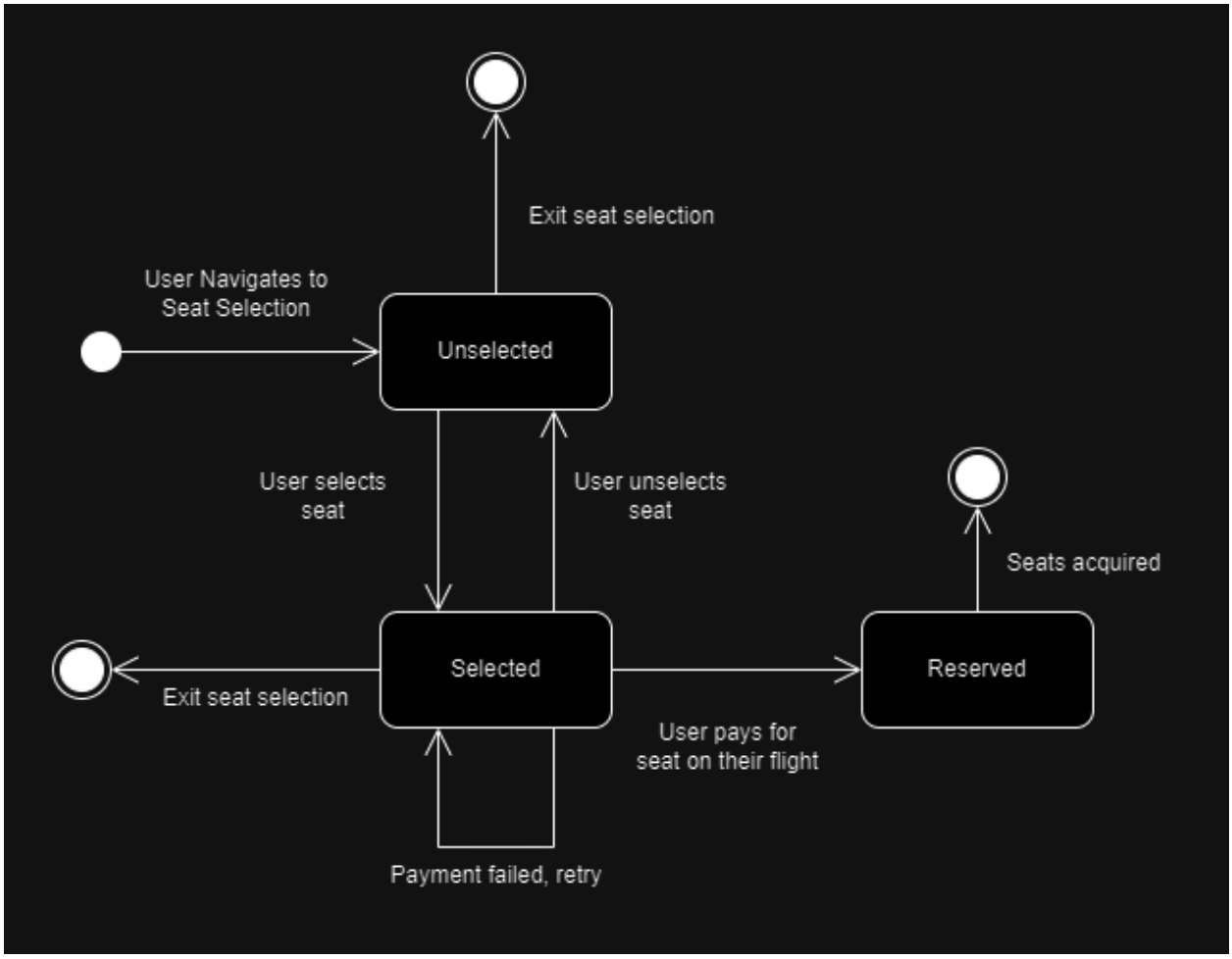
Login State:


Re-enter credentials
Log in
Invalid credentials
Valid credentials
Online
Incorrect login
Login complete
Login failed

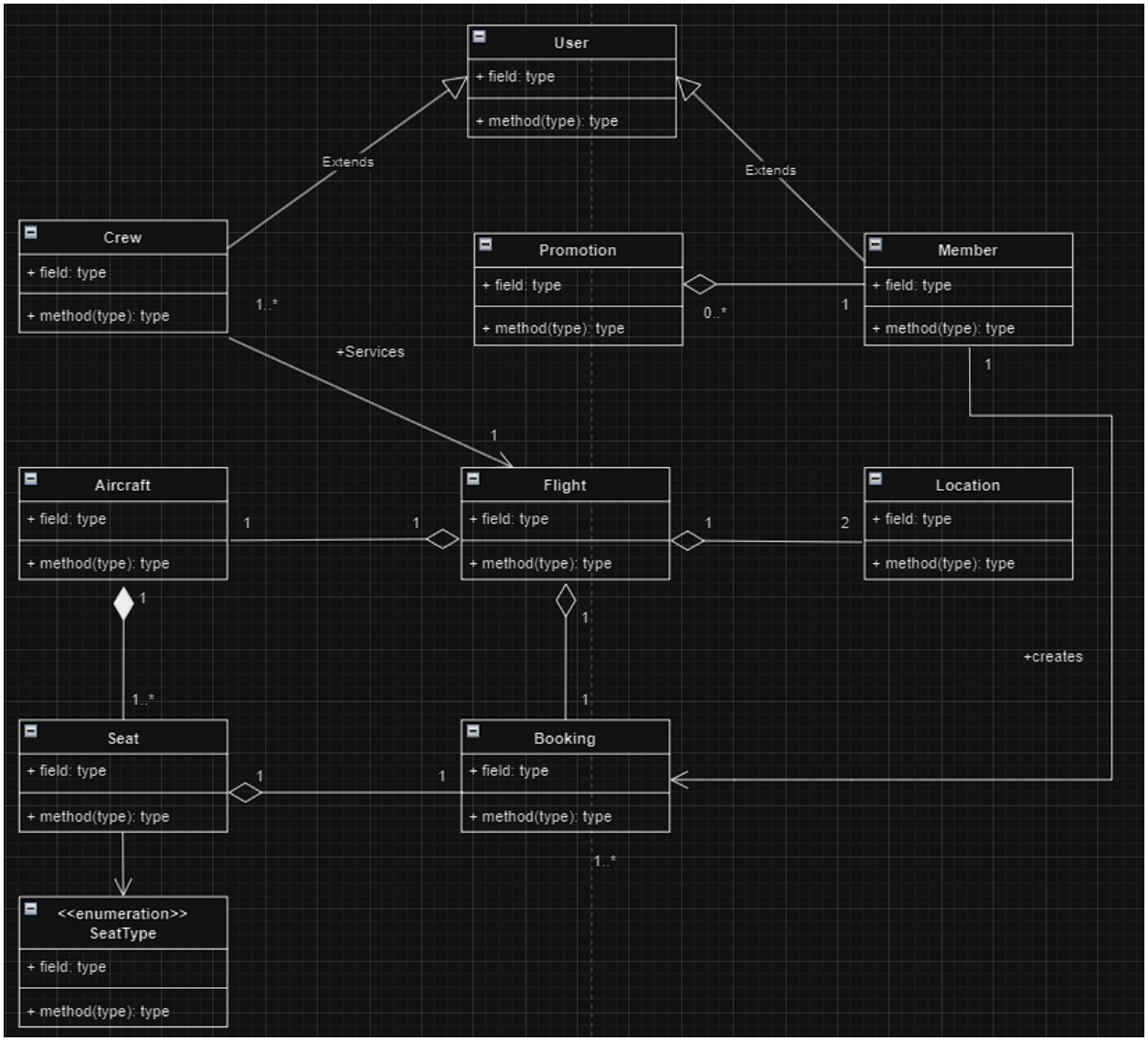
Flight Reservation:



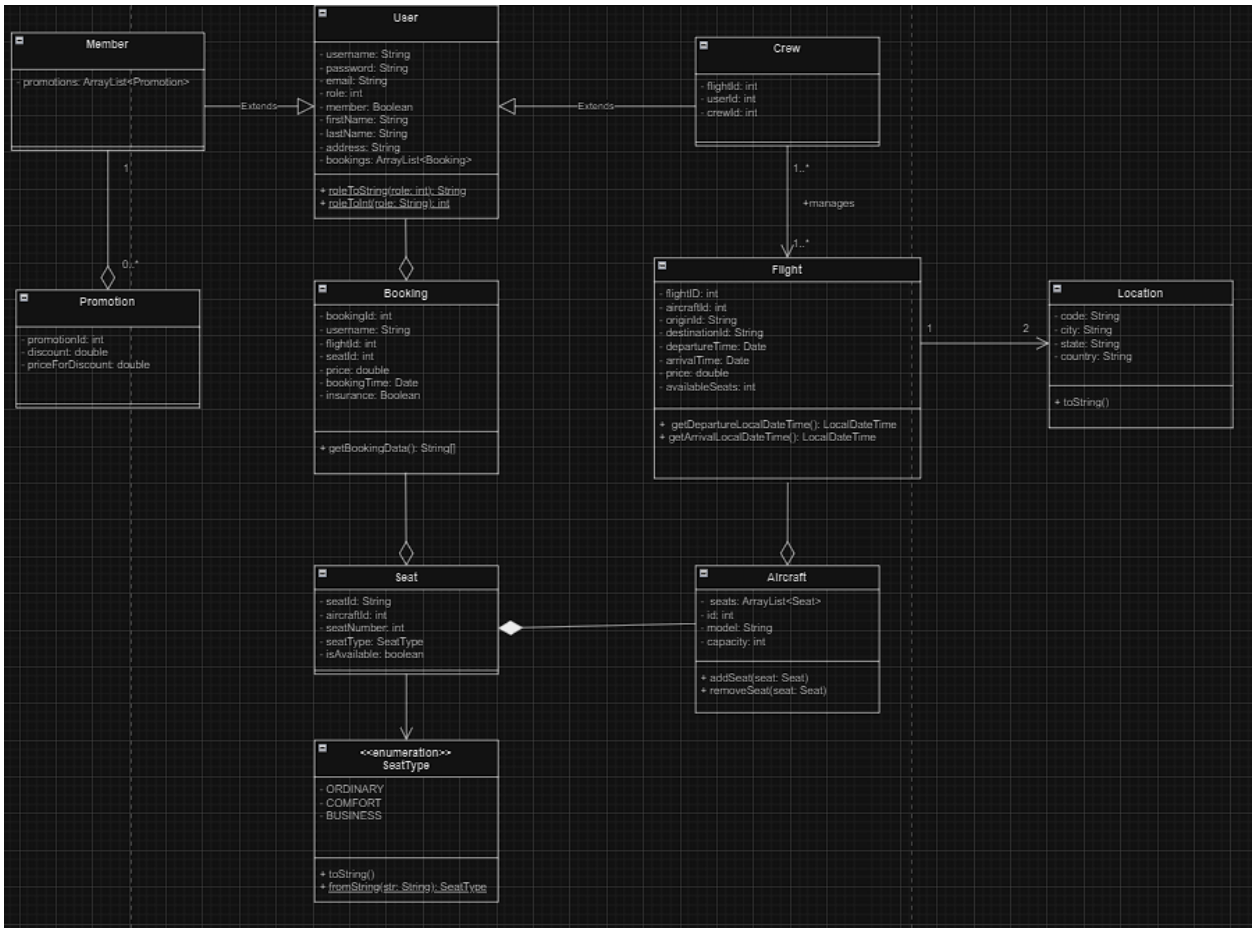
Seat Selection:



Systems Domain Class Diagram



Systems Domain Class Diagram Pt. 2



(**See attached for high resolution version**)

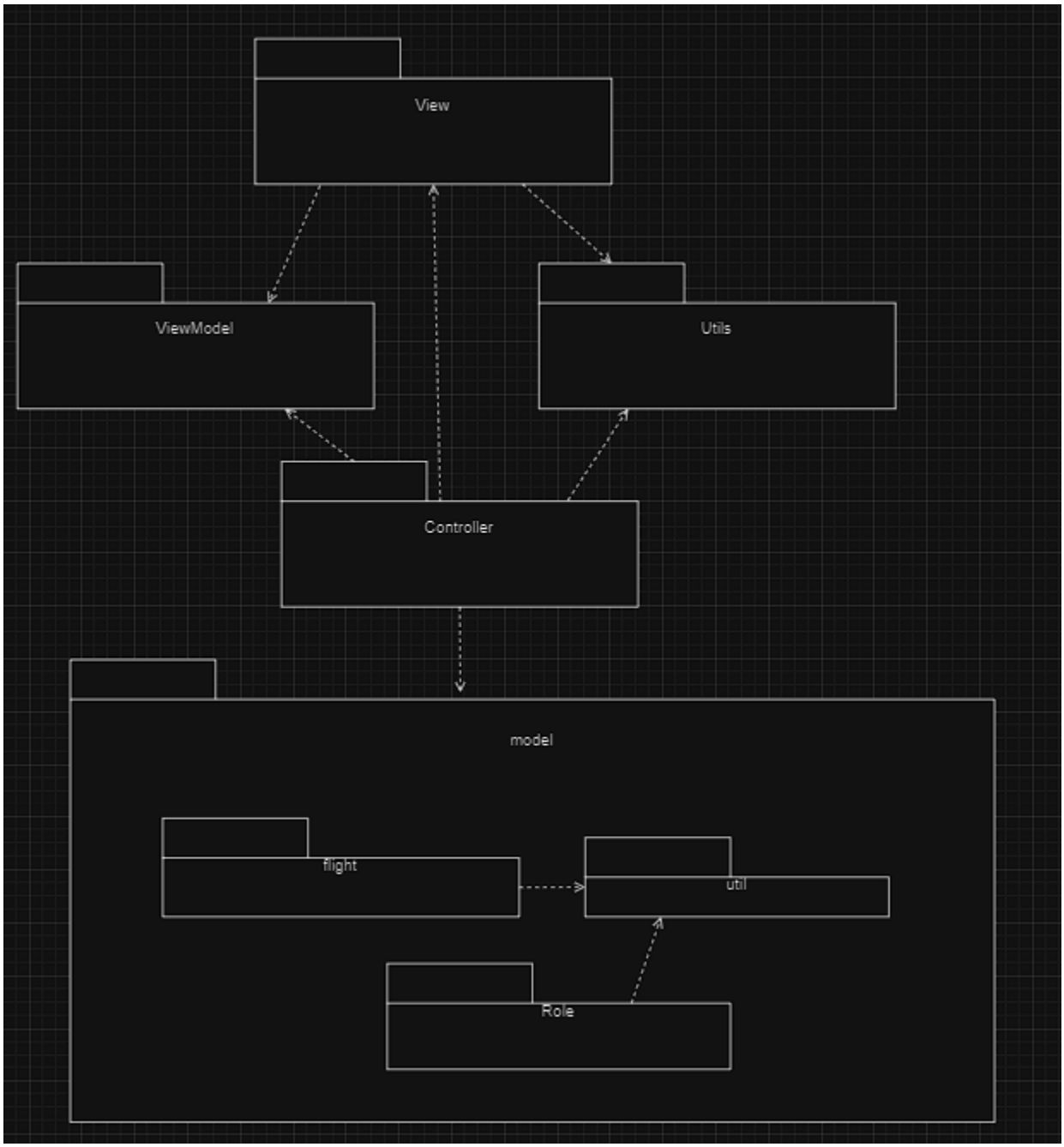
Part C:

System’s Detailed Design-Class Diagram

See attached file

Part D:

Package Diagram



Deployment Diagram