Software Requirements and Design Document

For

Automate Airline Management System

Prepared by: Sheheryar Ramzan & Farquleet Farhat

NUCES

27/11/22

Table of Contents

1. Introduction

1.1 Purpose

AMS v1.2 by OptiReliSys is a system which will allow customers to book flight using this system. AMS will manage the bookings and flight schedule of an airline.

1.2 Product Scope

To create a system that will allow potential airline passengers of airline to

1. Search and book available flights
2. Select seats for their flights
3. Pay for the booking
4. Print their boarding pass only after payment has been made
5. Access aircraft maintenance reports
6. Notified of flight cancellation or delays

1.3 Title

Automated Airline Management System

1.4 Objectives

Aim of this project is to explore the current challenges of the airline management system and design an improved **automated** system that will allow passengers to book their flights with the follow objectives.

1. To design an improved AMS that will solve the problems being faced by the existing system which include:
    a. Inability of passengers to select preferred seats for their chosen flight
    b. No option of passengers printing their boarding pass
    c. No notification of passengers of flight cancellations or delays
    d. Lack of access to aircraft maintenance reports
2. To implement the design and create an airline system that captures self-booking
3. Our provided systems can either be installed at airline counters or accessed online. As a result, the check-in process speeds up and will be fully automated.

1.5 Problem Statement

We discovered that the Airline Management System used by Airlines offered the basic features of a reservation system such as searching for flights, selecting available flights, and paying for the

reservation. This discovery brought about the idea of creating an improved-automated management system. Some of the problems being faced by the existing system include

1. Passengers are unable to select their preferred seats through the current reservation system resulting in wastage of time at the check in counters.
2. No option of passengers printing their boarding pass from the existing system
3. Passengers are not notified in the event of flight cancellation or delays
4. No access to aircraft maintenance reports to ease passenger fears as regards to air travel and its disasters

2. Overall Description

2.1 Product Perspective

This product is replacement of current product.

2.2 Product Functions

Functions performed by product:

- *Management of booking of flight*
- *Management of flight schedule*
- *Check Status of flight*

2.3 List of Use Cases

1. Manage Flight Schedules
2. Manage Customers
3. Check Flight Status
4. Verify card details
1. Manage Booking(s)
2. Print Boarding Pass
3. Make Payment
4. Reserve seat

2.4 Extended Use Cases

Use Case Name	Manage Booking(s)
Scope of System Under Design	Automated Airline Management System
Level	User Goal

<table>
<tr><td>Primary Actor</td><td>Customer</td></tr>
<tr><td>Stakeholder & Interests</td><td>Customer: Wants fast service with minimal effort. Wants an easily visible display of flight schedules and respective prices. Wants proof of purchase to confirm booking.
Administrator: Wants to be able to quickly perform override operations, and easily debug customer problems
Company: Wants to accurately record bookings and satisfy customer needs. Wants automatic and fast update of booking inventory.</td></tr>
<tr><td>Pre-Condition</td><td>Customer is Identified and authenticated.</td></tr>
<tr><td>Post-Condition</td><td>Booking is saved. Seat status is updated. Booking inventory is updated.</td></tr>
<tr><td>Main Success Scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Responsibility</th></tr>
<tr><td>1. Customer opens website and logs into his/her account</td><td></td></tr>
<tr><td></td><td>2. System validates credentials of user</td></tr>
<tr><td>3. Customer requests a listing of available flights providing the criteria: date, departure airport, and arrival airport.</td><td></td></tr>
<tr><td></td><td>4. Based on criteria system returns a list of available flights.</td></tr>
<tr><td>5. Customer requests to manage one of his bookings, providing his identification number and payment information.</td><td></td></tr>
<tr><td></td><td>6. System confirms the flight availability, processes payment and records reservation.</td></tr>
</table>
</td></tr>
</table>

	7. System confirms customers about booking with a confirmation number.
Extensions	1. If the information entered during login is incorrect the system will generate an alert ● 1a. System asks user to re-enter login details or provides user with other options to sign in for example ID number. 4. No flights or seats are available for searched criteria ● 4a. System generates an alert 1. User needs to search again for available flights to book 2. System displays most relevant flights according to search details provided. 5. Invalid identification number and payment details ● 5a. System signals error and asks user to enter details again.

Use Case Name	Login and signup
Scope of System Under Design	Automated Airline Reservation System
Level	User Goal
Primary Actor	Customer
Stakeholder & Interests	**Customer:** Will be able to book their desired flight.
Pre-Condition	1. Active phone number 2. Active email account

<table>
<tr><td>Post-Condition</td><td>1. Customer status is updated.
2. User inventory is updated.</td></tr>
<tr><td>Main Success scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.Visit the website. And enter essential details for signing in.

3.Customer successfully signs up.</td><td>2.System verifies the customer credentials and saves the details in customer register.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>2. System verification failed and unable to sign in.</td></tr>
</table>

Use Case Name	Search a flight
Scope of System Under Design	Automated Airline Management System
Level	User Goal
Primary Actor	Customer
Stakeholder & Interests	**Customer:** Wants fast service with minimal effort. Wants an easily visible display of prices of respective tickets. **Administrator**: Wants to be able to quickly perform override operations, and easily debug customer problems
Pre-Condition	Website and customer's status must be active
Post-Condition	Searched flight is found
Main Success scenario	**Actor Action** / **System Response** 1.Customer visits website and login to his/her account

	3.Customer requests a listing of available flights providing the criteria: date, departure airport, and arrival airport. 2.System verifies users' credentials 4.Based on criteria, the system returns and presents a list of available flights.
Extensions	1. If the information entered during login is incorrect the system will generate an alert ● 1a. System asks user to re-enter login details or provides user with other options to sign in for example ID number 3. Requested flight from customer is not available ● 3a. System generates an alert for no flight found 1. User needs to search again for available flights to book 2. System displays most relevant flights according to search details

Use Case Name	Print Boarding Pass
Scope of System Under Design	Automated Airline Management System
Level	User Goal
Primary Actor	Customer
Stakeholder & Interests	**Customer**: Wants fast service with minimal effort. Wants proof of purchase to confirm booking.

<table>
<tr><td>Pre-Condition</td><td>Customer status is active and has already booked a flight</td></tr>
<tr><td>Post-Condition</td><td>Boarding pass is printed</td></tr>
<tr><td>Main Success scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1.Customer visits website and login to his/her account

3.Customer requests for his/her boarding card.

5.Customer leaves with boarding pass</td><td>2.System validates user credentials.

4.System process and generates boarding pass</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1. If the information entered during login is incorrect the system will generate an alert

● 1a. System asks user to re-enter login details or provides user with other options to sign in for example ID number

3. Boarding pass not found

● 3a. System signals error and asks user to book a flight first.</td></tr>
</table>

Use Case Name	Make Payment
Scope of System Under Design	Automated Airline Management System
Level	User Goal

<table>
<tr><td>Primary Actor</td><td>Customer</td></tr>
<tr><td>Stakeholder & Interests</td><td>1. Government Tax Agencies: Wants to collect tax from every ticket sale. Multiple agencies such as national, state, country.
2. Payment Authorization Service: Wants to receive digital authorization requests in the correct format and protocol. Wants to accurately account for their payables to the airline.
3. Customer: Wants fast service with minimal effort. Wants an easily visible display of prices of respective tickets. Wants proof as purchase receipt.
4. Company: Wants to accurately record transactions and satisfy customer interests. Wants to ensure that Payment Authorization Service payment receivables are recorded. Wants some fault tolerance to allow sales capture even if server components (e.g., remote credit validation) are unavailable. Wants automatic and fast update of accounting and inventory.</td></tr>
<tr><td>Pre-Condition</td><td>Customer has an active bank account card</td></tr>
<tr><td>Post-Condition</td><td>Ticket sales are saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. Payment authorization approvals are recorded.</td></tr>
<tr><td>Main Success scenario</td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1. Customer visits website and logs into his/her account.

3. Customer chooses flight to book</td><td>2. System verifies his/her credentials

4. System records the information and presents the description along with total price</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>6. Customer enter their payment details

11. Customer leaves with receipt</td><td>5. System asks user for the payment details

7. System validates bank card details

8. Handles payment (Fare deducted)

9. System logs completed sale and send sale and payment information to the external accounting system (for accounting and commissions) and inventory system (to update inventory).

10. System presents receipt</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>1. If the information entered during login is incorrect the system will generate an alert

● 1a. System asks user to re-enter login details or provides user with other options to sign in for example ID number.

3. Chosen flight has no seats available

● 3a. System generates alert and asks user to choose other flight

6. Invalid identification number and payment details</td></tr>
</table>

	• 6a. System signals error and asks user to enter details again. 7. System detects failure to communicate with bank service for validation of card details. 9. System detects failure to communicate with external tax calculation system service.

Use Case Name	Manage Flight Schedule
Scope of System Under Design	Automated Airline Management System
Level	User goal
Primary Actor	Administrator
Stakeholder & Interests	Administrator: ease and convenient way to deal with flight records.
Pre-Condition	1.Essential credentials for login.
Post-Condition	1.Update the flight schedule catalog. 2.Make the changes visible to the other users.
Main Success scenario	

Actor Action	**System Response**
1. Administrator opens website and logs into his/her account.	
	2.System verifies the credentials and redirects the Admin to the Administrator panel.
3.Administrator schedules a new flight with necessary	

	details. | 4.Flight schedule register is updated, and changes are visible to other users.
Extensions	1.Administrator does not have the credential. ● 1a. Unable to manage flight schedule. 2.Verification of credentials fails due to system failure. ● 2a. Unable to manage flight schedule.

Use Case Name	Manage customers
Scope of System Under Design	Automated Airline Management System
Level	User goal
Primary Actor	Administrator
Stakeholder & Interests	**Administrator:** Ease and convenient way to deal with customer records. **Customer:** Becoming the customer of the airline and being informed about future deals.
Pre-Condition	1. Essential Credentials. 2. Customer's information.
Post-Condition	1. Update Customer register. 2. Informing customers about changes if any.
Main Success scenario	**Actor Action** | **System Response** 1.Administrator opens website and logs into his/her account. | 2.System verifies the

	3.create a new customer record through the details provided during the sign in. .	credentials and redirects the Administrator to the Administrator panel. 4.Update the customer register.
Extensions	1.Administrator does not have the credential. ● 1a. Unable to manage customers. 2.Verification of credentials fails due to system failure. ● 2a. Unable to manage customer.	

Use Case Name	Check Flight Status
Scope of System Under Design	Automated Airline Management System
Level	User goal
Primary Actor	Customer
Stakeholder & Interests	Customer & Staff: to know about the status (delayed/on time) of their flight.
Pre-Condition	1.Logged into account. 2.Booked or reserved a ticket.
Post-Condition	
Main Success scenario	

<table>
<tr><td></td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>1. User opens website and logs into his/her account.

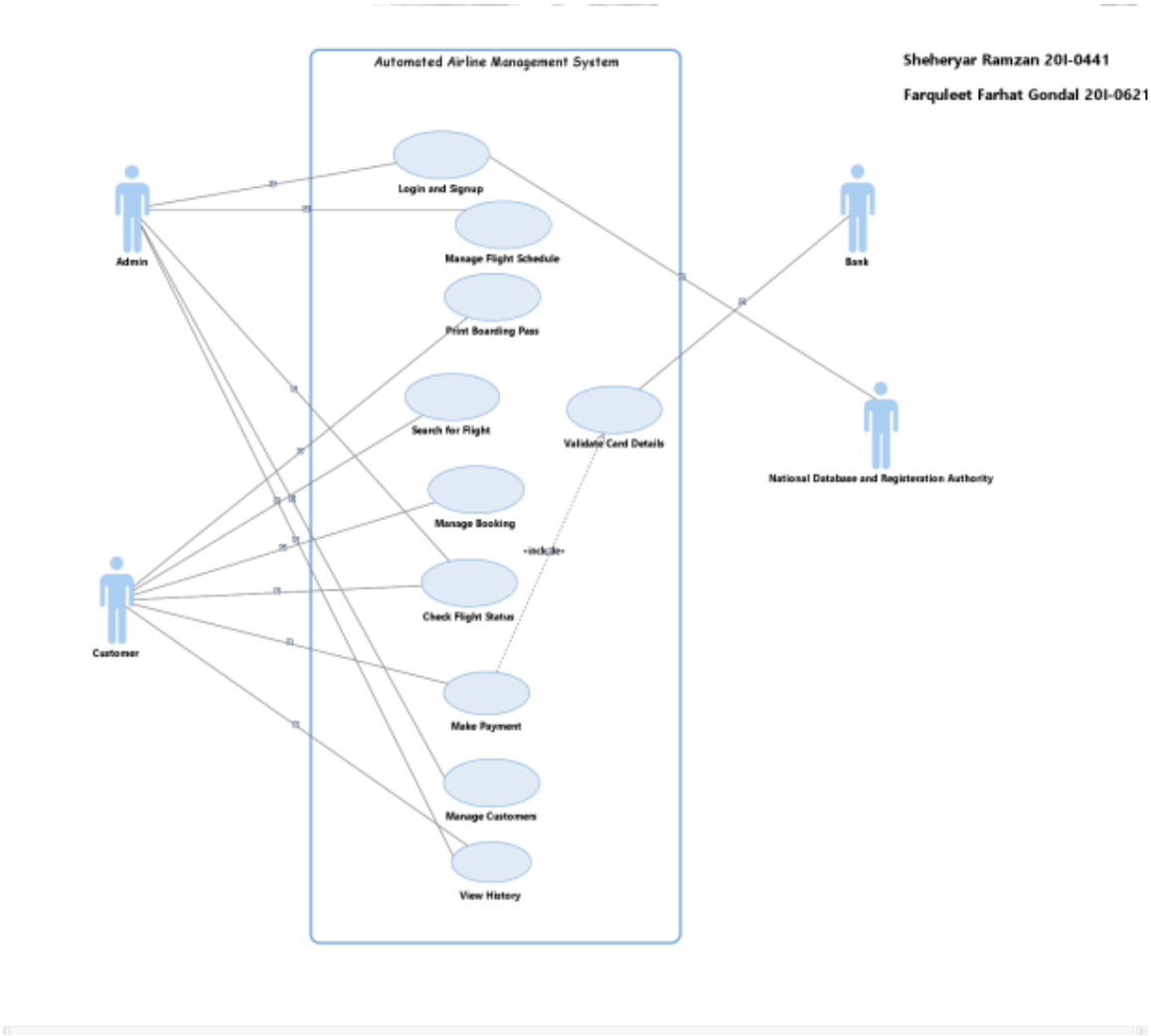
3.Search for a booked flight.

5.Check the status of the flight by clicking on it.</td><td>2.System verifies credentials.

4.Shows all booked flights.

6.Shows the current status of the flight.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>2. System verification fails due to system failure.
● 2a. Unable to login to check status.
3. Doesn’t have a flight booked
● 3a. Unable to check status.</td></tr>
</table>

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

- *Response Time: response time of all functions is good.*
- *Portable: As program is coded in java, which is cross platform language, so program is not architecture specific.*
- *User Friendly: UI is user friendly and simple to make it easier for the customer to interact twit the program.*

3.2 Safety Requirements

No possible loss, damage or harm can be done due to use of this product.

3.3 Security Requirements

Program is coded using the encapsulation concept of OOP which make it impossible for people to messing up your code without going through proper channel.

3.4 Software Quality Attributes

- *Reliability:*
  Main component of reliability of program is database. Database is continuously updated, which keeps backup.
- *Supportability:*
  UI is simple and user friendly which it easier to navigate through. Program is developed in java making easier to run due to independence of architecture.

3.5 Business Rules

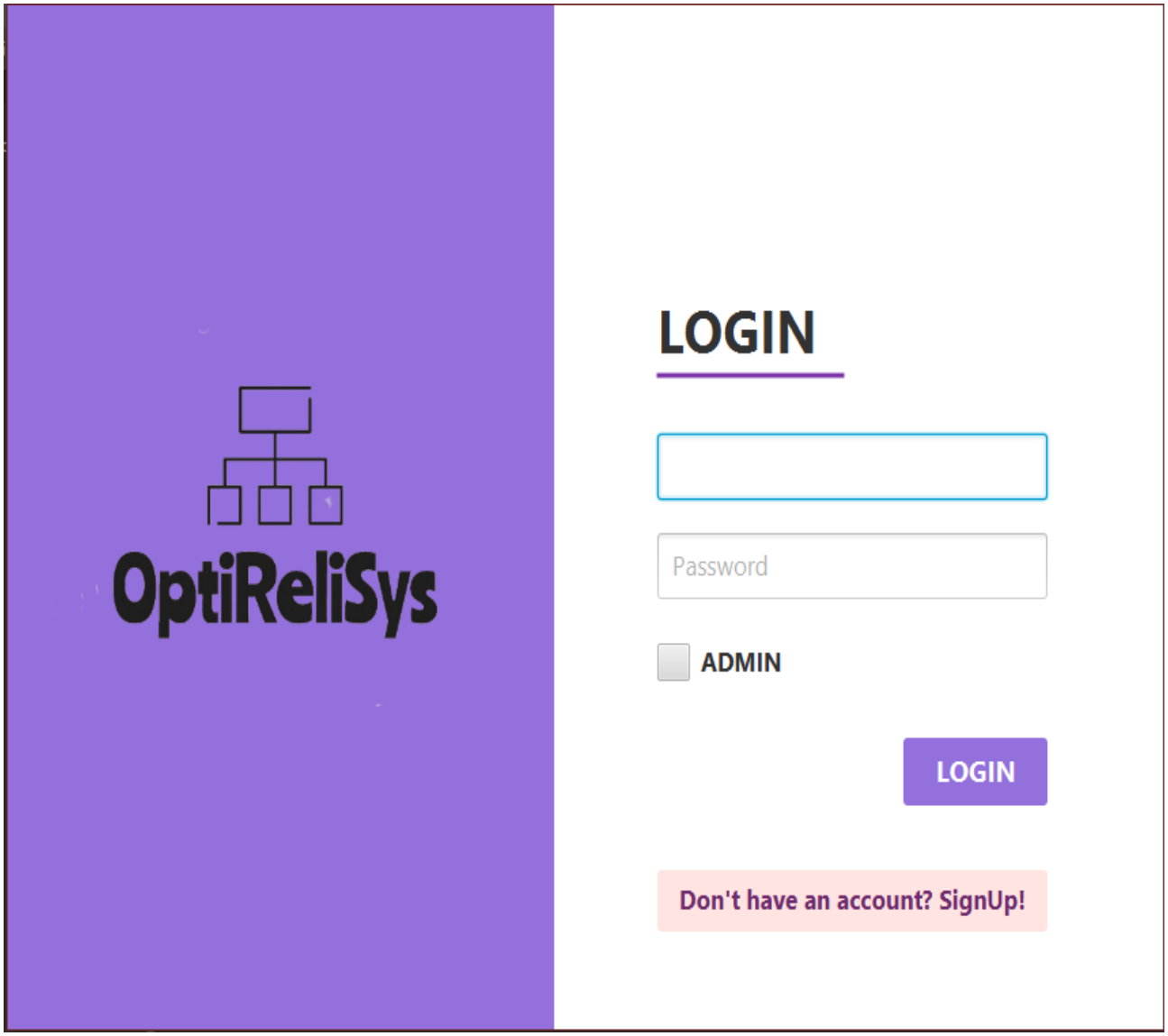
- *Customers can manage flight bookings, view, check status and delete them*
- *Administrator can manage flight schedule, view, and delete them*

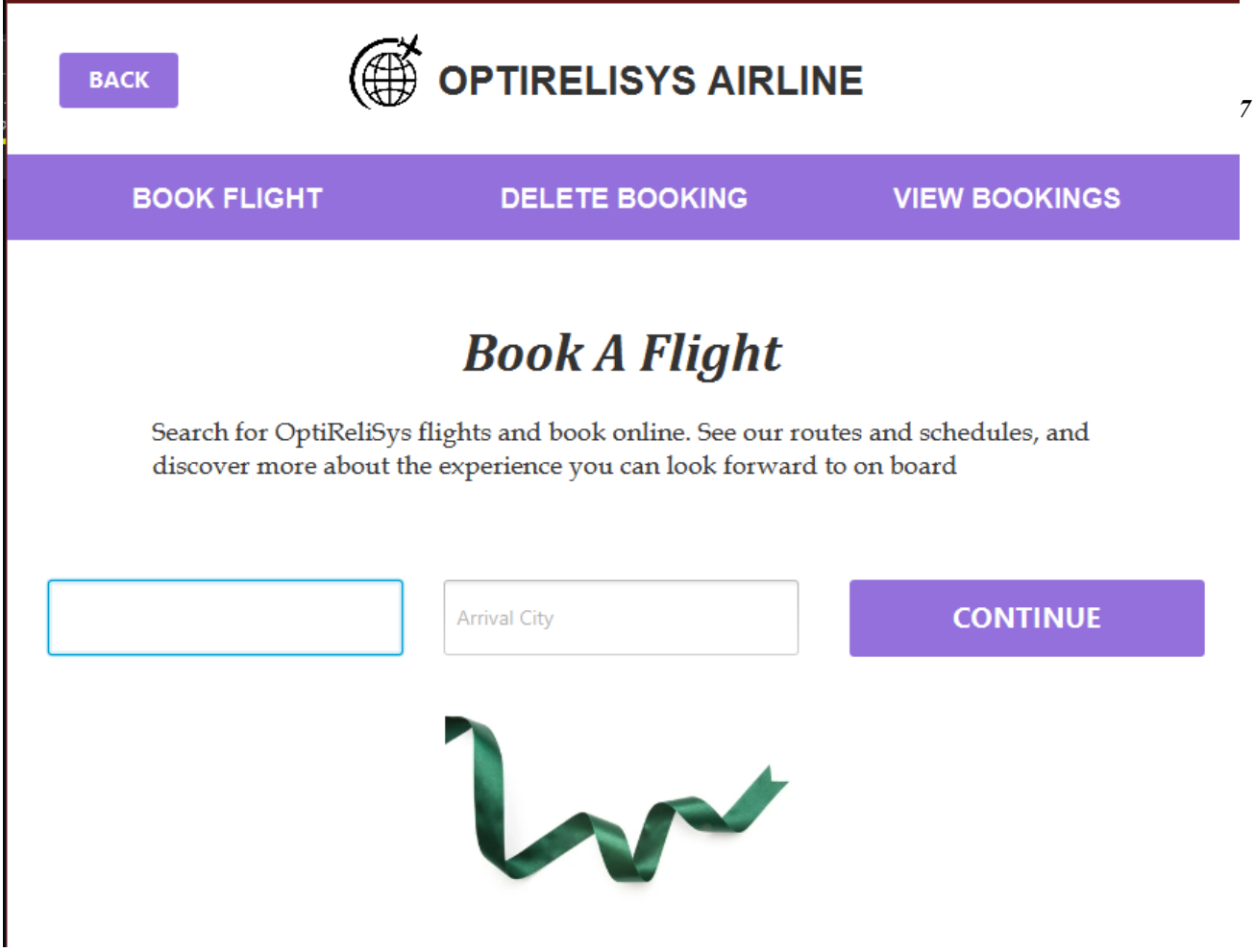
3.6 Operating Environment

Basic requirement to run this system:

- Oracle
- Ojdbc11 package
- Windows Server 2008
- RAM I Gb
- Processor Pentium IV

3.7 User Interface

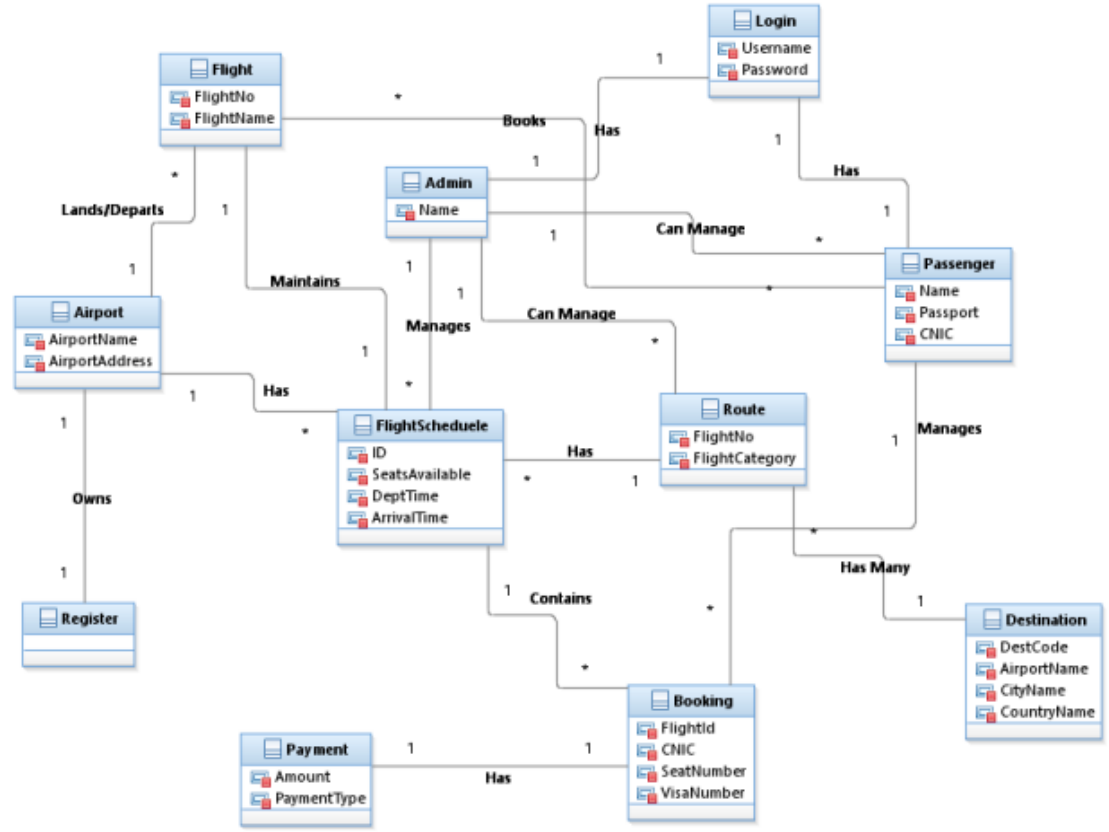




Every page in program is linked to each other. Once you login as a customer. You are redirected to Customer Home page where you can choose between booking a new flight or delete or view already booked flight.

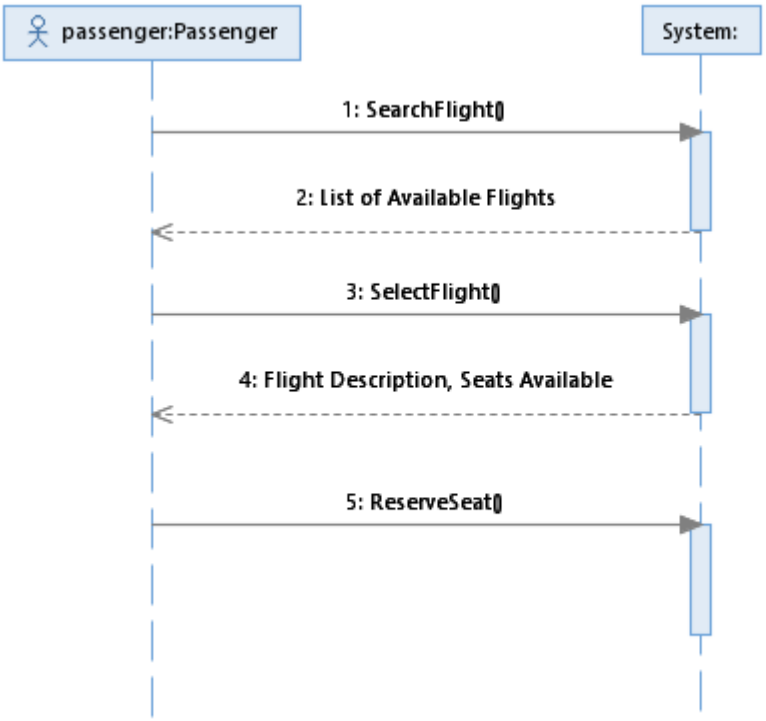
4. Domain Model

Domain Model

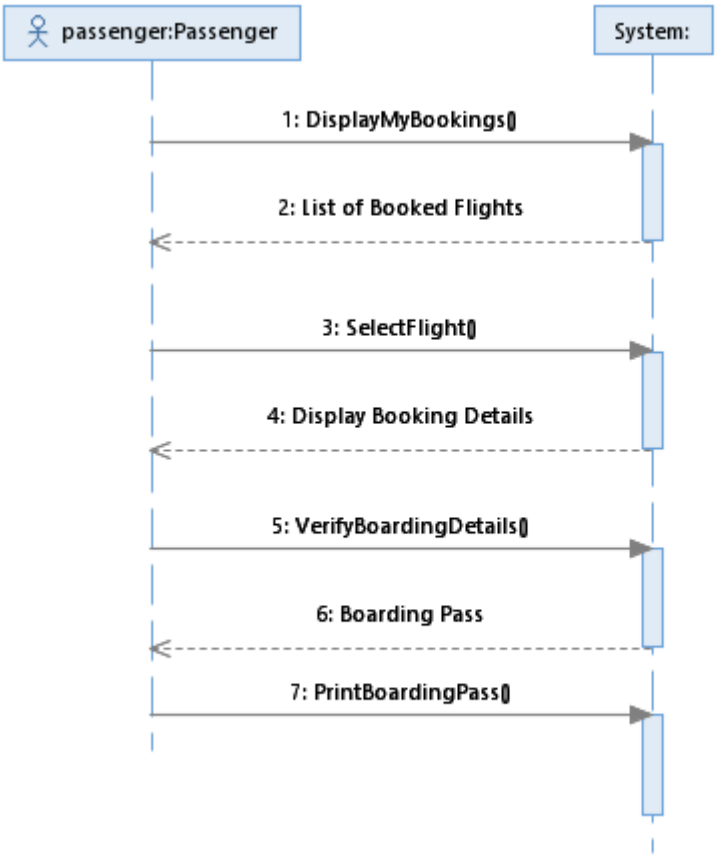


5. System Sequence Diagram

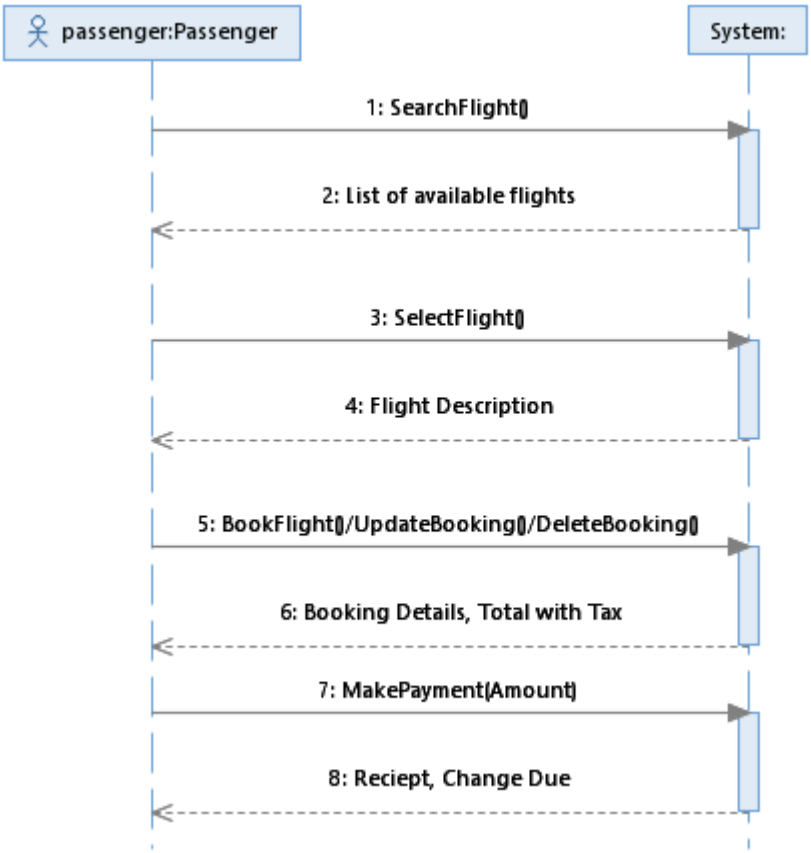
1. Reserve Seat



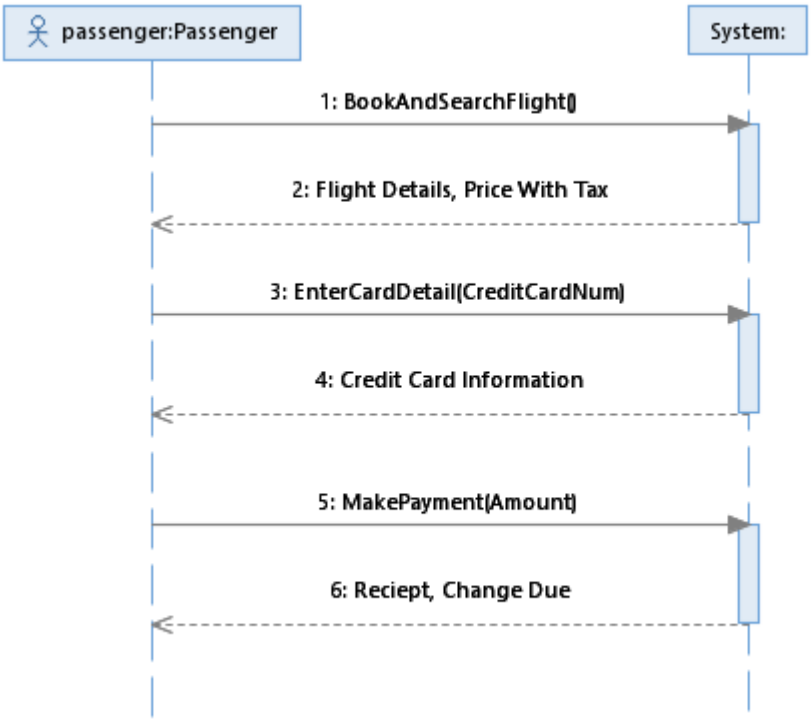
2. Print Boarding Pass



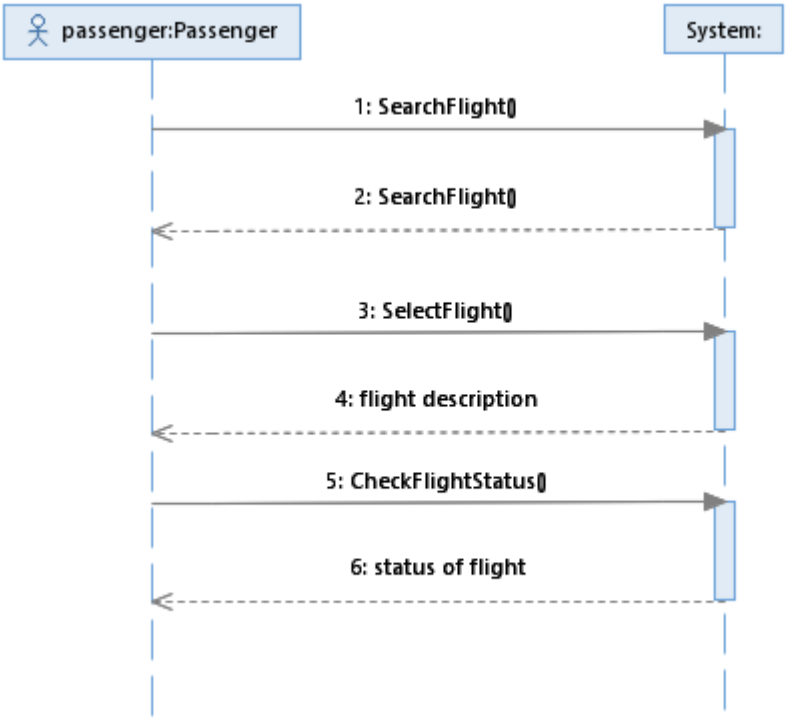
3. Manage Bookings



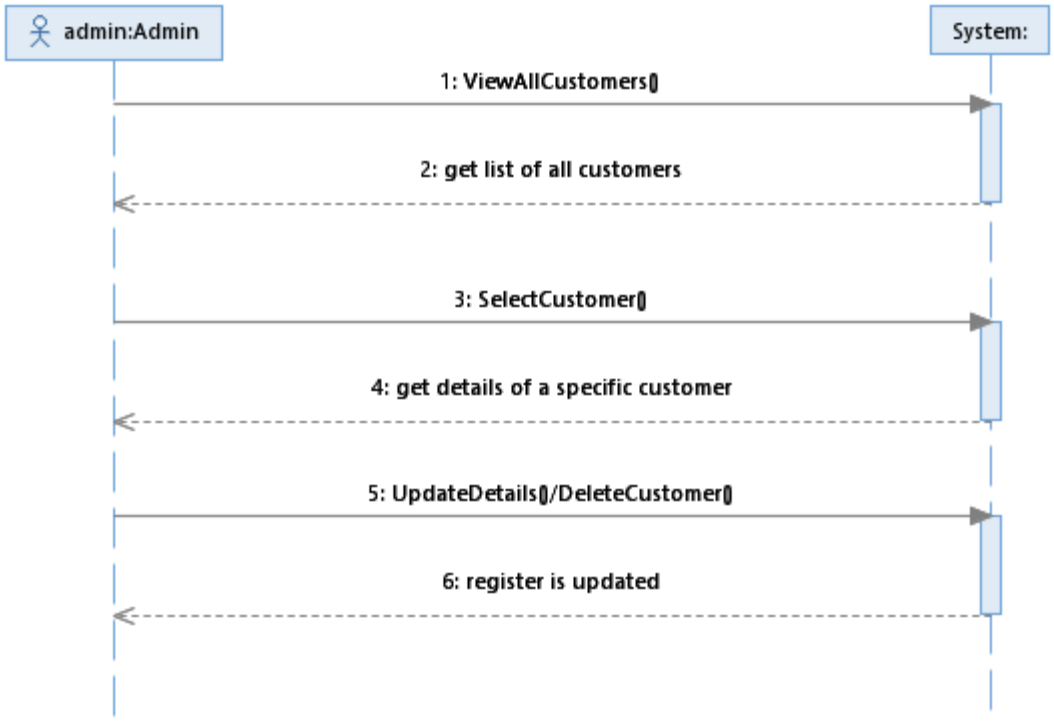
4. Make Payment



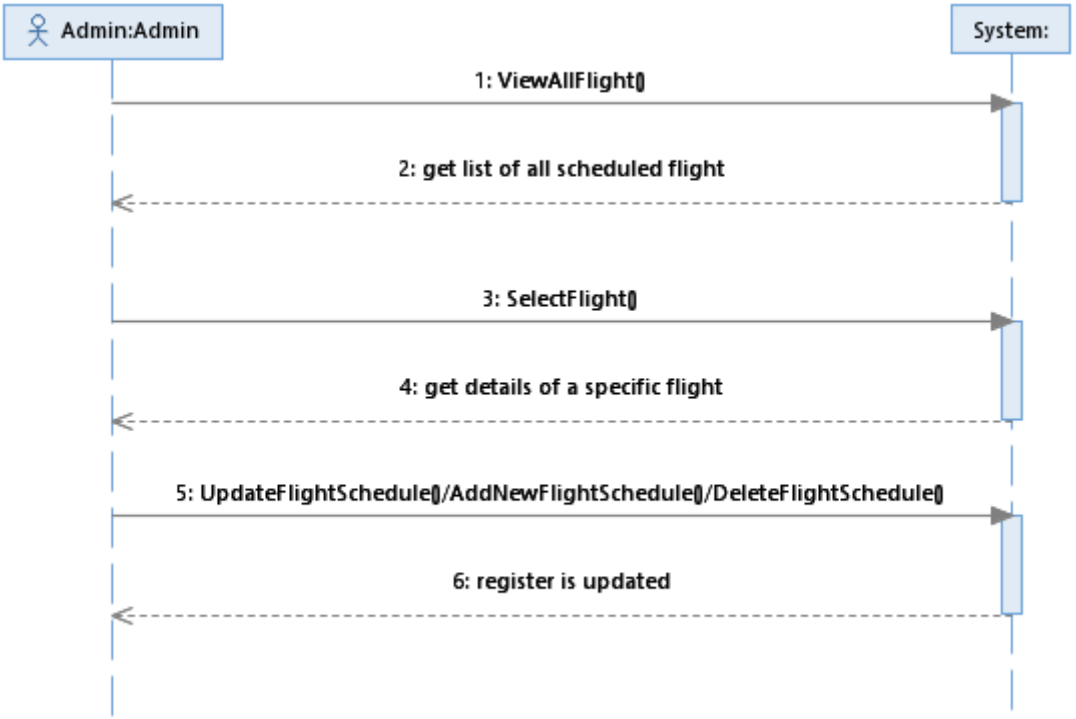
5. Check Flight Status



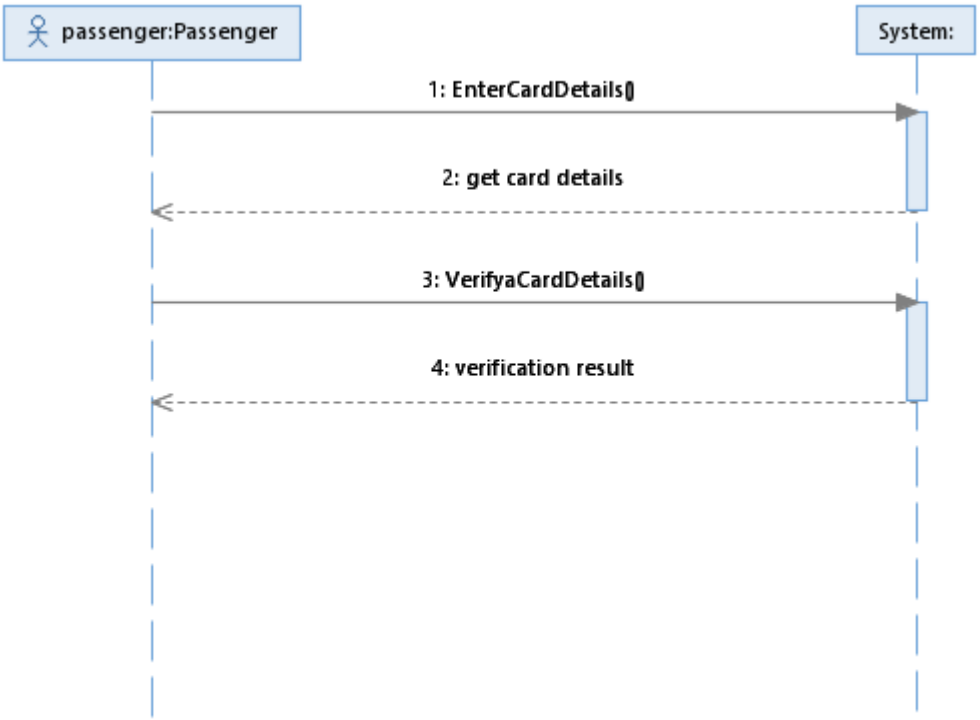
6. Manage Customers



7. Manage Flight Schedule

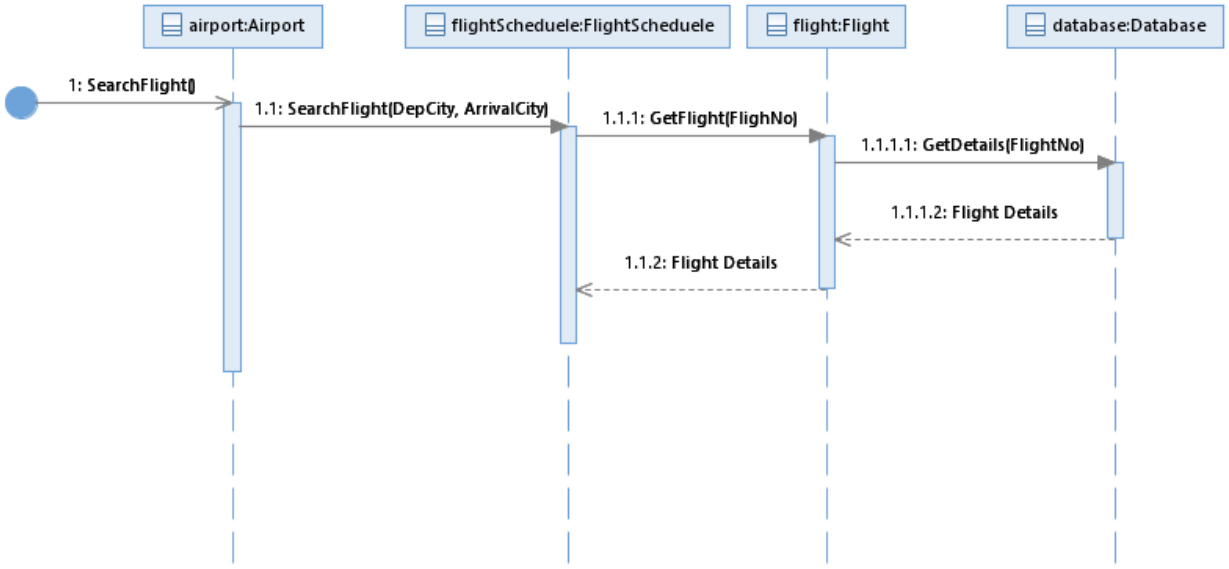


8. Verify Card Details

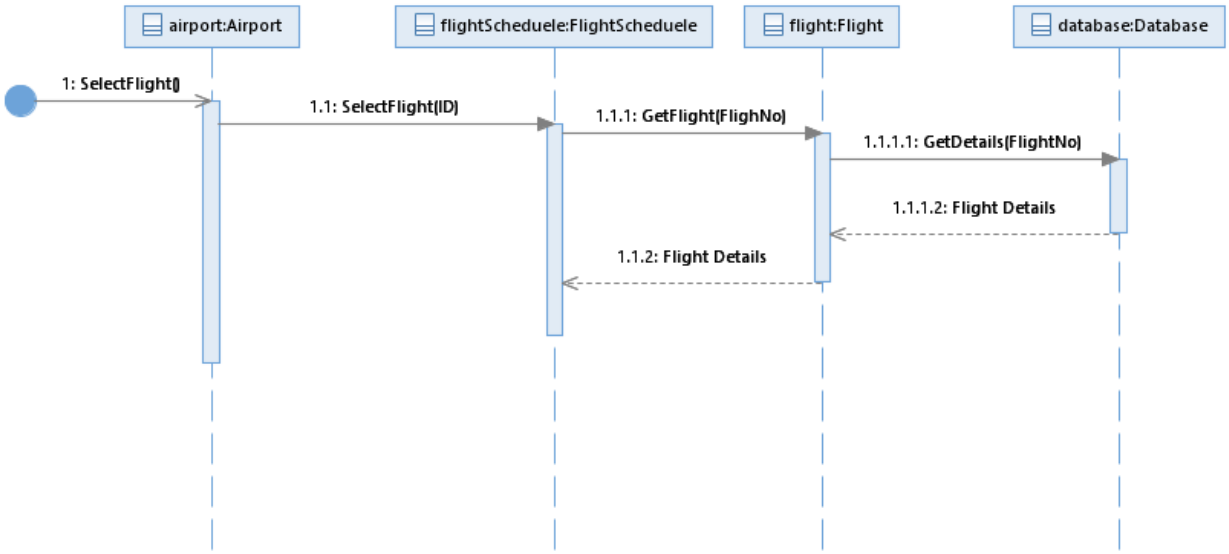


6. Sequence Diagram

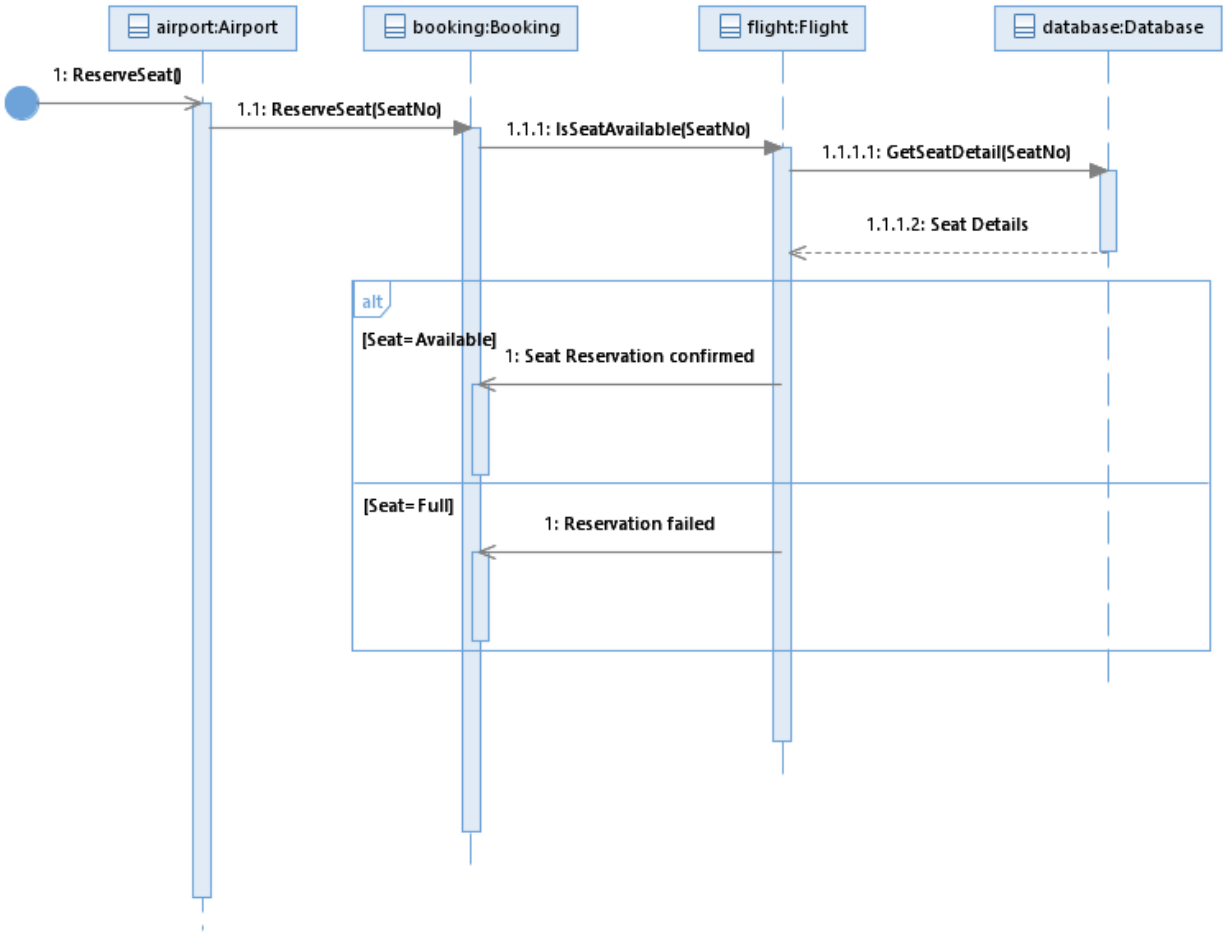
1. **Search Flight**



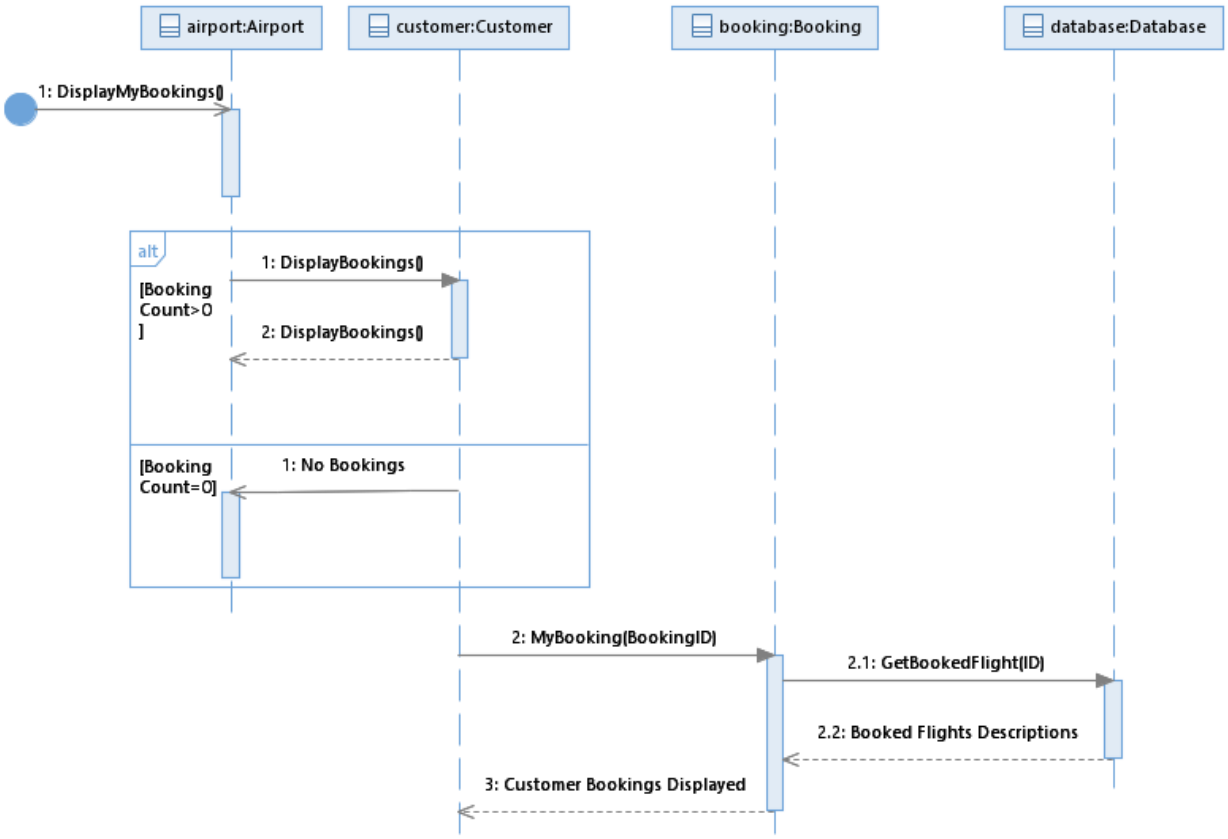
2. Select Flight



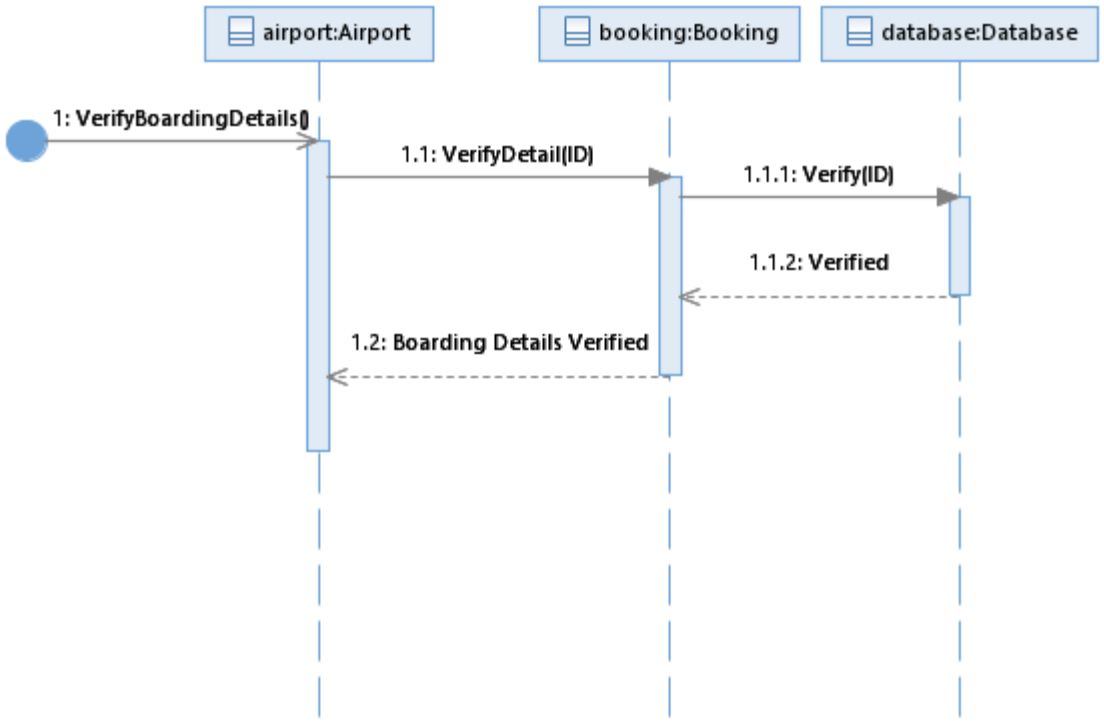
1. Reserve Seat



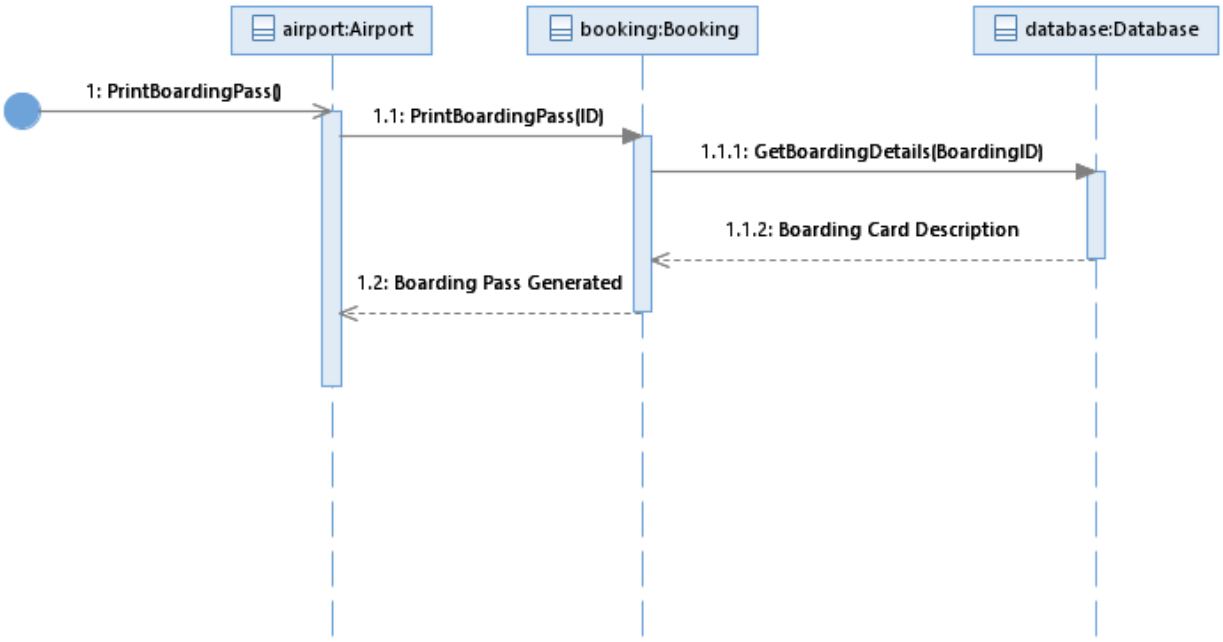
2. DisplayMyBookings



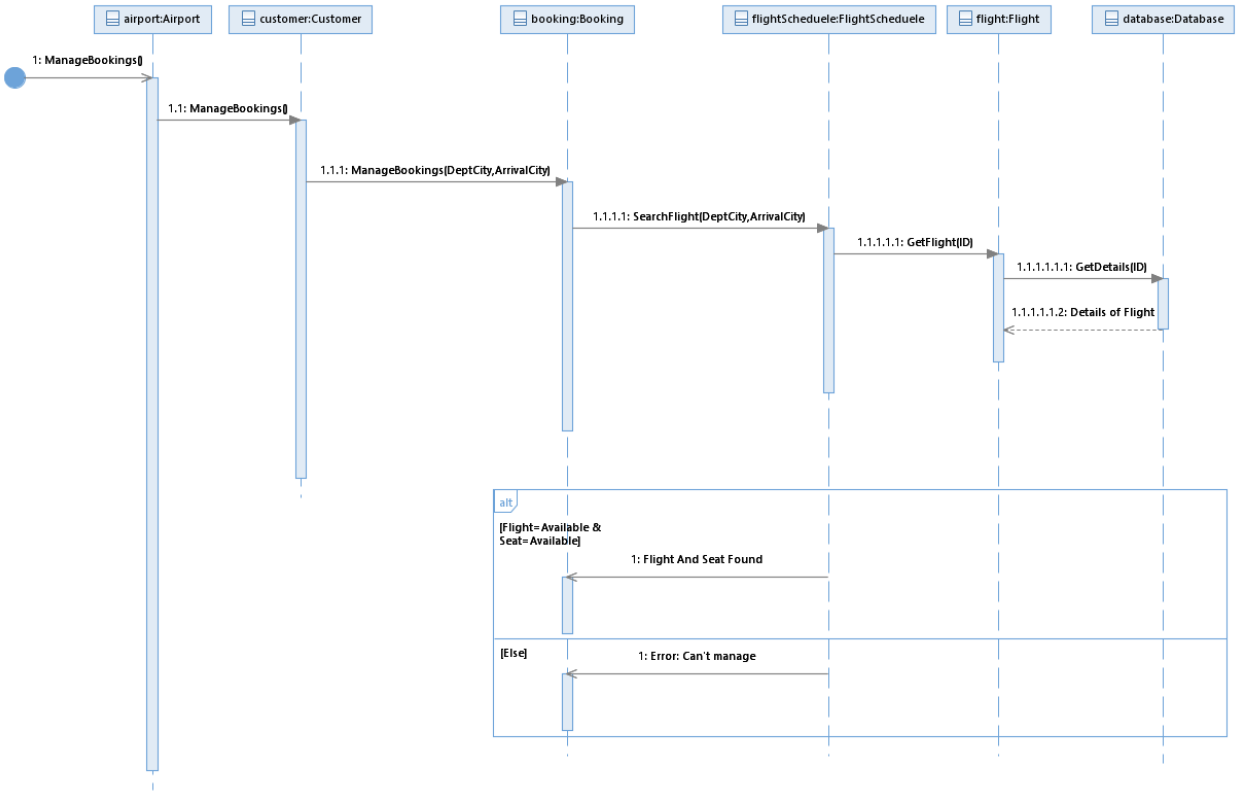
3. VerifyBoardingDetails



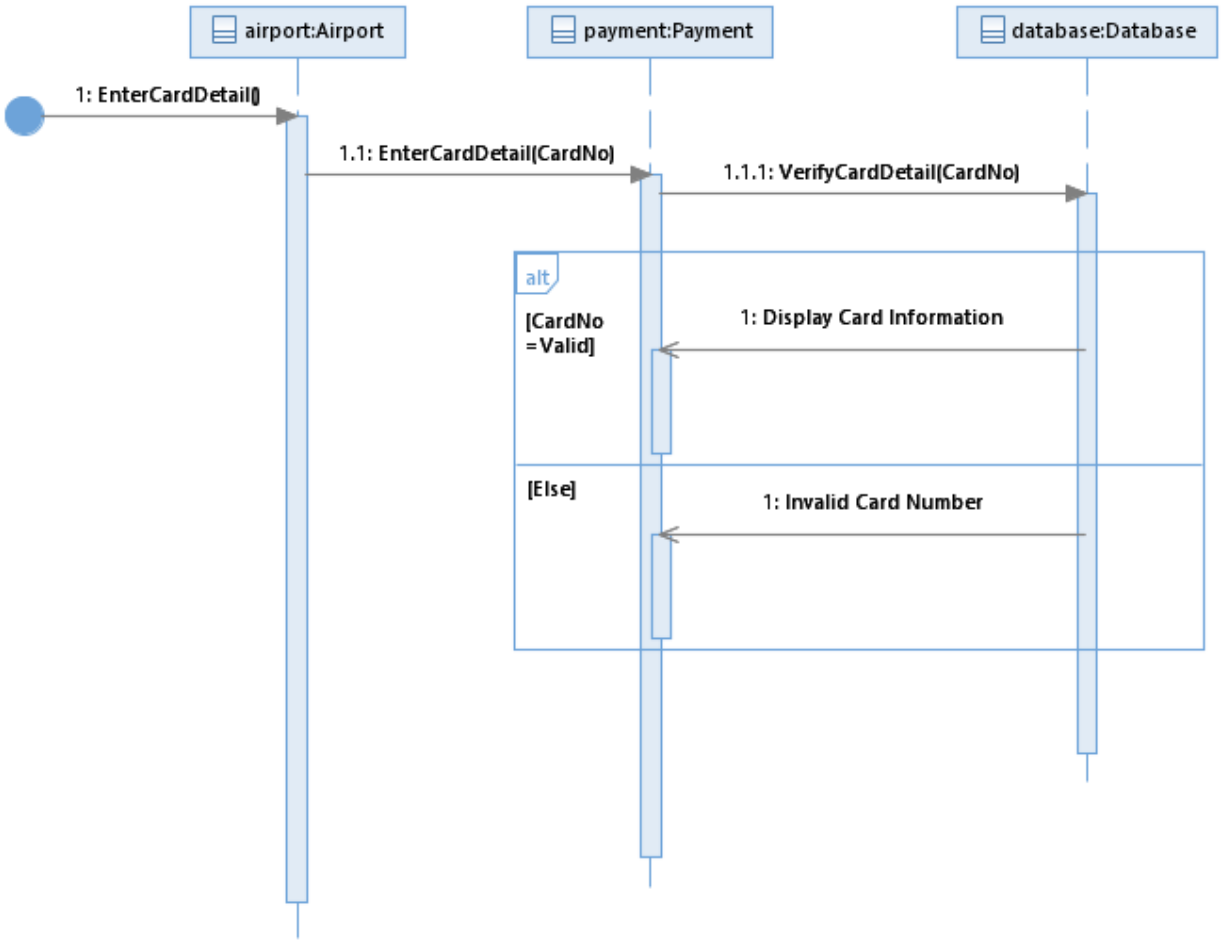
4. PrintBoardingPass



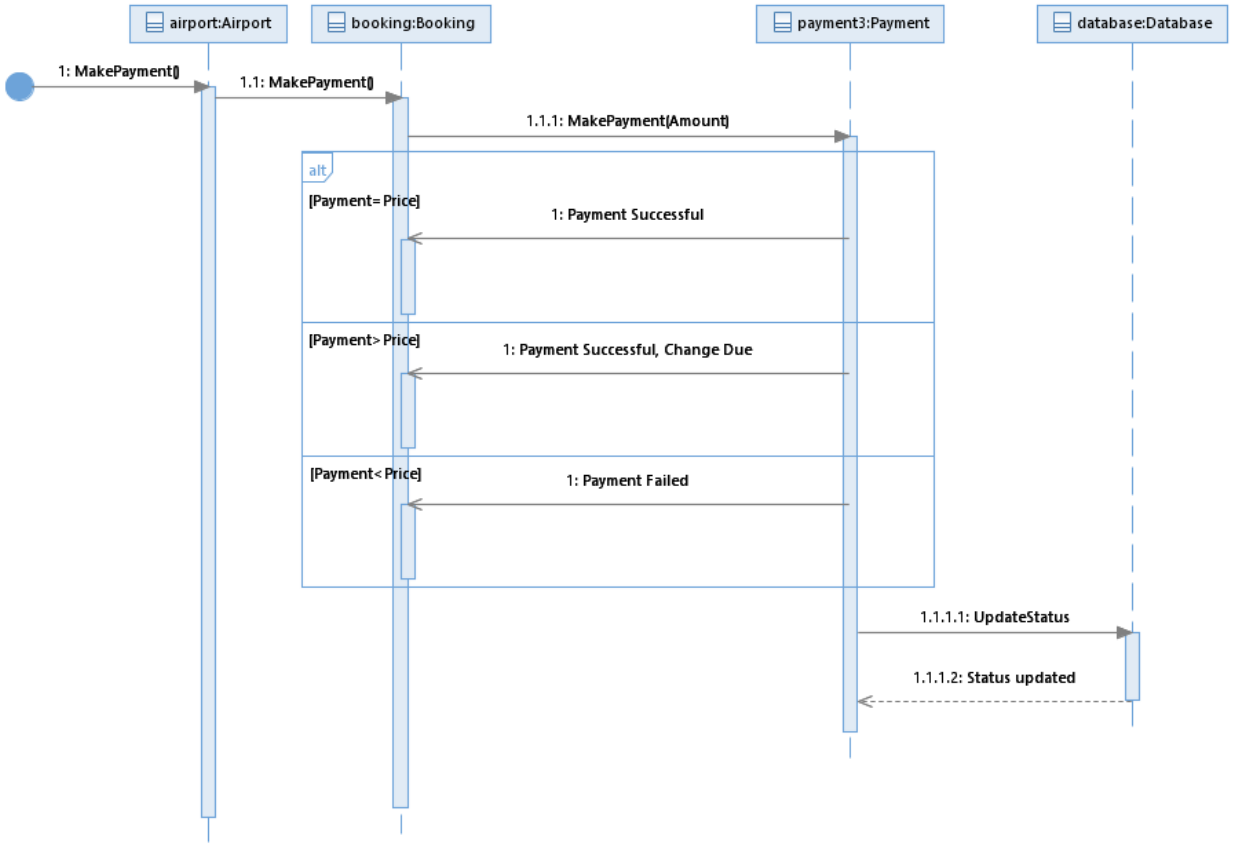
5. ManageBookings



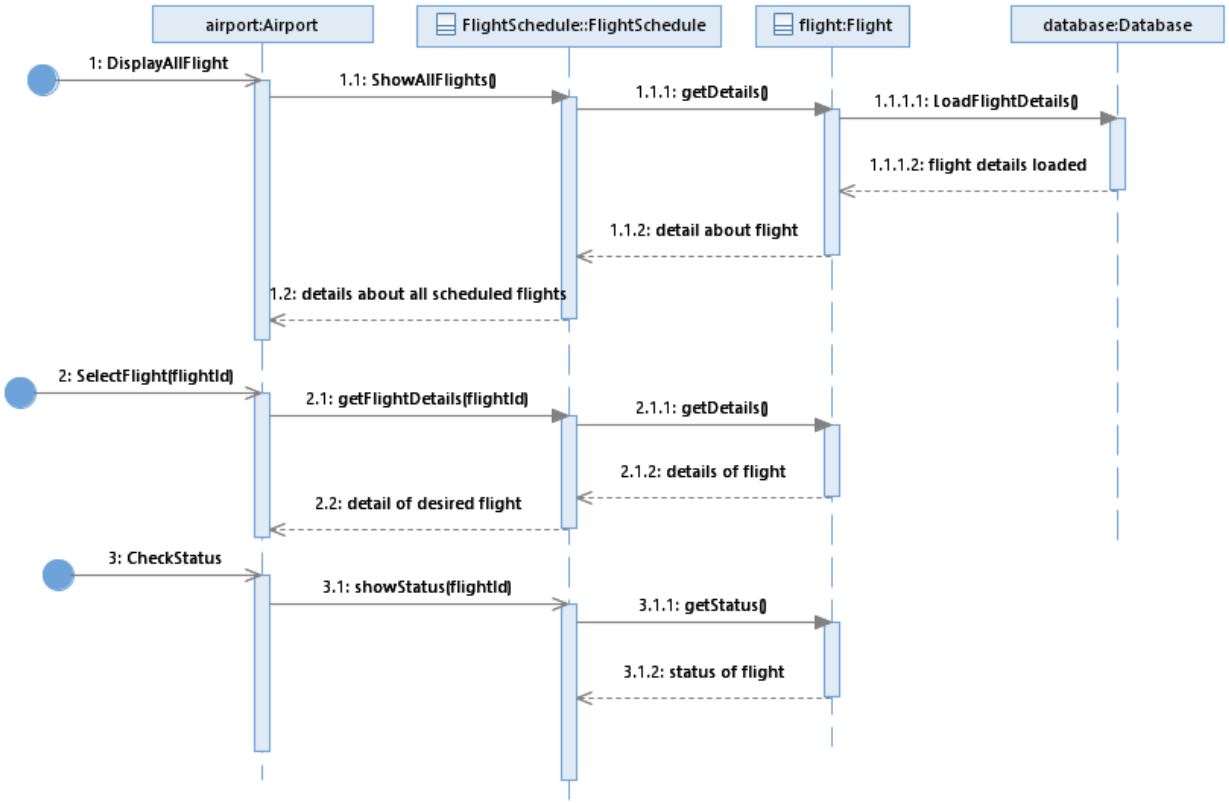
6. EnterCardDetails



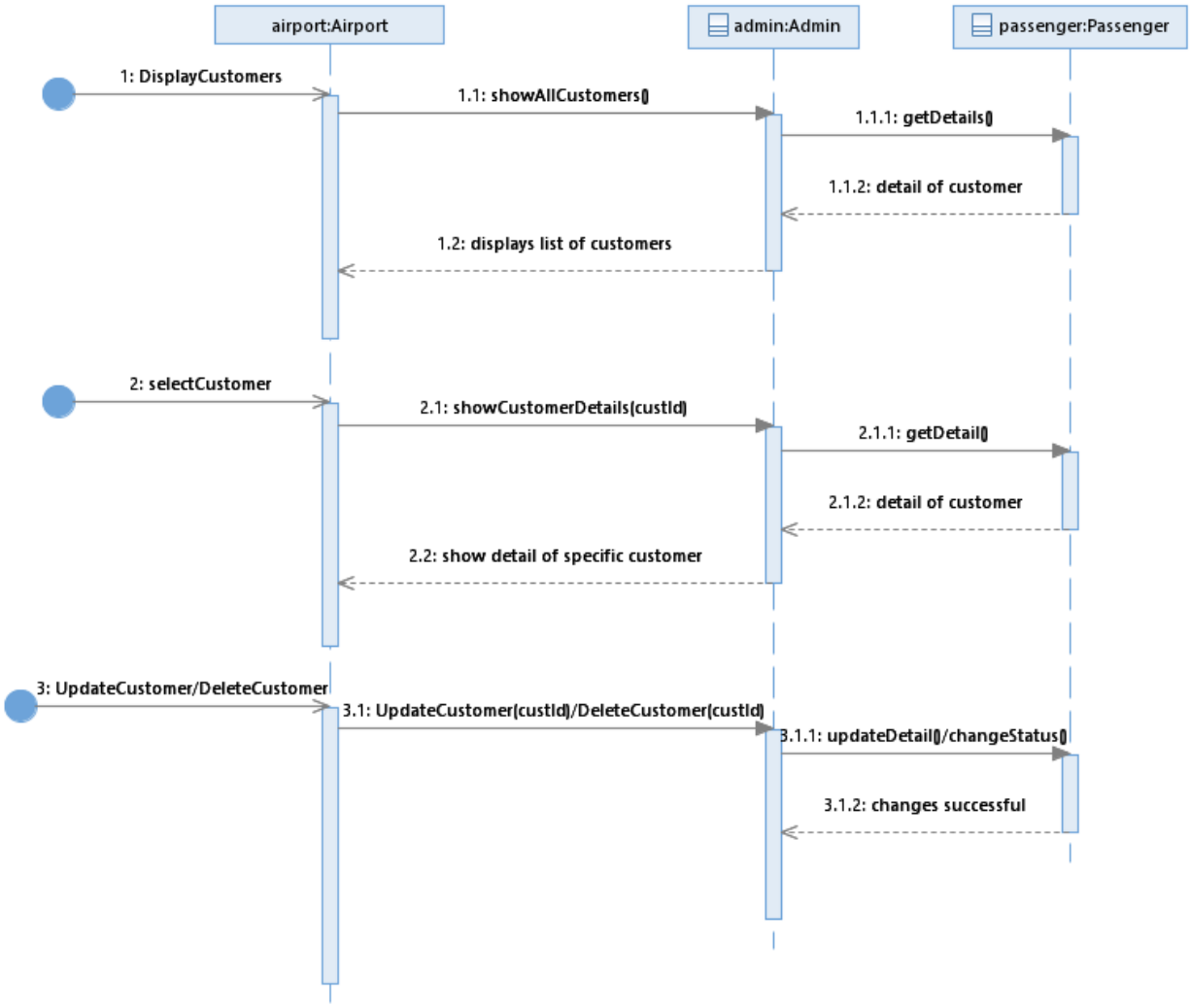
7. MakePayment



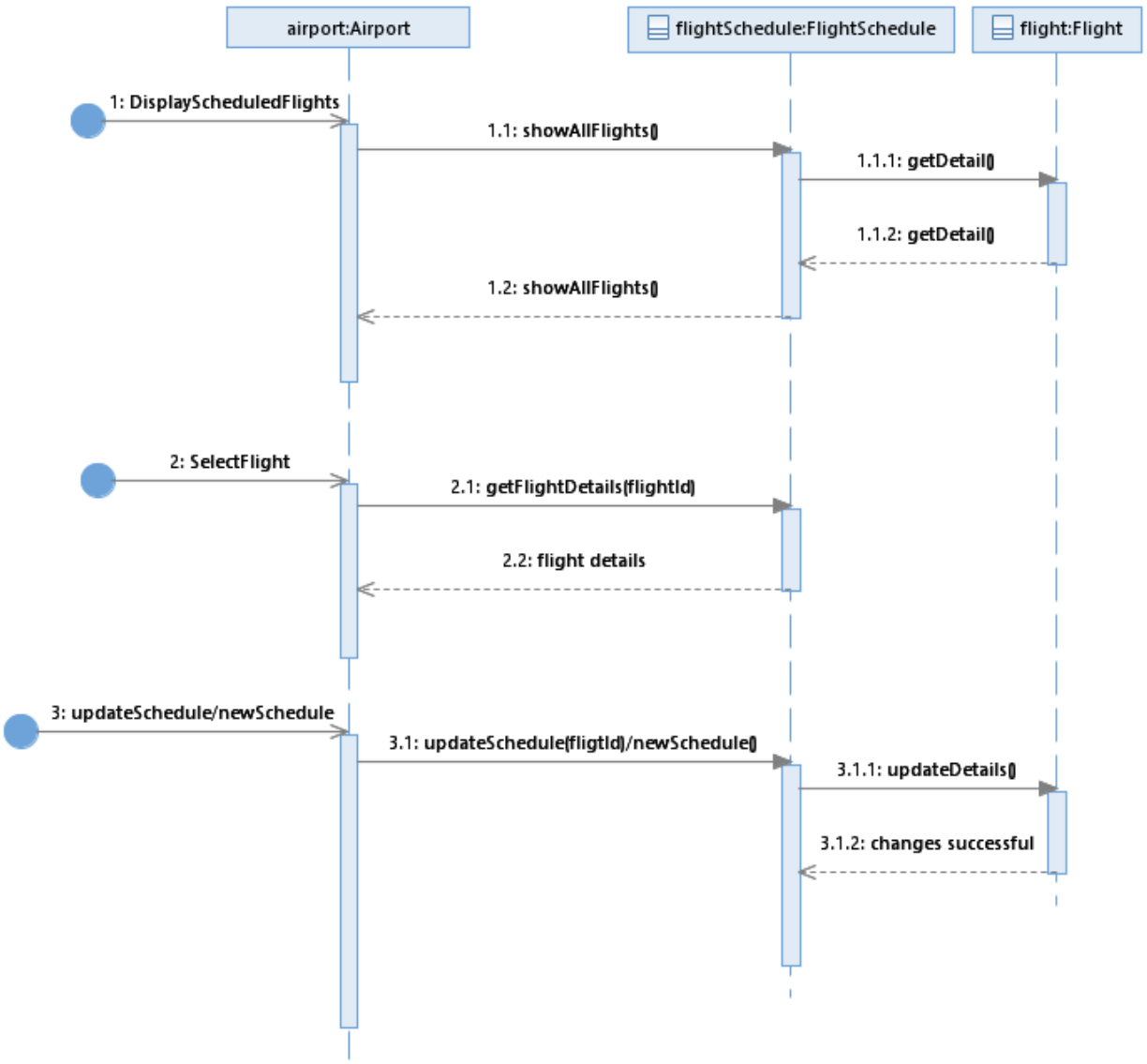
CheckFlightStatus:



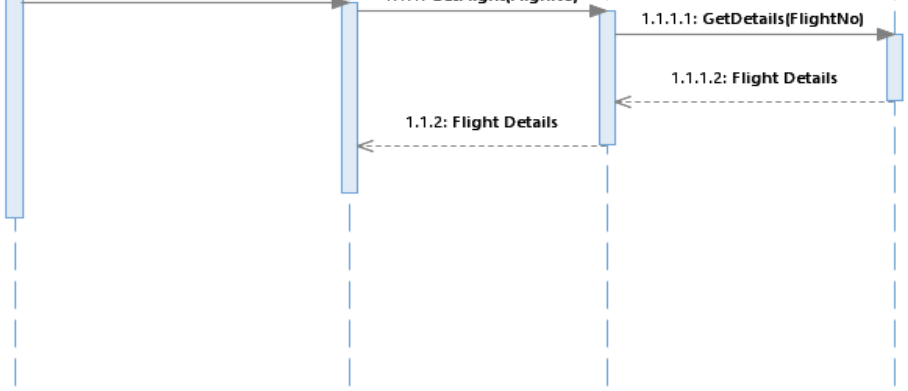
Manage Customer



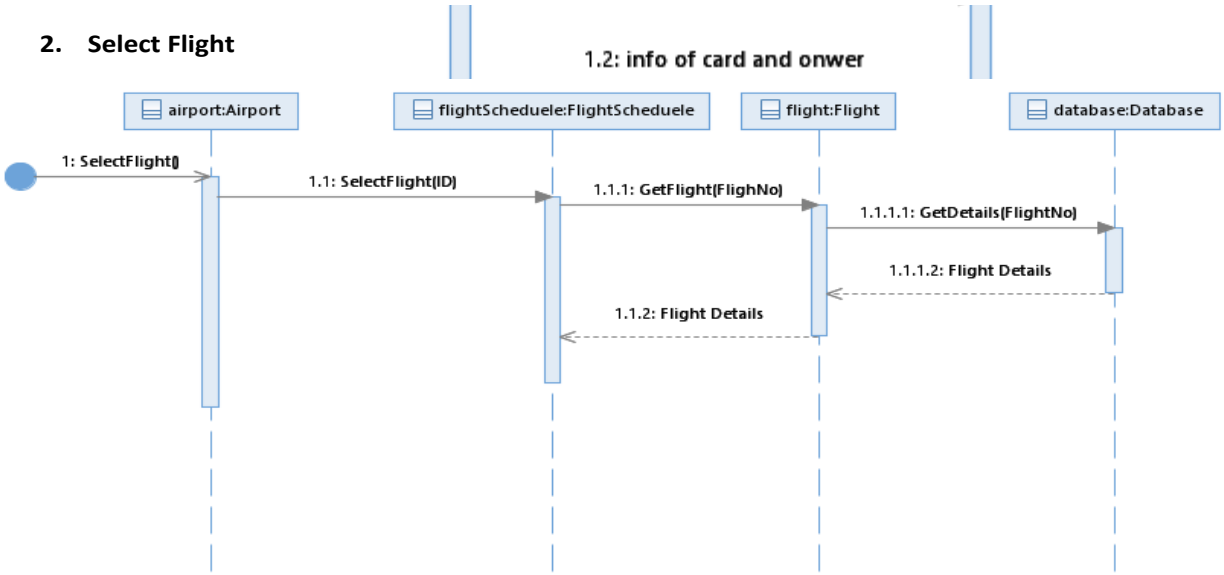
Manage Schedule



Verify Card Details



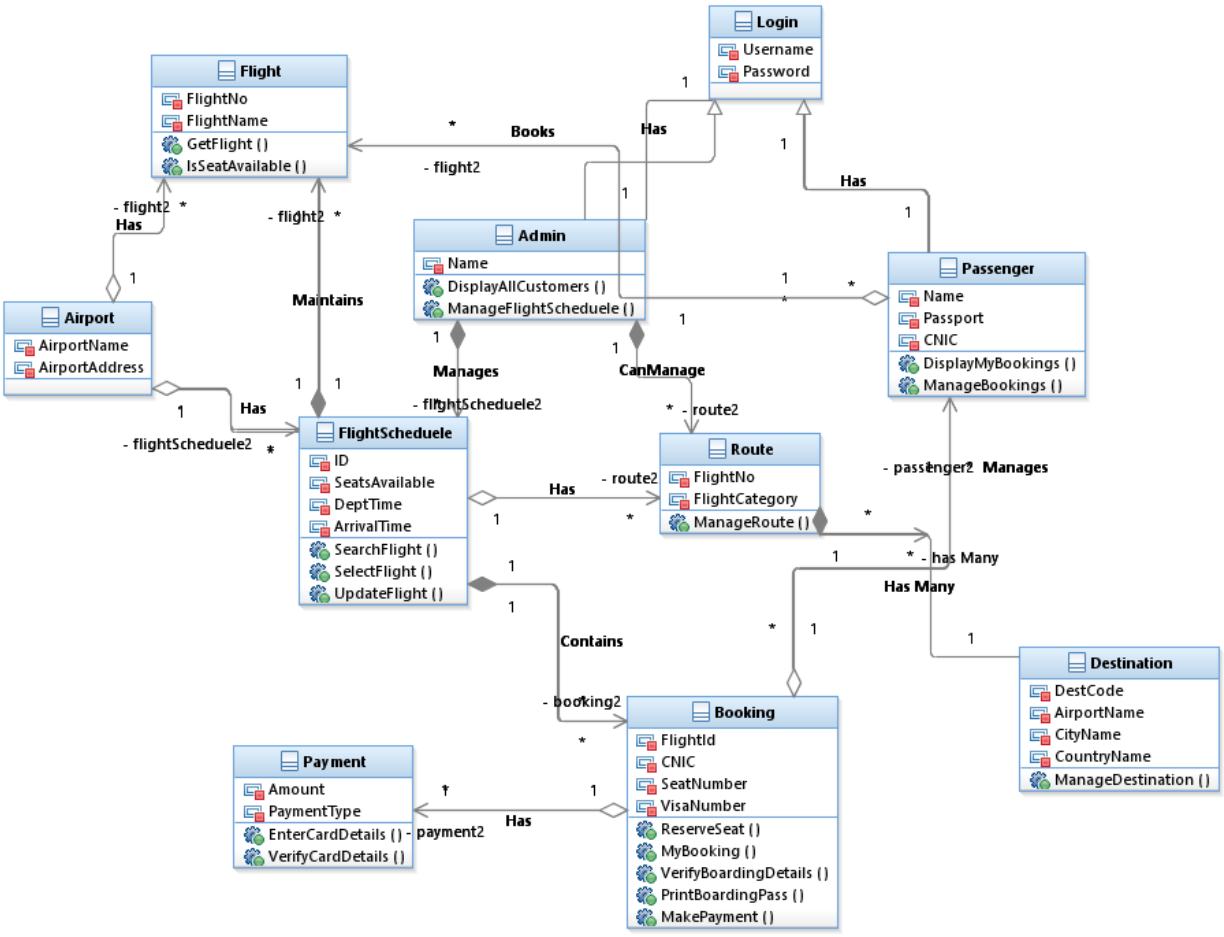
2. **Select Flight**



3. **Reserve Seat**

7. Class Diagram

1.


Login
Username
Password
Flight
FlightNo
FlightName
GetFlight ()
IsSeatAvailable ()
Books
Has
- flight2
Has
Admin
Name
DisplayAllCustomers ()
ManageFlightScheduele ()
Passenger
Name
Passport
CNIC
DisplayMyBookings ()
ManageBookings ()
Airport
AirportName
AirportAddress
Maintains
Manages
CanManage
- flightScheduele2
- route2
FlightScheduele
ID
SeatsAvailable
DeptTime
ArrivalTime
SearchFlight ()
SelectFlight ()
UpdateFlight ()
Route
FlightNo
FlightCategory
ManageRoute ()
- passenger2
Manages
- has Many
Has Many
Contains
- booking2
Destination
DestCode
AirportName
CityName
CountryName
ManageDestination ()
Booking
FlightId
CNIC
SeatNumber
VisaNumber
ReserveSeat ()
MyBooking ()
VerifyBoardingDetails ()
PrintBoardingPass ()
MakePayment ()
Payment
Amount
PaymentType
EnterCardDetails ()
VerifyCardDetails ()
- payment2