UNIVERSITY OF CALGARY

ENSF 480

Principles of Software Design

Term Project

Student Name: Marcos Perez (30141681)

Noshin Zion (30163372)

Mobha Khan (30139868)

Jori Duguid (30145690)

Group Number: 27

Date Submitted: 02 December 2023

Design Phase

Part A

System Description

System Initialization: When the system launches, customers are welcomed to the flight reservation web application by an attractive home page with navigation bars in the upper right corner. The system is built on the Spring Boot framework, which allows Java and a MySQL database to work together flawlessly. The primary entry point is the home page, which offers customers a user-friendly interface for easy access to main functionalities. Users can choose to log in as a guest, registered user, airline agent, or administrator when they arrive. To ensure security, each choice requires authentication before proceeding.

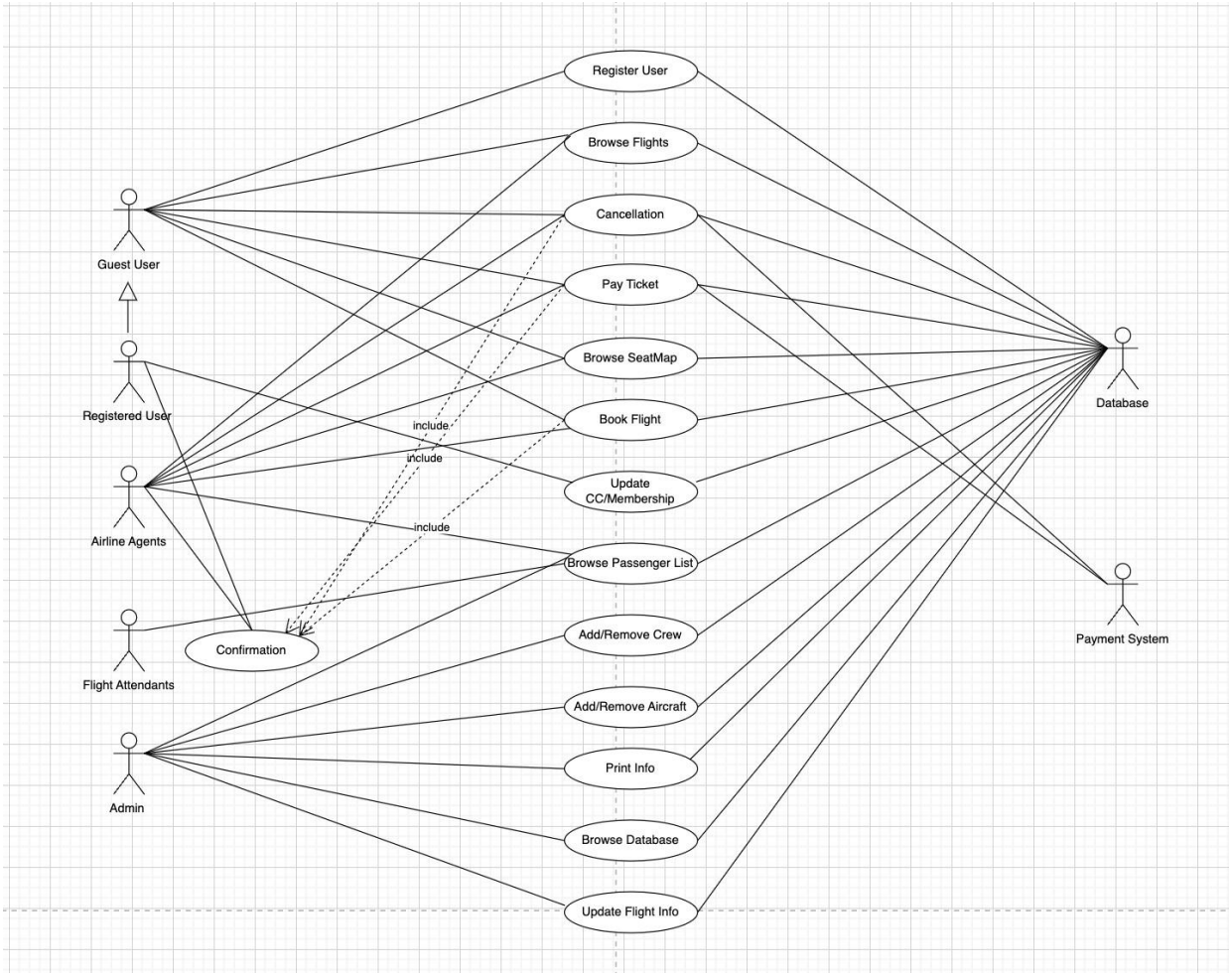
Flight Reservation Process: The procedure for booking a flight is meant to be simple. Users begin by viewing a list of flights that are available in our database. They can start the booking procedure when they've decided and selected a flight. Users are assisted in choosing their preferred seats, whether in business or regular class, using a graphical seat map. Users provide their credit card details and complete the payment process to validate their reservation. Upon successful payment, users get an email confirming their booking. Users can also cancel their flights, and an email notification will be sent.

Registered User and Guest Users: Those who register as registered users receive a personalized experience. In addition to saving their credit card information, they can subscribe to news updates and articles. Even though guest users' experience is simpler in comparison, they still need to log in for important functionality. Both kinds have a simple booking procedure.

Airline Agents: Airline agents have access to passenger lists for flights in addition to the user interface that is like that of registered and guest users. This feature makes it possible for agents to effectively oversee and manage passenger details.

Admin: System administrators can effectively manage crucial data because they have extensive control over the entire platform. They can manage flights (adding, removing, and altering details), supervise crews and aircraft (adding or removing them for best use), and modify flight destinations in response to demand. The system's database contains detailed information that the administrators can examine to make sure everything is correct.

System's Use-case diagram (First Draft)

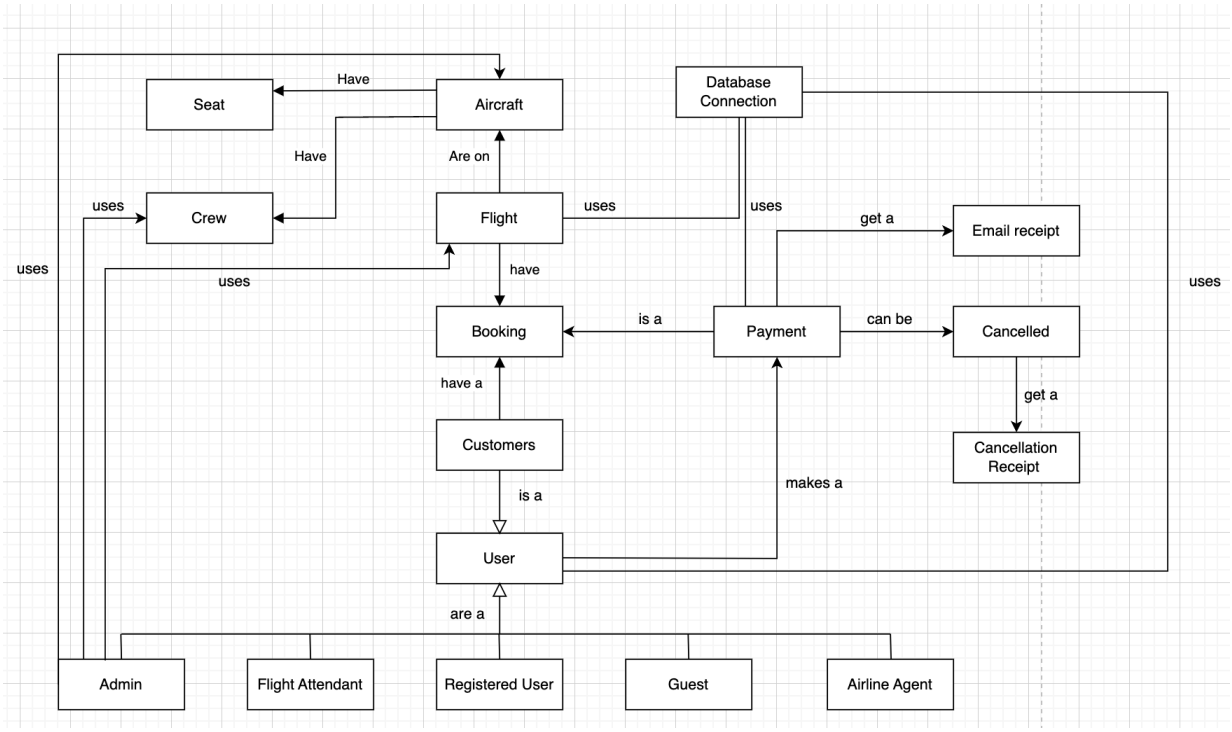


Scenario Description:

1. Register User
Scenario: A guest user decides to become a registered user.
Guest User accesses the Registration Form on the Web Application.
Guest User submits the form with personal details.
Web Application validates the data and creates a new User Account in the Database.
Database stores the User Account information.
Web Application sends a Confirmation Email to the Guest User's email address.
Guest User confirms and becomes a Registered User.
2. Browse Flights
Scenario: A user searches for flights to a specific destination.
User inputs destination and date into the Search Function on the Web Application.
Search Function queries the Flight Information from the Database.
Database returns Available Flights to the Web Application.
User views the list of Available Flights.
3. Cancellation
Scenario: A registered user cancels a booking.
Registered User selects the booking to cancel on the Web Application.
Cancellation Service processes the request and updates the Booking Status in the Database.
Database confirms the update.
Payment System initiates a Refund if applicable.
Web Application sends a Cancellation Confirmation to the Registered User.
4. Pay Ticket
Scenario: A user completes payment for a ticket.
User selects a flight and proceeds to Checkout on the Web Application.
Checkout Service sends the Payment Details to the Payment System.
Payment System processes the Payment and sends a Payment Confirmation.
Database updates the Booking Status as paid.
Web Application sends a ticket and Receipt to the User.
5. Browse SeatMap
Scenario: A user views the seat map of a flight to choose a seat.
User selects a specific flight on the Web Application.
Web Application requests the Seat Map from the Database.
Database provides the Seat Map.
User views the graphical Seat Map and selects a seat.
6. Book Flight

Scenario: A user books a flight.
User selects a flight and a seat, then provides payment details.
Booking Service creates a new booking with the flight, seat, and user details.
Database stores the Booking Record.
Payment System processes the payment.
Web Application sends a Booking Confirmation with an Ticket to the User.
7. Update CC/Membership
Scenario: A registered user updates their credit card information or membership details.
Registered User accesses the Membership Area on the Web Application.
Membership Service retrieves current details from the Database.
Registered User submits updated Credit Card Information or Membership Preferences.
Database updates the User Profile.
Web Application confirms the updates to the Registered User.
8. Browse Passenger List
Scenario: An airline agent checks the passenger list for a flight.
Airline Agent logs into the Agent Portal and selects a flight.
Agent Portal queries the Database for the Passenger List.
Database returns the Passenger List.
Airline Agent reviews the Passenger List.
9. Add/Remove Crew
Scenario: An admin manages the crew for a flight.
Admin accesses the Crew Management module.
Crew Management service queries the Database for current Crew Members.
Admin adds or removes crews.
Database updates the Crew Roster.
Web Application confirms the changes to the Admin.
10. Print Info
Scenario: An admin prints user information or flight details.
Admin selects the Information to print in the Admin Portal.
Admin Portal retrieves the data

System's Conceptual Model



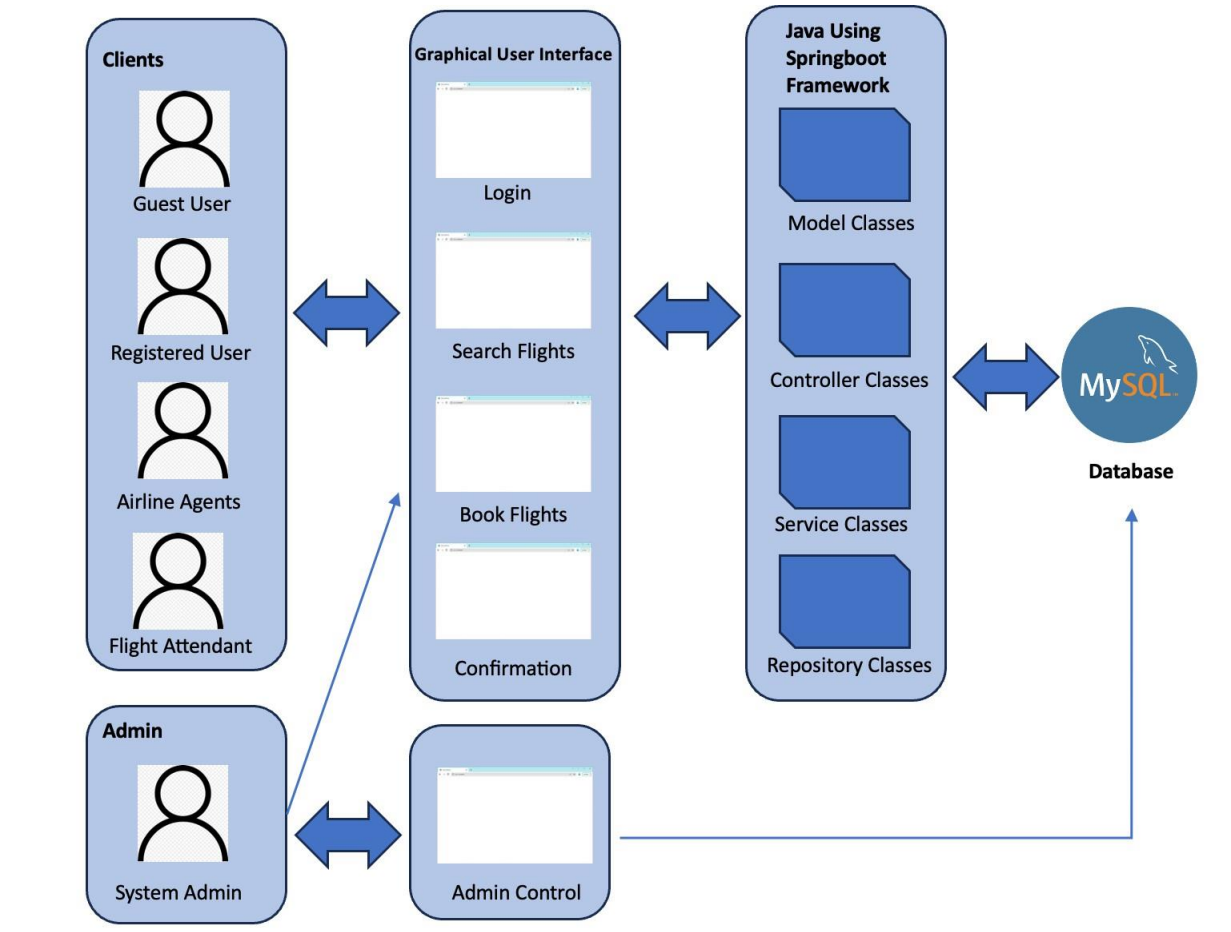
Part B

System's Architecture

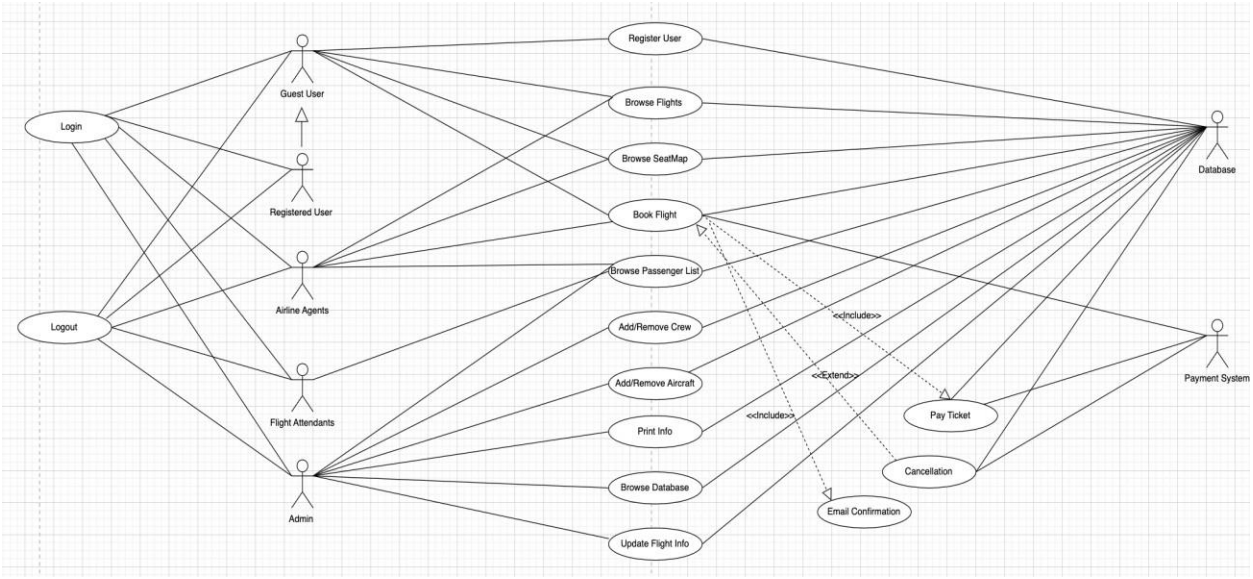
Textual Description

The flight reservation system is a web-based architecture. The system operates on a client-server model, in which a central server handles user interactions (client), enabling effective data flow and communication. React, Bootstrap CSS, and other technologies are used to create the website's front end, making it simple for users to browse flights, choose seats, and make reservations. The logic is handled by a strong server running Spring Boot in Java on the backend. It communicates with a MySQL database, which safely holds important data like user profiles, flight schedules, and payment histories.

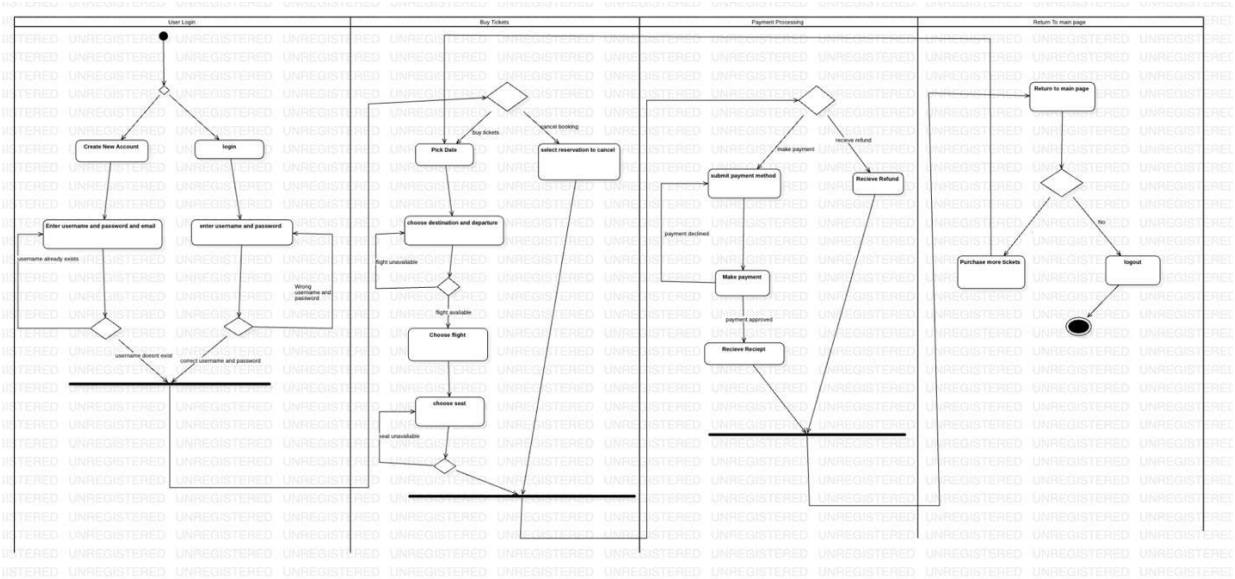
Graphical Presentation



Updated Use Case Diagram

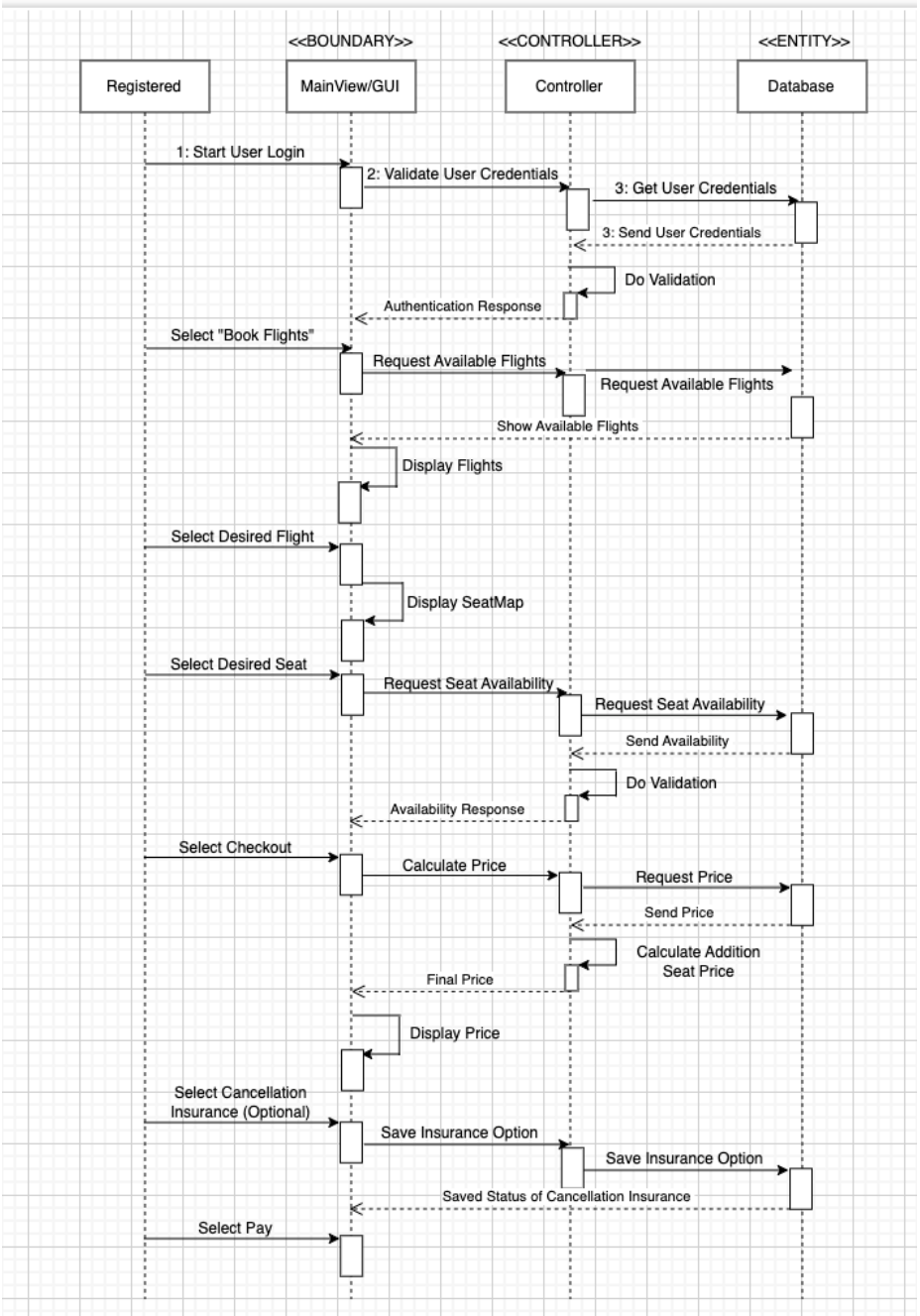


Systems activity diagram

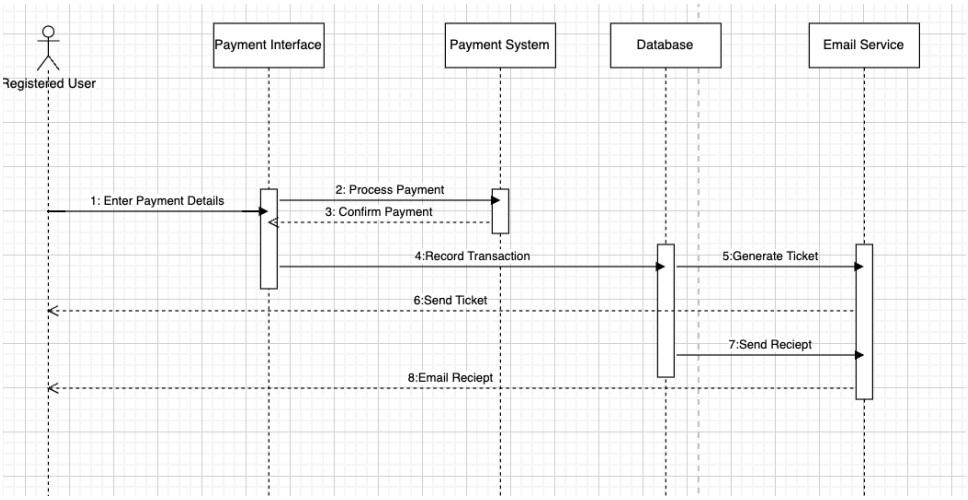


Sequence Diagrams

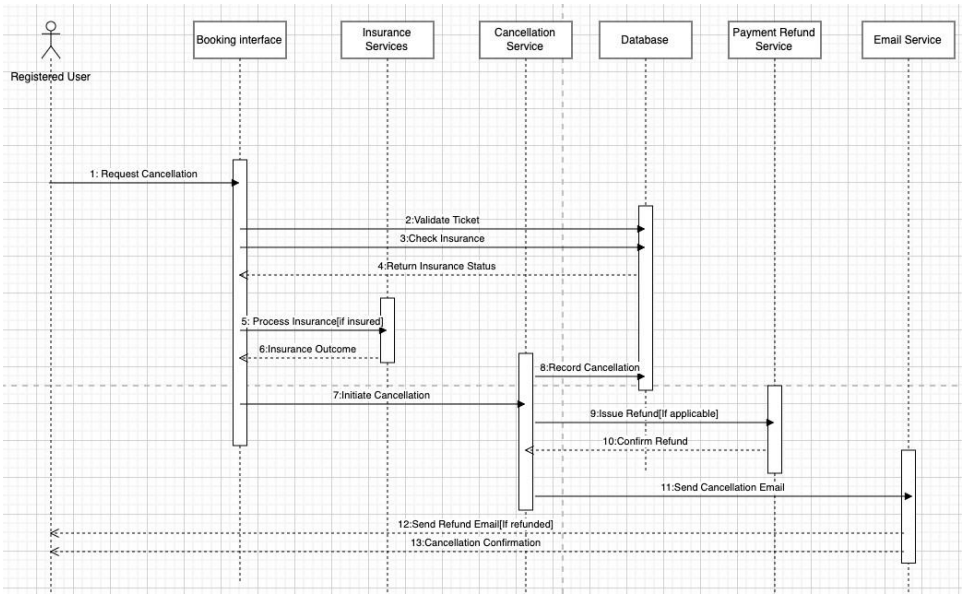
Login and Book Flight



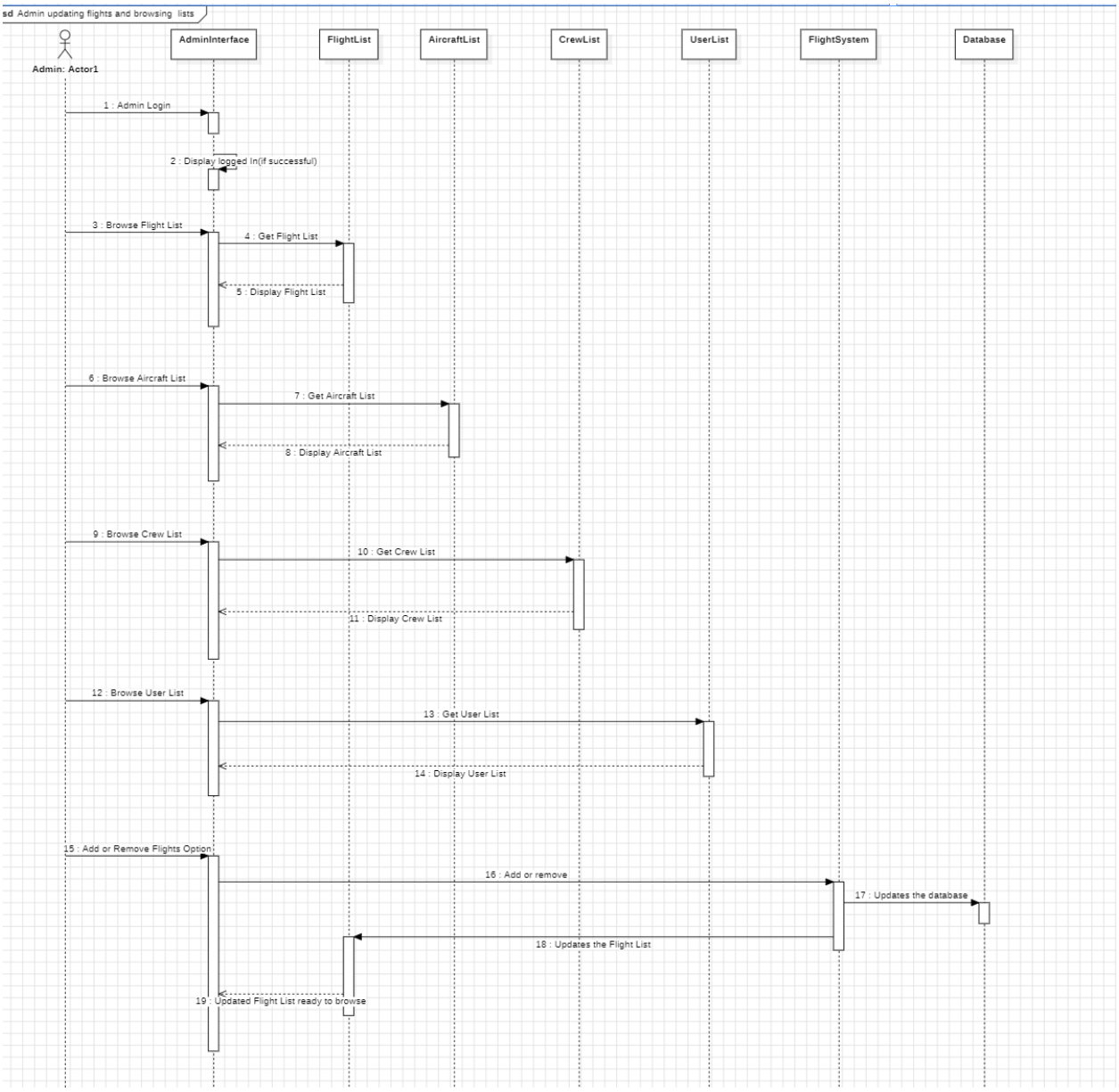
User Pays for A Ticket



User Cancels a Ticket

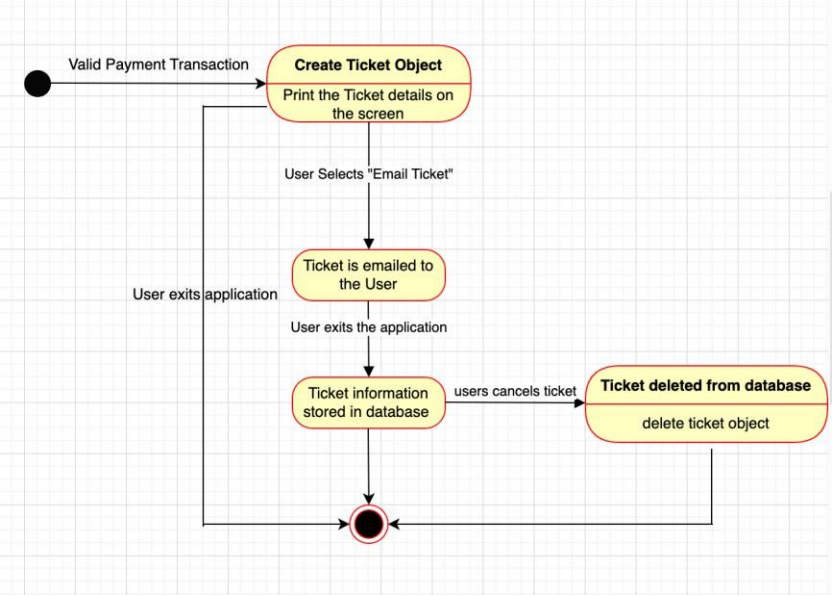


Admin updating flights and Browsing Lists

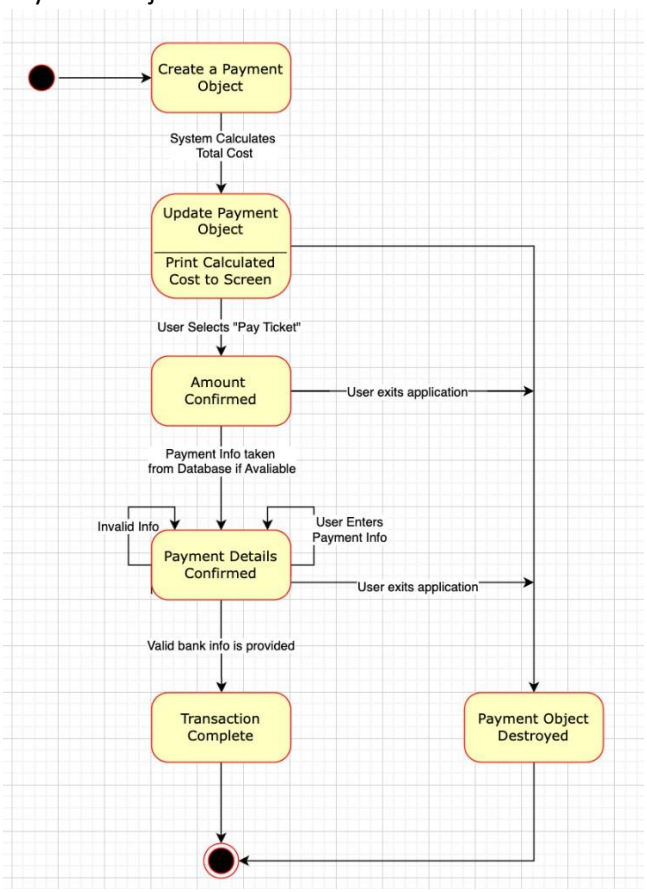


State Transition Diagram

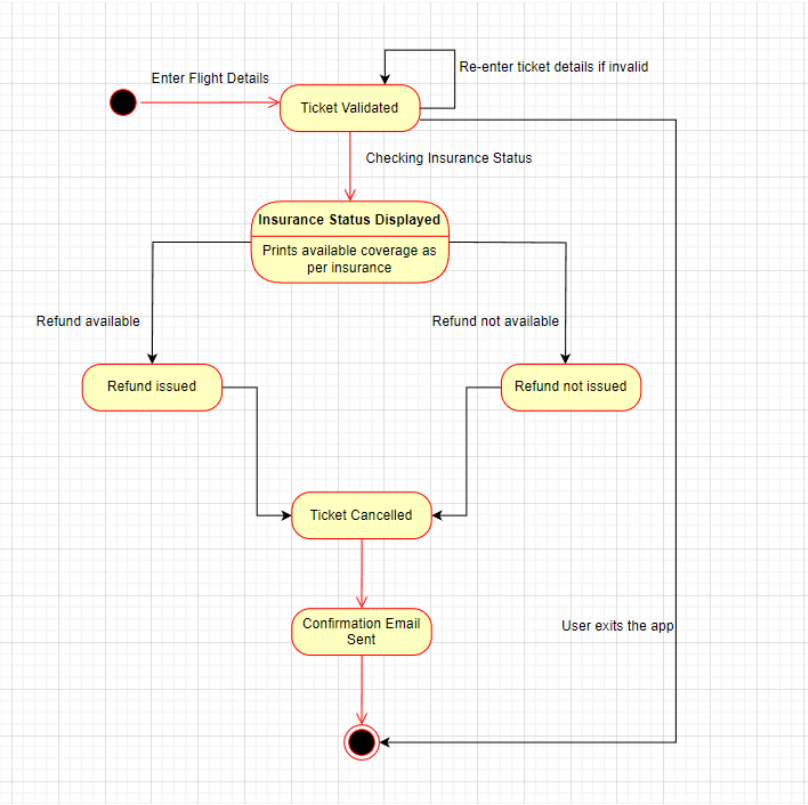
Ticket Object



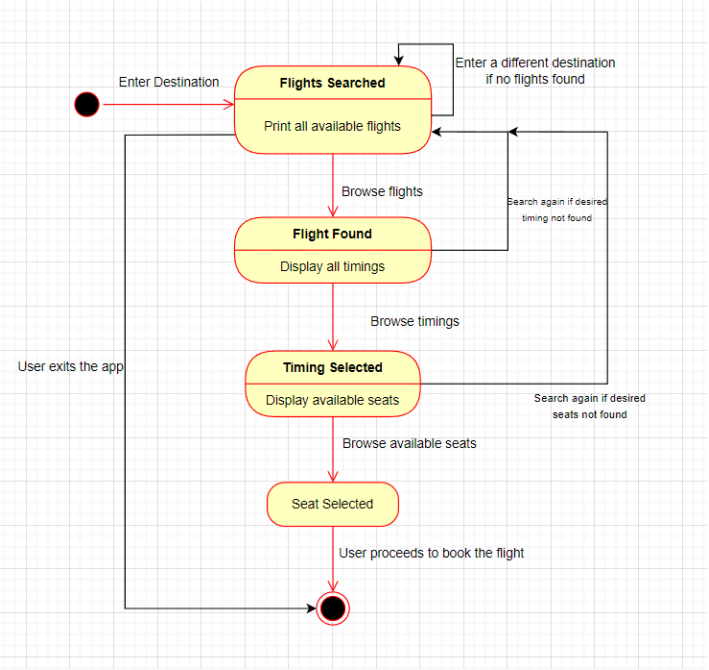
Payment Object



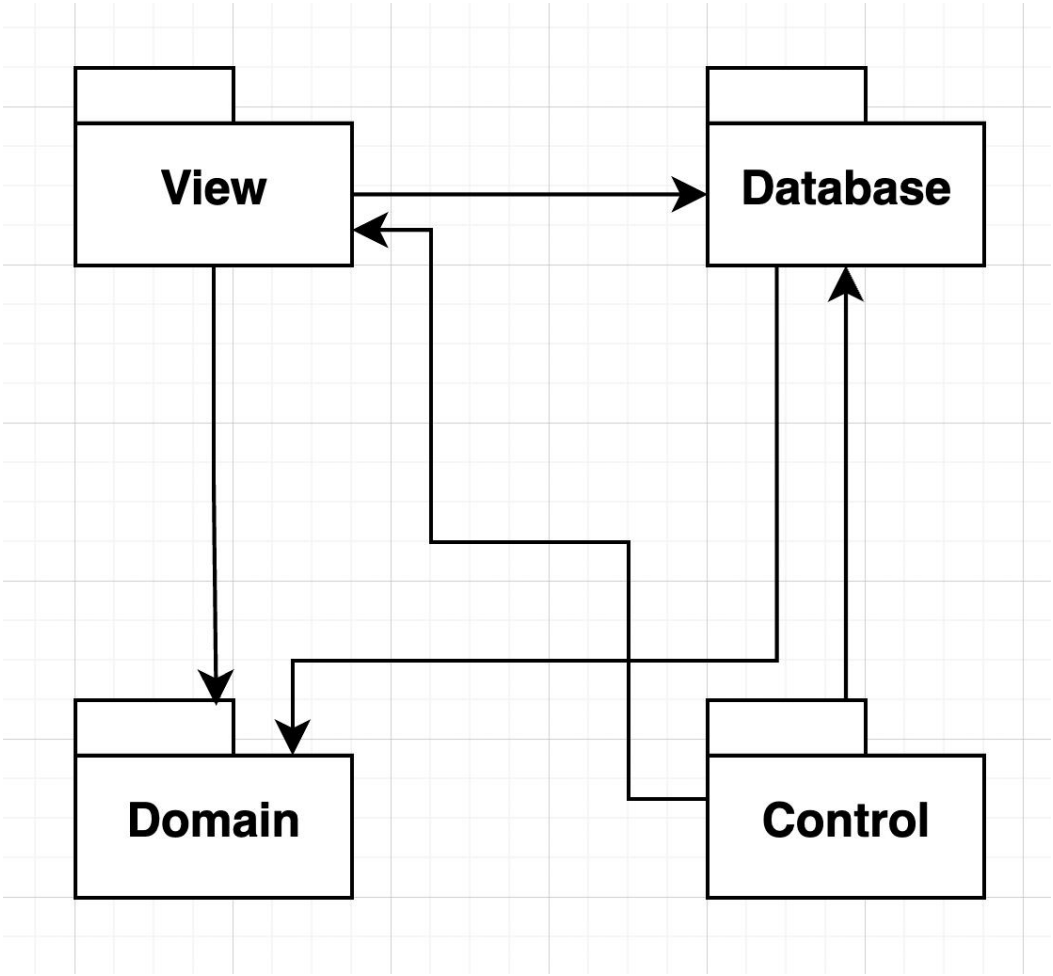
Cancel Ticket



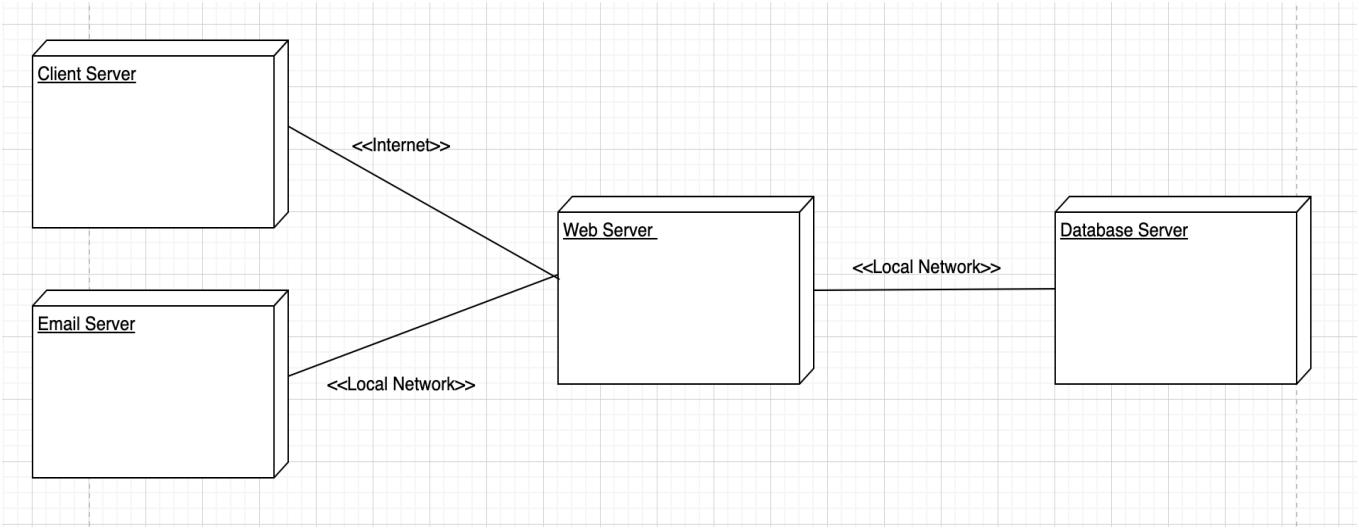
Book Ticket



Package Diagram



Deployment Diagram



The following diagrams are added as separate just for the sake of readability:

- System's Domain Diagram (without attributes and functionalities)
- System's Domain Diagram
- System's Activity Diagram
- Diagram for part C of the design phase (including all the boundary, entity and the controller classes)