

# **AIR TICKET RESERVATION SYSTEM**

## **Abstract:**

- Air ticket reservation System is going to be a computerised system which will be used to store and retrieve information and conduct transactions related to air travel.
- It will allow the airline passenger to search for flights that are available between the two travel cities, namely the “Departure city” and “Arrival city” for a particular departure and arrival dates. The system will display all the flight’s details such as flight no, name, price and duration of journey etc.

## **Problem Statement:**

- The airport counter staff are very much prone to manual errors during the airline reservation process.
- Passengers have to wait in long queues just for getting their boarding pass and check-in luggage. In times of Covid-19 and social distancing standing in close proximity will come under high risk factor.
- The manual system is very time consuming for the passengers and that’s the only reason they have to come two hours prior to the airport.
- With the manual system the operating cost of the system is very high. The cost comprises the salary of 20-25 people, stationary, building rent, electricity, water, telephone etc.

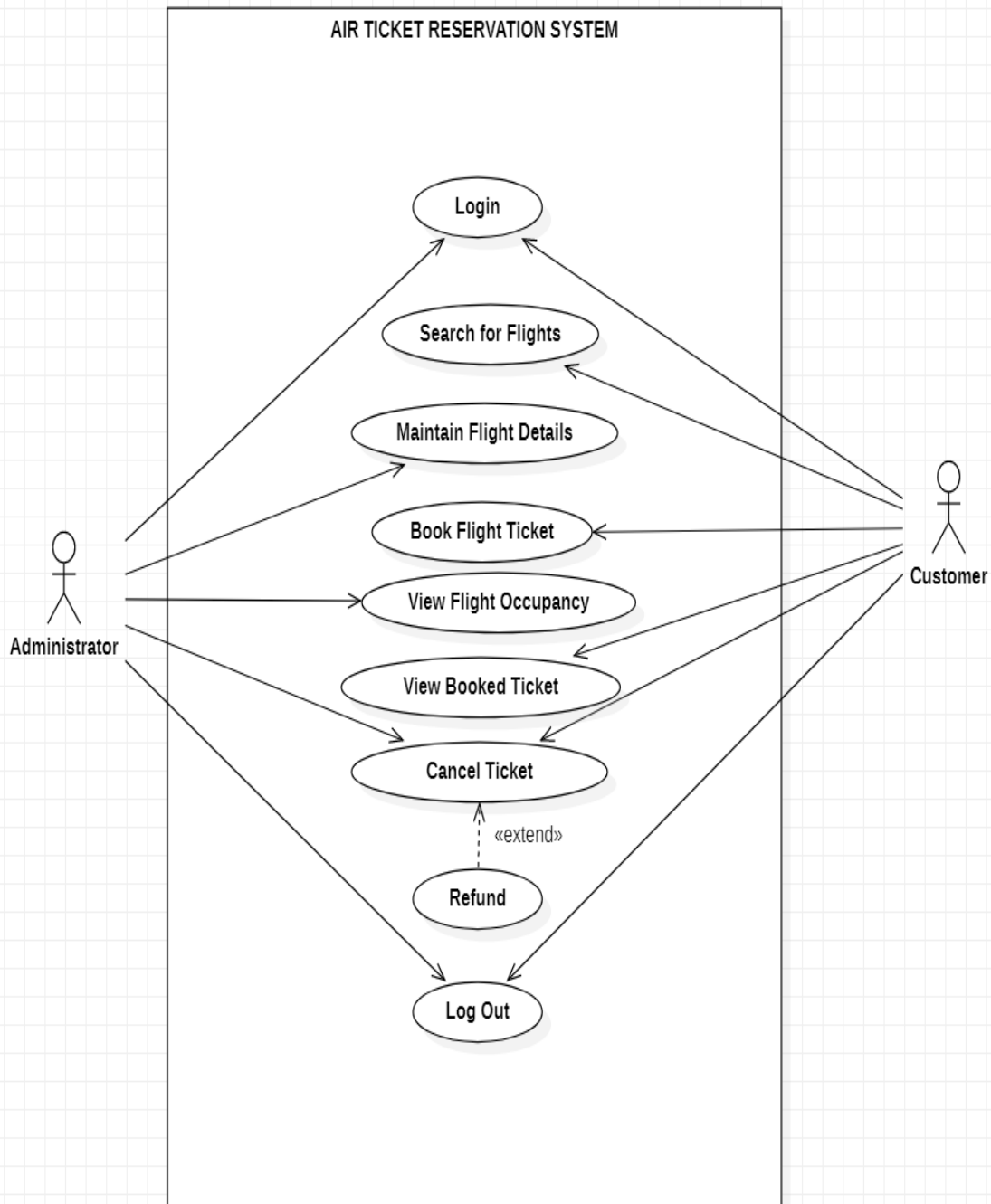
## **Proposed System:**

- As the bookings will be online there will not be any human errors.
- As the bookings are going to be online the social distancing constraint will be followed thereby reducing risk of contamination.
- Passengers can book their flights online thereby greatly reducing their waiting time to board the flight.
- The online system is going to reduce the operating cost by reducing the staff by around 40% percent thereby making it economically feasible.

## INITIAL REQUIREMENTS DOCUMENT

<b>Title of The Project</b>	Air Ticket Reservation System
<b>Stakeholders Involved in Capturing Requirements</b>	Customers, Administrator, Project leader, Support Staff
<b>Techniques Used in requirement Capturing</b>	Interviews and brainstorming
<b>Name of the person along with designation</b>	Dr. Rahul Katarya Suman Nandi
<b>Date</b>	25 <sup>th</sup> November 2020
<b>Version</b>	1.0
<b>Consolidated List of initial requirements:</b> <ol style="list-style-type: none"><li>1. A system is to be implemented which can be accessed by the Customers at any point of time.</li><li>2. The system shall be able to generate login ID and password to the system operator.</li><li>3. There are three types of members who will interact: Customers, Administrator, Support Staff</li><li>4. The Administrator should be able to add new flights to the system.</li><li>5. The Administrator should be able to update the details of the flight.</li><li>6. The system should require a user to register, in order to carry out any operations with it aside from checking the availability of tickets.</li><li>7. This system should keep a record of all failure login attempts with user login, terminal login and time.</li><li>8. The system should allow users to view flight details and book tickets by paying the required amount for the reserved tickets.</li><li>9. The system should present customers with the option to reschedule their travel trip and permit them to update their details anytime.</li><li>10. The users should be able to cancel their reserved tickets under a given time frame.</li><li>11. The user should be able to contact the airline company in case of any discrepancies.</li><li>12. The customer should have the option to search for the flights on specific dates and allow them to sort the flights according to their preferences.</li><li>13. The customer should be able to access the system independent of the web browser used.</li></ol>	

## USE CASE DIAGRAM



# **Software Requirements Specification**

**Version 1.0**

**November 28<sup>th</sup> 2020**

**Air Ticket Reservation System**

**Suman Nandi (2K20/SWE/23)**  
**MTech. Software Engineering**

**1st Semester**

## **Table of Contents**

<b>Table of Contents</b>	<b>5</b>
<b>Revision History</b>	<b>6</b>
<b>1. Introduction</b>	<b>7</b>
1.1. Purpose	7
1.2. Scope	7
1.3. Definition, Acronym, Abbreviation	7
1.4. References	7
1.5. Overview	8
<b>2. Overall Description</b>	<b>8</b>
2.1. Product Perspective	8
2.2. Product Functions	8
2.3. Operating Environment	9
2.4. Design And implementation Constraints	9
2.5. User Characteristics	9
2.6. Assumptions and Dependencies	9
<b>3. Specific Requirements</b>	<b>10</b>
3.1. External Interface Requirements	10
3.1.1. User Interfaces	12
3.1.2. Hardware Interfaces	12
3.1.3. Software Interfaces	12
3.1.4. Communication Interfaces	12
3.2. Functional Requirements	13
<b>4. System Features</b>	<b>23</b>
4.1. Login	23
4.2. Book Tickets	23
4.3. Search for Flights	23
4.4. View Ticket	23
4.5. Cancel Ticket	24
4.6. Logout	24
<b>5. Other Non-Functional Requirements</b>	<b>24</b>
5.1. Performance Requirements	24
5.2. Safety Requirements	24
5.3. Security Requirements	24
5.4. Software Quality Attributes	25
<b>Appendix A: Glossary</b>	<b>25</b>
<b>Appendix B: Analysis Models</b>	<b>26</b>
<b>Use Case Diagram</b>	<b>26</b>

## Revision History

<b>Name</b>	<b>Date</b>	<b>Version</b>
<b>Air Ticket Reservation System</b>	<b>November 30th 2020</b>	<b>1.0</b>

# 1. INTRODUCTION

Ticket reservation system for airlines has to be developed. The system developed should be able to allow the customers to login and book tickets online. It should contain the following features:

- The customer will have the functionality to search for one-way, round trip and multiple destination flights by choosing specific dates and destinations.
- While reserving tickets the system obtains following information from the user Passenger Name, Sex, Age, Address. Credit Card Number, Bank Name. Flight number, Flight name, Date of Journey and number of tickets to be booked.
- Based on the availability of tickets, the ticket has to be issued. The ticket issued should contain the following information –ticket number, flight no, flight name, date of journey, number of passengers, sex, age and departure time.
- Cancellation of booked tickets should be available.

## 1.1. PURPOSE

The purpose of this document is to present a detailed description of the Air Ticket Reservation System. It will explain the purpose and features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of the software and also potential developers. The document is subject the change as the project progresses. The given version of the document is the initial one. Further changes of the project will be recorded to the document.

## 1.2. SCOPE

- The product is titled Airline Reservation System (ARS).
- The product will perform the following tasks –
  - The software that is being developed can be used to check the availability of the flight tickets for the specified flight, destination and date of journey.
  - If the tickets are available to the user needs and specification, then the software provide a facility to book the tickets.
  - If the passenger wants to cancel the tickets, he can use the cancellation module of the Airline Reservation System.

## 1.3. Definition, Acronym, Abbreviation

ATRS: Airline Ticket Reservation System

## 1.4. References

- Object-Oriented Software Engineering by Yogesh Singh & Ruchika Malhotra, PHI Learning Pvt. Ltd., 2012.
- Google.

## **1.5. Overview**

- The SRS contains an analysis of the requirements necessary to help easy design.
- The overall description provides interface requirements for the Airline Reservation system, product perspective, hardware interfaces software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.
- Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect Airline Reservation system in any fashion.

## **2. OVERALL DESCRIPTION**

### **2.1. PRODUCT PERSPECTIVE**

The ATRS is designed for everyone who wishes to easily book air tickets. This software provides options for viewing different flights available with different timings for a particular date and provides customers with the facility to book a ticket, modify or cancel a particular reservation but it does not provide the customers with details of the cost of the ticket and it does not allow the customer to modify a particular part of his reservation and he/she can modify all his details.

### **2.2. PRODUCT FUNCTIONS**

1. A system is to be implemented which can run on the web server.
2. The system shall be able to generate Login Id and password to the system operator.
3. There are 3 types of members in the reservation system: User, Airlines and Admin.
4. The administrator shall be able to maintain details of all the flights.
5. The administrator shall be able to maintain details of all the users.
6. The system shall be able to book tickets.
7. The system should give the user the option to cancel the confirmed ticket.
8. A guest user can only check the availability of tickets and cannot book or buy tickets.
9. The system shall be able to fetch all the flights from origin to destination.
10. The system shall allow user to access the details about the arrival and departure times by requesting the user to input the flight number and date.
11. The user should be able to view all the past bookings.
12. The system shall be able to update the profile of the users.
13. The system shall be able to update the schedules of flights.



## **2.3. OPERATING ENVIRONMENT**

The ATRS is designed to be accessed through most of the versions windows operating systems such as:

- Windows 2000
- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows 10

## **2.4. DESIGN AND IMPLEMENTATION CONSTRAINTS**

Air Ticket Reservation System is developed in python language (Django Framework). We are using DB SQL Lite for the database. We are using bootstrap for its rich user-friendly interface.

## **2.5. USER CHARACTERISTICS**

1. The intended users of this software need not have specific knowledge as to what is the internal operation of the system. Thus, the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user.
2. The Product is absolutely user friendly, so the intended users can be naïve users.
3. The product does not expect the user to possess any technical background. Any person who knows how to use the mouse and the keyboard can successfully use this product.

## **2.6. ASSUMPTIONS AND DEPENDENCIES**

The requirements stated in the SRS could be affected by the following factors:

- One major dependency that the project might face is the changes that need to be incorporated with the changes in the airline policies regarding different services. As the policies changes the system needs to be updated with the same immediately. A delay in doing the same will result in tremendous loss to the airline. So, this should be changed as and when required by the developer.
- At this stage no quantitative measures are imposed on the software in terms of speed and memory although it is implied that all functions will be optimized with respect to speed and memory.
- It is furthermore assumed that the scope of the package will increase considerably in the future.

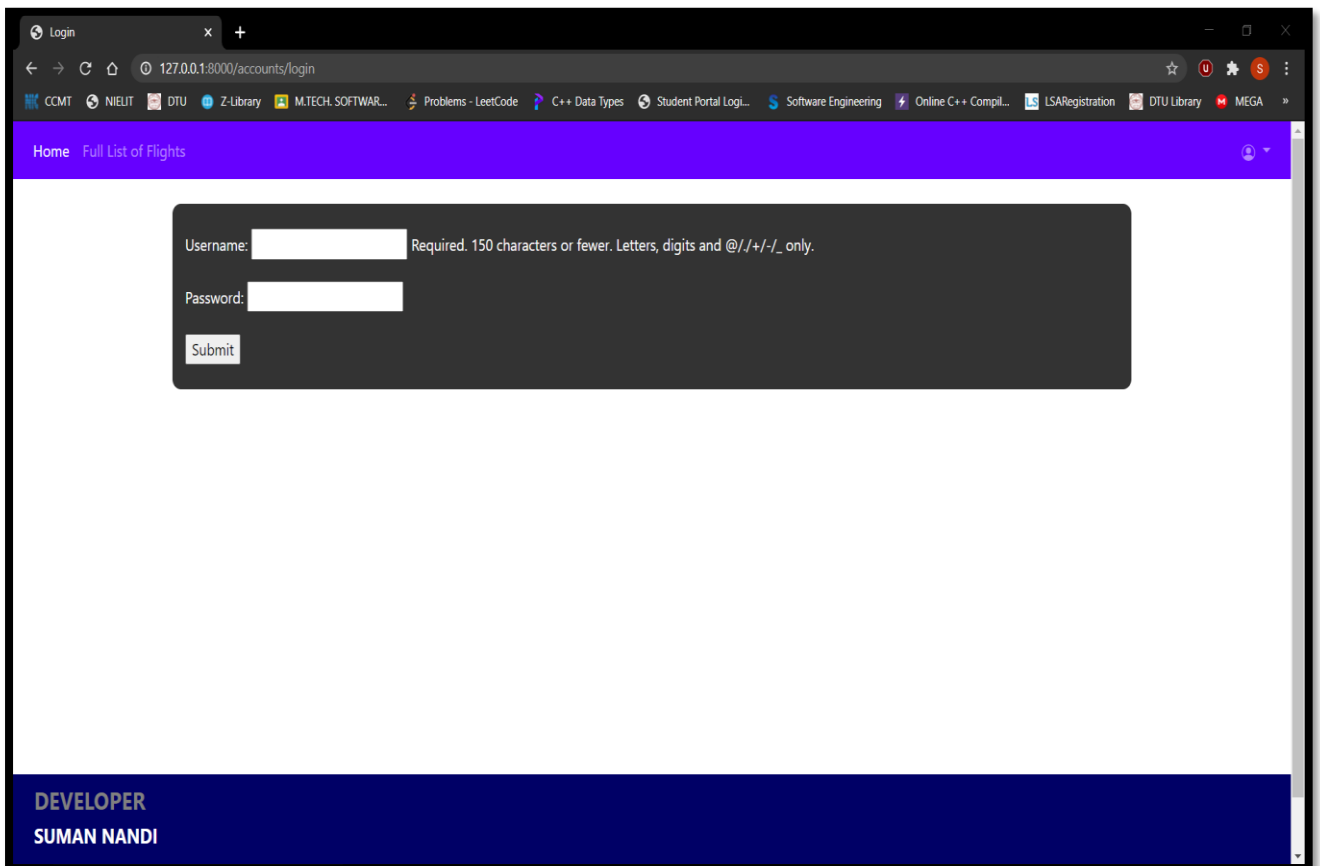
### 3. SPECIFIC REQUIREMENTS

#### 3.1. EXTERNAL INTERFACE REQUIREMENTS

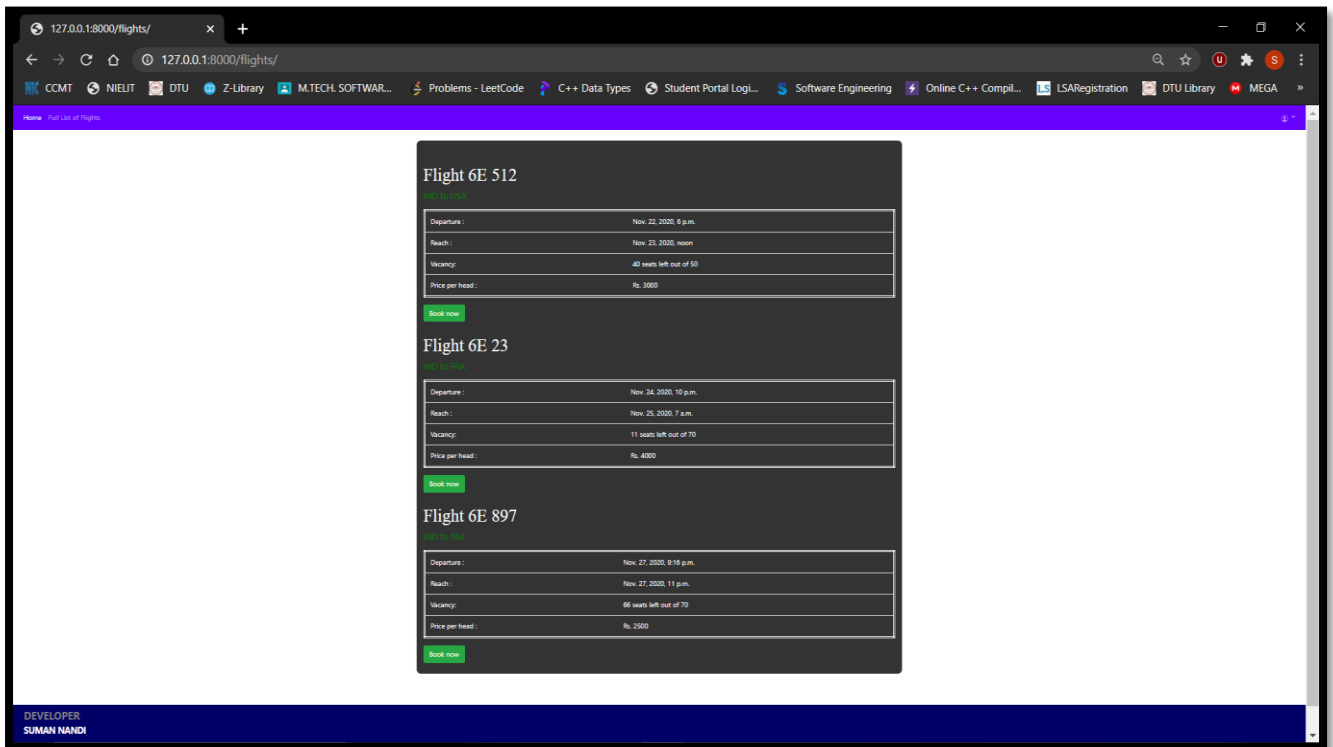
##### 3.1.1. USER INTERFACES

I have attached the actual screenshots below of my project Air Ticket Reservation System.

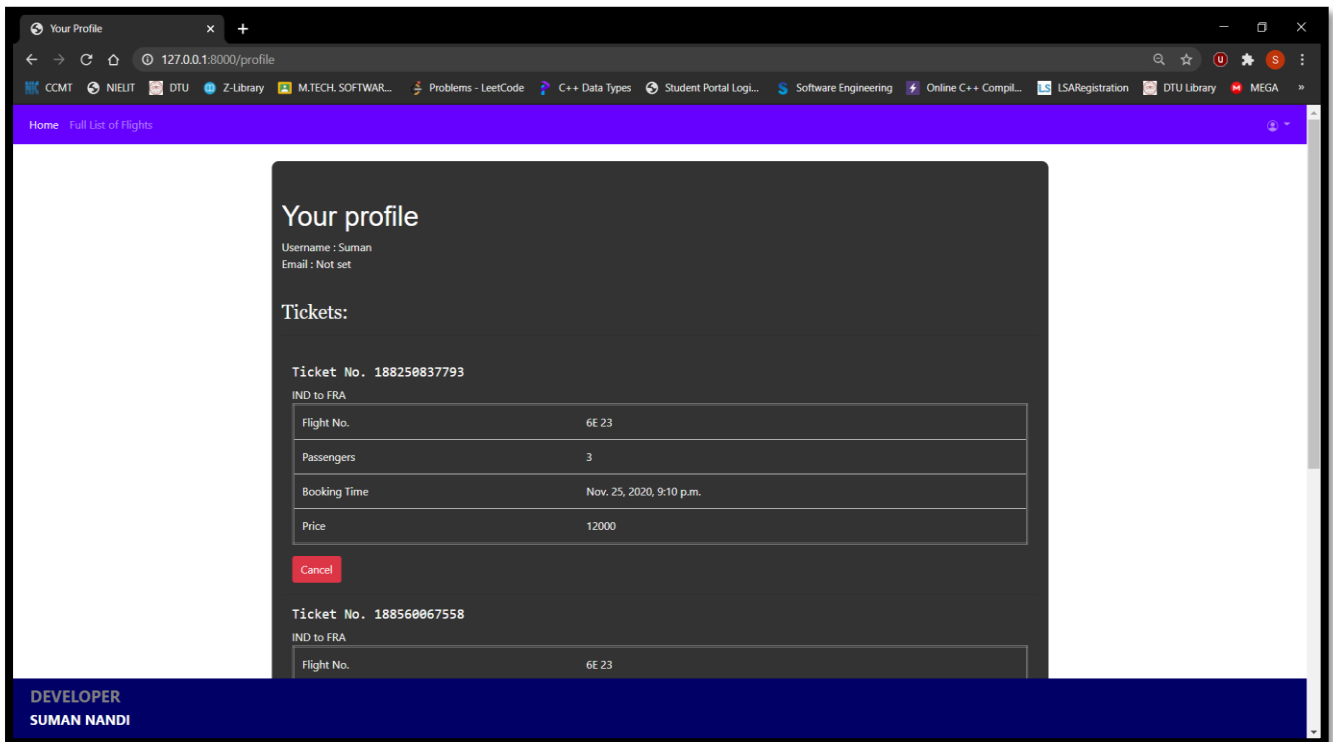
##### 3.1.1.1. LOGIN INTERFACE



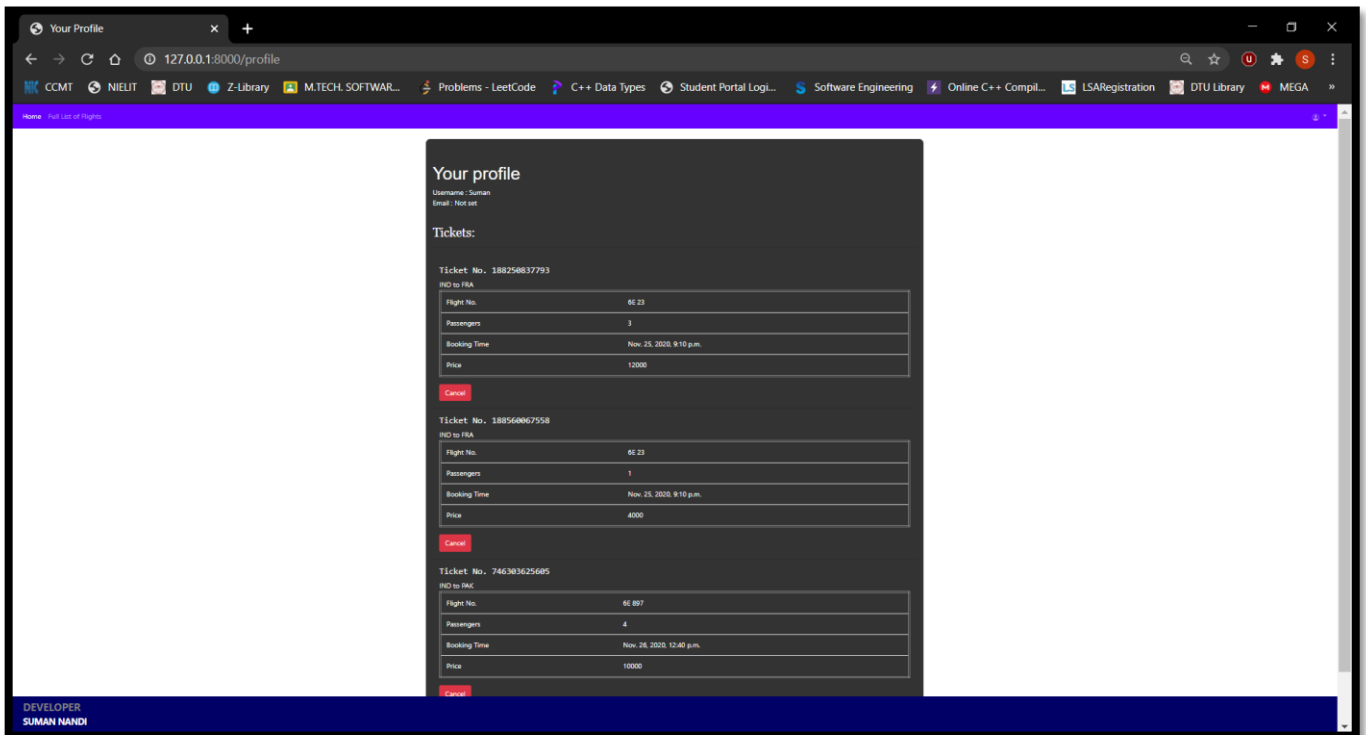
### 3.1.1.2. BOOK TICKET INTERFACE



### 3.1.1.3. VIEW TICKET INTERFACE



### 3.1.1.4. BOOKED TICKET LIST



### 3.1.2. HARDWARE INTERFACES

The minimum hardware requirements of Air Ticket Reservation System are a 1 Gigahertz CPU and 2 Gigabytes of RAM. The system must interface with the standard output device, keyboard and mouse to interact with this software.

### 3.1.3. SOFTWARE INTERFACES

ATRS is a desktop application and designed to be accessed by most of the versions of windows operating systems.

### 3.1.4. COMMUNICATION INTERFACES

ATRS might require an internet connection to install new updates.

## 3.2. FUNCTIONAL REQUIREMENTS

### 3.2.1. LOGIN USE CASE DESCRIPTION

<b>Introduction:</b> This use case describes all the steps followed in order to login into Air Ticket Reservation System.
<b>Actors:</b> Administrator, Customer
<b>Precondition:</b> The administrator/Customer must have a valid username and password.
<b>Postcondition:</b> If the username and password is valid, the actor is successfully logged in to the system; else an error message will be displayed.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. The system will request the actor to enter their login details for the system. The actor fills in the following details:<ul style="list-style-type: none"><li>- Username</li><li>- Password</li></ul></li><li>2. If login details are valid after clicking the Login button, the actor is successfully logged into the system.</li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li>1. <b>Invalid Entry</b> If Username / password entered by the actor is invalid or of any field username/password are empty then the system will display an error message and the actor returns to the basic flow.</li><li>2. <b>User Exits</b> User can exit at any time and use case ends.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> None

### 3.2.2. BOOK FLIGHT TICKET USE CASE

<b>Introduction:</b> This use case describes the steps taken in order to book a ticket.
<b>Actors:</b> Administrator, Customer
<b>Precondition:</b> The Administrator/Customer must be logged onto the system before use case begins.
<b>Postcondition:</b> If the use case is successful, a ticket is booked and the database is updated else the system remains unchanged.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. This use case starts when the Administrator/Customer/Airline wishes to book a ticket.</li><li>2. The system requests Administrator/Passenger/Airline to enter basic details that includes:<ul style="list-style-type: none"><li>- Name</li><li>- Number of Passengers</li><li>- Source</li><li>- Destination</li><li>- Type of Flight</li></ul></li><li>3. After Successful insertion of all the details the user will be asked to pay for the ticket and the ticket is successfully booked.</li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li><b>1. Invalid Details:</b> If in Book Details any invalid detail is mentioned or left blank, the system will display an error message and return to basic flow.</li><li><b>2. User Exit:</b> User exit Unexpectedly, the system starts from the beginning.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> LOGIN

### 3.2.3. CANCEL TICKET USE CASE

<b>Introduction:</b> This use case describes the steps taken in order to cancel a ticket.
<b>Actors:</b> Administrator, Customer
<b>Precondition:</b> The Administrator/Customer must be logged onto the system before use case begins.
<b>Postcondition:</b> If the use case is successful, a ticket is cancelled and the database is updated, the system remains unchanged.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. This use case starts when the Administrator/Customer wishes to cancel a ticket.</li><li>2. The use case requires the Administrator/Customer to enter the BOOKING ID into the system that he/she wants to cancel.</li><li>3. The user selects cancel ticket.</li><li>4. The appropriate databases are updated and the refund is given to user and the ticket is cancelled.</li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li><b>1. Invalid Booking ID:</b> If the Booking ID entered is not present, the system will display the message of BOOKING ID NOT FOUND and will ask the user to enter the BOOKING ID again.</li><li><b>2. Time to Cancel Ticket Exceeds</b> If the user cancels the ticket after the 24 hours period, no refund will be given to the user and ticket will be cancelled.</li><li><b>3. User Exit</b> User exit Unexpectedly, the system goes from beginning and no changes in the system takes place.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> LOGIN, REDUND

### 3.2.4. VIEW BOOKED TICKET USE CASE

<b>Introduction:</b> This use case describes the steps taken in order to view a ticket.
<b>Actors:</b> Administrator, Customer
<b>Precondition:</b> The Administrator/Customer must be logged onto the system before use case begins.
<b>Postcondition:</b> If the use case is successful, a ticket is shown to the user else the system remains unchanged.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. This use case starts when the Administrator/Customer wishes to view a ticket.</li><li>2. The user clicks on the VIEW TICKET tab to view the tickets booked.</li><li>3. The Booked ticket should display the following details:<ul style="list-style-type: none"><li>- Name</li><li>- Number of Passengers</li><li>- Source</li><li>- Destination</li><li>- Type of Flight</li></ul></li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li>1. <b>Invalid Ticket Number</b> If the Ticket Number entered is not correct, the system will display the message of INVALID TICKET NUMBER and will ask the user to enter the Ticket Number again.</li><li>2. <b>User Exit</b> User exit Unexpectedly, the system goes from beginning and no changes in the system takes place.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> LOGIN



### 3.2.5. REFUND USE CASE DESCRIPTION

<b>Introduction:</b> This use case describes all the steps followed in order to apply for the refund of cancelled ticket in Air Ticket Reservation System.
<b>Actors:</b> Customer
<b>Precondition:</b> The Customer must be logged into the system and applied for cancellation of ticket.
<b>Postcondition:</b> Refund if any must be processed in accordance with the refund policy of the system.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. The system will request the actor to cancel the ticket in order to apply for a refund.</li><li>2. If ticket is cancelled in given time period, a refund will be processed according to the refund policy of the system.</li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li>1. <b>Invalid Ticket Number</b> If the Ticket Number entered is not correct, the system will display the message of INVALID TICKET NUMBER and will ask the user to enter the Ticket Number again.</li><li>2. <b>User Exits</b> User can exit at any time and use case ends.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> CANCEL TICKET

### 3.2.6. LOGOUT USE CASE DESCRIPTION

<b>Introduction:</b> This use case describes all the steps followed in order to logout the Air Ticket Reservation System.
<b>Actors:</b> Administrator, Customer
<b>Precondition:</b> Administrator/Customer must be logged into the system.
<b>Postcondition:</b> The actor is successfully logged out of the system.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. The system will request the actor to click the logout button of the system.</li><li>2. The actor will get logged out of the system and return to the login page.</li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li>1. <b>User Exits</b> User can exit at any time and use case ends.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> LOGIN

### 3.2.7. SEARCH FOR FLIGHT USE CASE

<b>Introduction:</b> This use case describes all the steps followed in order to search for flight in Air Ticket Reservation System.
<b>Actors:</b> Customer
<b>Precondition:</b> Customer must be logged into the system.
<b>Postcondition:</b> Flight schedules are displayed with all the details.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. This use case starts when the Customer wishes to search for flights.</li><li>2. The Administrator/Customer needs to enter the following details:<ul style="list-style-type: none"><li>- Source City</li><li>- Destination City</li><li>- Date of Flight</li></ul></li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li>1. <b>Invalid Date of Flight</b> If the Date of flight is not in correct format or the date entered is before the current date, the system will display the message RE-ENTER DATE.</li><li>2. <b>Source/ Destination City is blank</b> If the Source/Destination city is left blank, the system will show the message SELECT SOURCE/DESTINATION.</li><li>3. <b>User Exit</b> User exit Unexpectedly, the goes to beginning and no changes in the system takes place.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> LOGIN

### 3.2.8. MAINTAIN FLIGHT DETAILS USE CASE

<b>Introduction:</b> This use case describes all the steps taken in order to maintain flight details.
<b>Actors:</b> Administrator
<b>Precondition:</b> Administrator must be logged into the system.
<b>Postcondition:</b> The flight information will be added/ updated / deleted / viewed.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. The use case starts when the administrator wants to maintain (Add, update, delete or view) the flight details.</li><li>2. Administrators can request for add/delete/update/view operation.<ul style="list-style-type: none"><li>- If the administrator requests for add operation then add flight sub flow executed.</li><li>- If the administrator requests for delete operation then delete flight sub flow executed.</li><li>- If the administrator requests for update operation then update flight sub flow executed.</li><li>- If administrator requests for view operation then view flight sub flow executed.</li></ul></li></ol> <b>1. Add Flight</b> <p>The system requests the administrator to fill some details. This include</p> <ul style="list-style-type: none"><li>○ Airplane Model Number</li><li>○ Airplane Registration Number</li><li>○ Airplane Capacity</li><li>○ Flight Number</li><li>○ Departing city</li><li>○ Arrival city</li><li>○ Flight Departure Date</li><li>○ Flight Departure Time</li><li>○ Flight Arrival Date</li><li>○ Flight Arrival Time</li></ul> <p>Flight gets added to the system.</p> <b>2. Delete Flight</b> <ul style="list-style-type: none"><li>• The system requests the administrator to enter the Flight Number of the flight.</li><li>• The system retrieves and displays the flight information.</li><li>• The administrator confirms the delete request.</li><li>• The system deletes the flight record.</li></ul>

### **3. Update Flight**

- The system requests the administrator to enter the flight number of the flight.
- The system retrieves the details, displays the details and goes to the add flight sub flow.
- After adding necessary details, the flight details are updated.

### **4. View Flight**

- The system requests the administrator to enter the flight number of the flight.
- The system retrieves and displays the flight information.

#### **Alternate Flows:**

##### **1. Invalid Entry**

If in the add flight or update flight sub flow, the actor enters invalid entry or leaves any field empty the system displays an error message. The actor returns to basic flow and may re-enter the invalid entry.

##### **2. Flight Already Exists**

If in the add flight sub flow, a flight with a specified flight number exists then the system generates an error message and the actor returns to basic flow.

##### **3. Flight Not Found**

If in the update flight, delete flight or view flight sub flow the flight information with the specified flight number does not exist, the system displays an error message and flow goes to basic flow.

##### **4. Update Cancelled**

If in the update flight sub flow, the actor decides not to update the details then the update is cancelled and flow goes to the beginning of basic flow.

##### **5. Delete Cancelled**

If in the delete flight sub flow, the actor decides not to delete the details then delete is cancelled and flow goes to the beginning of basic flow.

##### **6. User Exits**

User exit Unexpectedly, then goes to beginning and no changes in the system takes place.

**Special Requirements:** None

**Associated Use Case:** LOGIN

### 3.2.9. VIEW FLIGHT OCCUPANCY USE CASE

<b>Introduction:</b> This use case describes all the steps followed in order to view the flight occupancy in Air Ticket Reservation System.
<b>Actors:</b> Administrator
<b>Precondition:</b> Administrator must be logged into the system.
<b>Postcondition:</b> Flight occupancy are shown with all the details.
<b>Flow of Events:</b>  <b>Basic Flow:</b> <ol style="list-style-type: none"><li>1. This use case starts when the Administrator/Airline wishes to view the flight occupancy.</li><li>2. The Administrator needs to enter the Flight Number of the flight.</li><li>3. The Details of the flight is displayed.</li></ol> <b>Alternate Flows:</b> <ol style="list-style-type: none"><li><b>1. Invalid Flight Number</b> If the flight number is not correct, the system will display the message INVALID FLIGHT NUMBER.</li><li><b>2. User Exit</b> User exit Unexpectedly, the goes to beginning and no changes in the system takes place.</li></ol>
<b>Special Requirements:</b> None
<b>Associated Use Case:</b> LOGIN

## **4. SYSTEM FEATURES**

This section demonstrates ATRS's most prominent features and explains how they can be used and the results they will give back to the user.

### **4.1. LOGIN**

- This function allows a registered user to login his account using his frequent flyer number with the airline and password. If a user is not registered, the website shall allow the user to enroll first. The system will check both the frequent flight number and password, when a user attempts to login.
- This provides security to the system by authenticating each member and provides confidence to the consumer that his/her personal information is secure.

### **4.2. BOOK TICKETS**

- The user can use the Book Ticket function to purchase seats for an airplane flight. The system shall present the user with information on all current flights. The user may then select a pair (departure and return) of flights on which to purchase seats. The user can indicate the number of seats and placement of such. Finally, the system shall guide the user completely through the checkout process.
- The heart of the business is selling seats on flights. This section provides the primary source of system transactions.

### **4.3. SEARCH FOR FLIGHT**

- The user will be able to view list of different flights on entering the source city, destination city and date of flight.
- The user will be able to select their favourite airline and book max of six tickets at once of their choice.

### **4.4. VIEW TICKET**

- The user can use the View Ticket function to view the booked tickets. Every information about the ticket which is booked by the user will be shown by this function such as time of booking, no of tickets, details of passenger, source and destination, class of ticket, payment details etc.

## **4.5. CANCEL TICKET**

- The user can use the Cancel Ticket function to cancel the already booked tickets for an airplane flight. The system shall present the user with information on which tickets to be cancelled if booked a greater number of tickets from a single account. The user may then select the corresponding ticket of flight, which needs to be cancelled. Finally, the system shall guide the user through the cancellation process and refund policy.

## **4.6. LOGOUT**

- The Logout section provides a way for the user to securely log out of the system. This process will save all user operations when he/she exits the system. If a user wishes to continue accessing the website, he/she must log-in again to access user features.
- Customers often use shared computers. Providing a way to clearly state and log-out gives our customers confidence that nobody else will use their flight-booking session.

## **5. OTHER NON-FUNCTIONAL REQUIREMENTS**

### **5.1. PERFORMANCE REQUIREMENTS**

- The minimum hardware requirements of Air Ticket Reservation System are a 1 Gigahertz CPU and 2 Gigabytes of RAM. The system must interface with the standard output device, keyboard and mouse to interact with this software.

### **5.2. SAFETY REQUIREMENTS**

- To ensure that no one of ATRS's users loses any data while using ATRS (due to a crash or a bug of some kind) the developer team updates ATRS regularly.

### **5.3. SECURITY REQUIREMENTS**

ATRS has two levels of users each with different privileges i.e.

- Administrator who is the super user and can perform most of the functions in the system such as creating users, view passengers list, view flight occupancy etc.
- Customer/User who can book tickets, cancel tickets, view ticket and search for flights.



## **5.4. SOFTWARE QUALITY ATTRIBUTES**

ATRS provides the users with very simple features. Due to its well designed and easy to use interface it can be used by any users. However, users must already have a basic knowledge of using a computer.

## **APPENDIX A: GLOSSARY**

SRS- Software Requirements Specification

CPU- Central Processing Unit

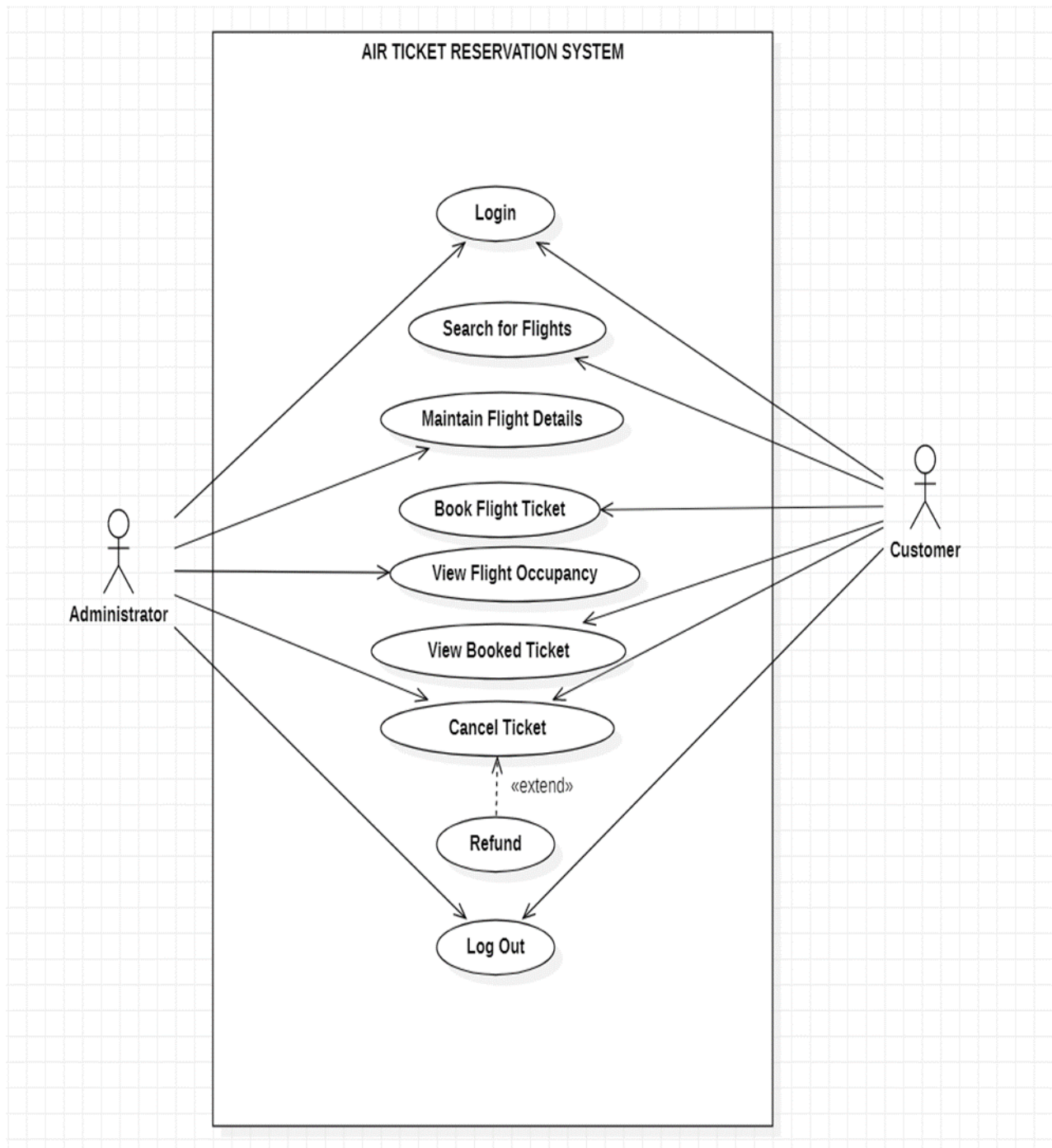
ATRS- Air Ticket Reservation System

IEEE- Institute of Electrical and Electronics Engineers

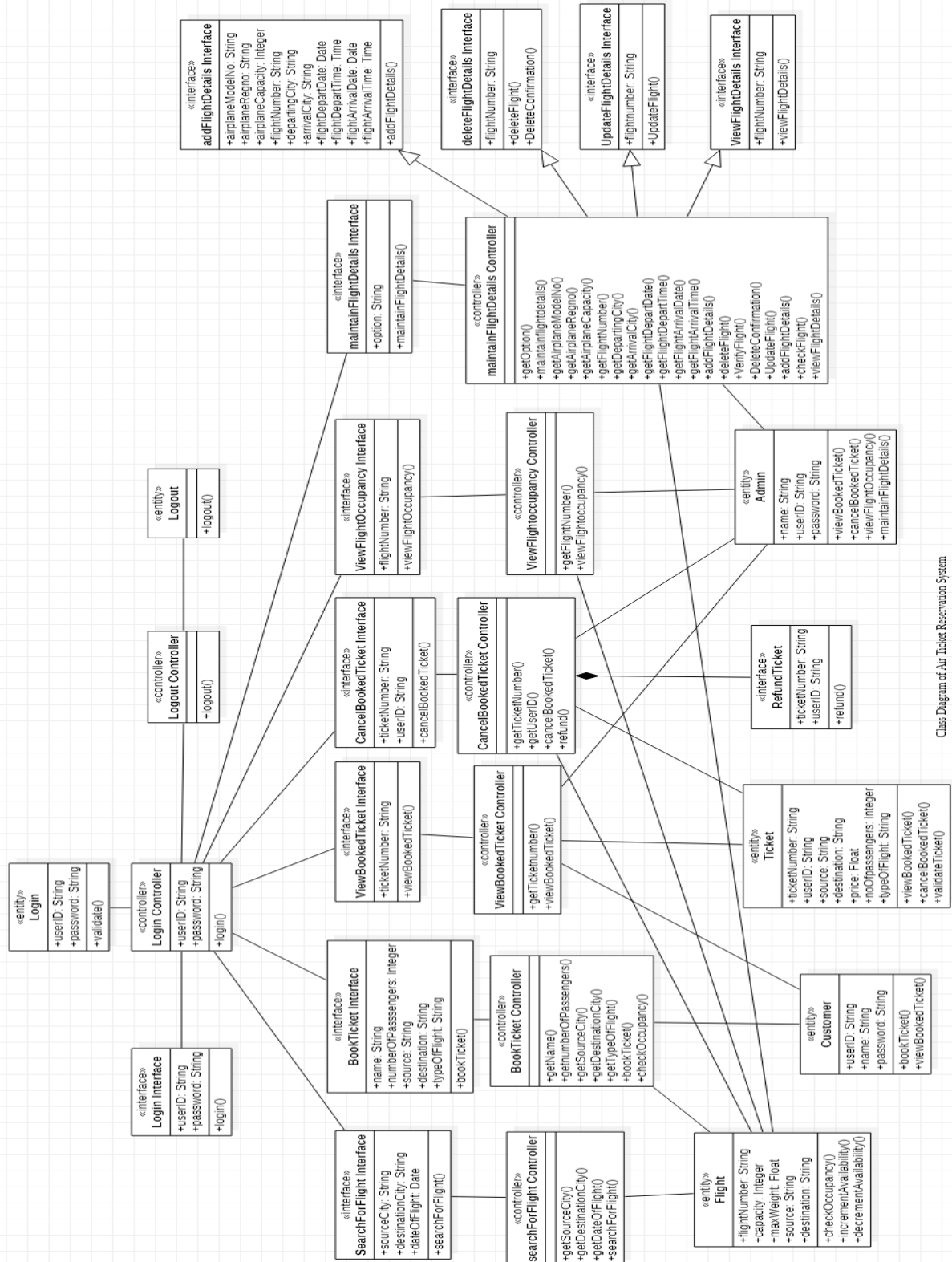
RAM- Random Access Memory

## APPENDIX B: ANALYSIS MODELS

### 1. USE CASE DIAGRAM



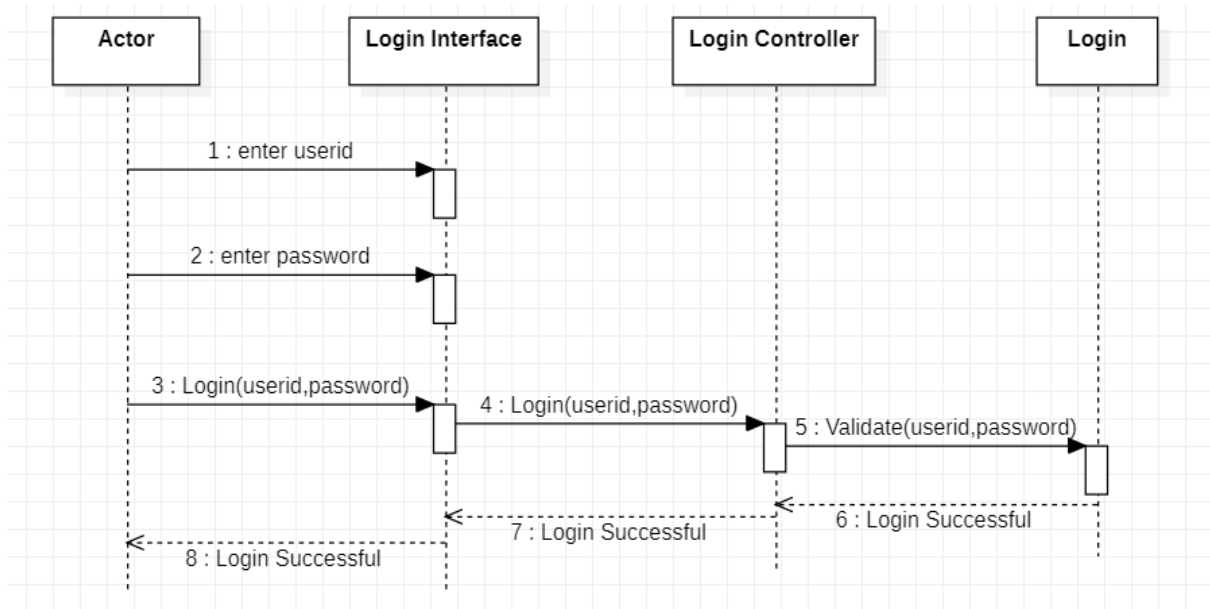
## 2. CLASS DIAGRAM



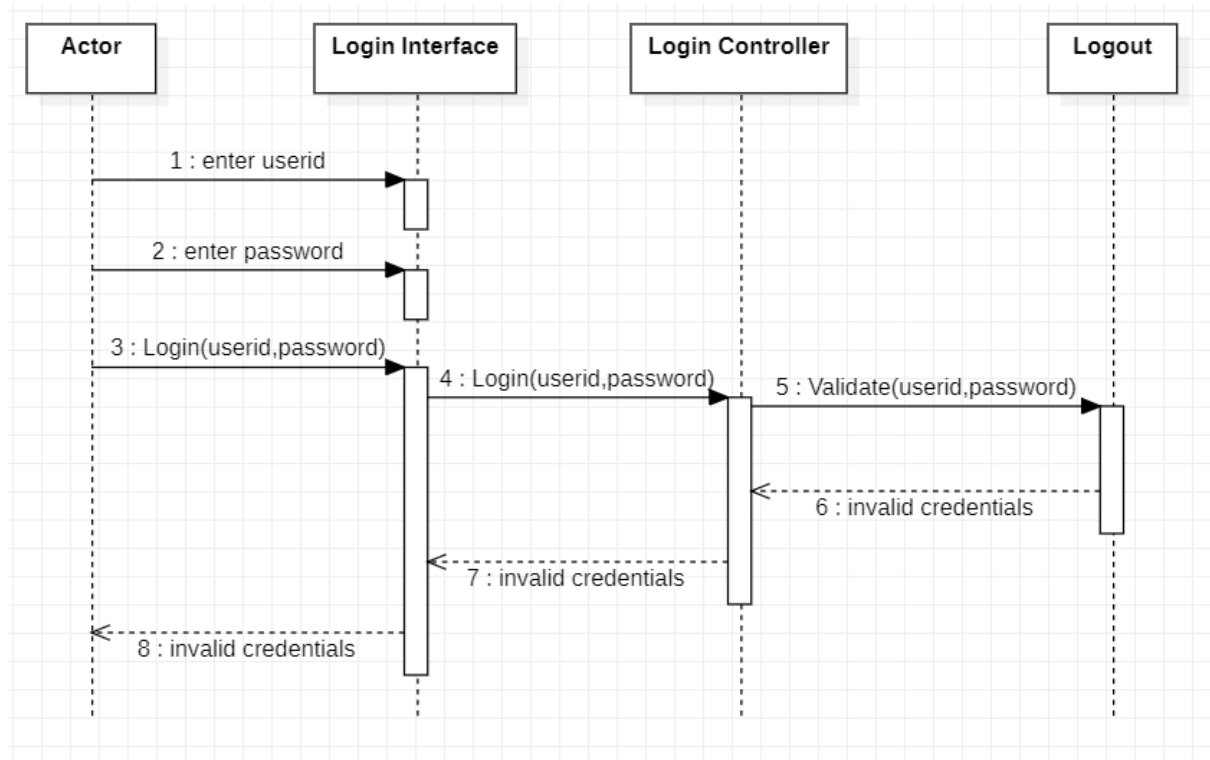
### 3. SEQUENCE DIAGRAM

#### 3.1. LOGIN

##### 3.1.1. BASIC FLOW

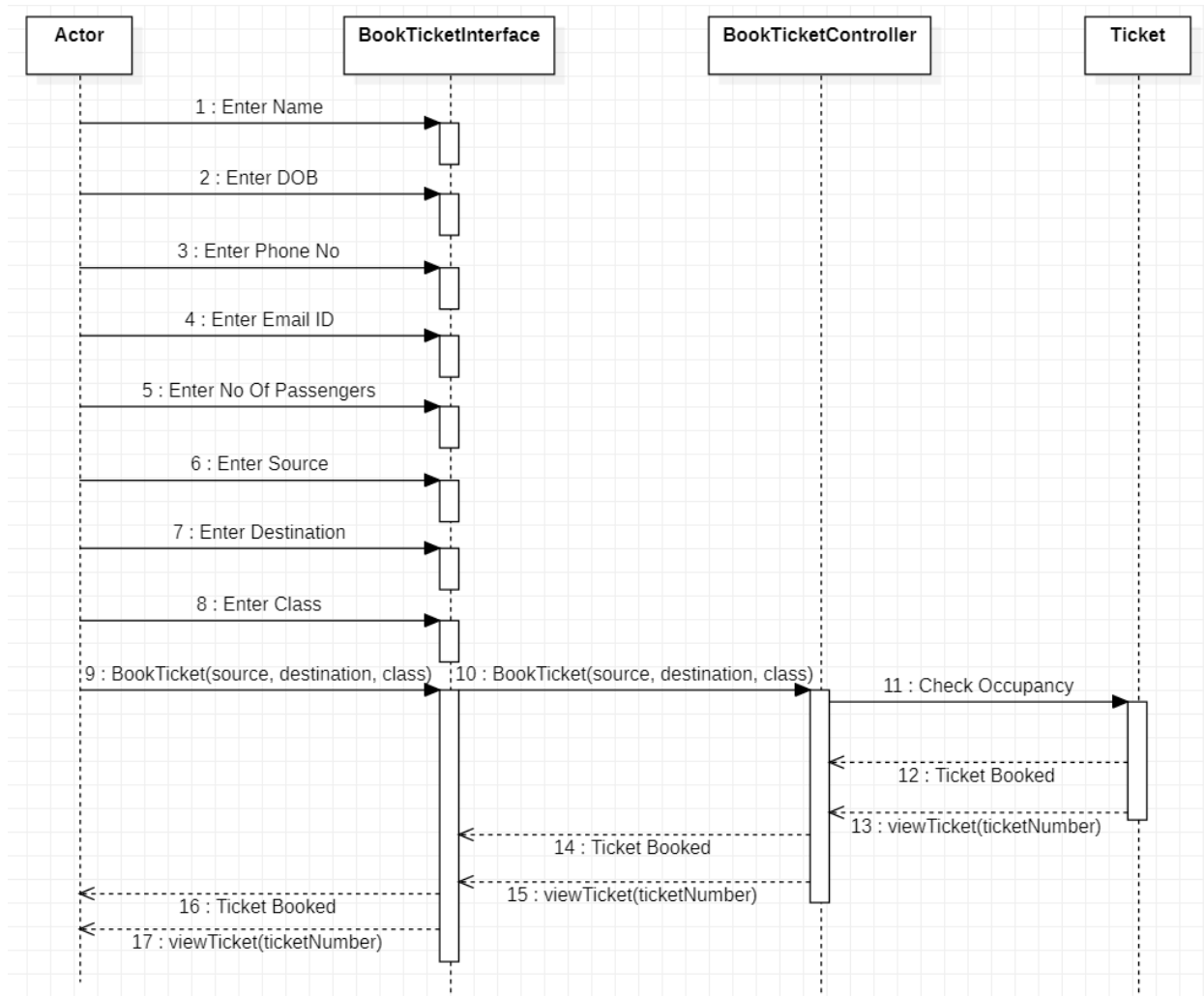


##### 3.1.2. ALTERNATE FLOW

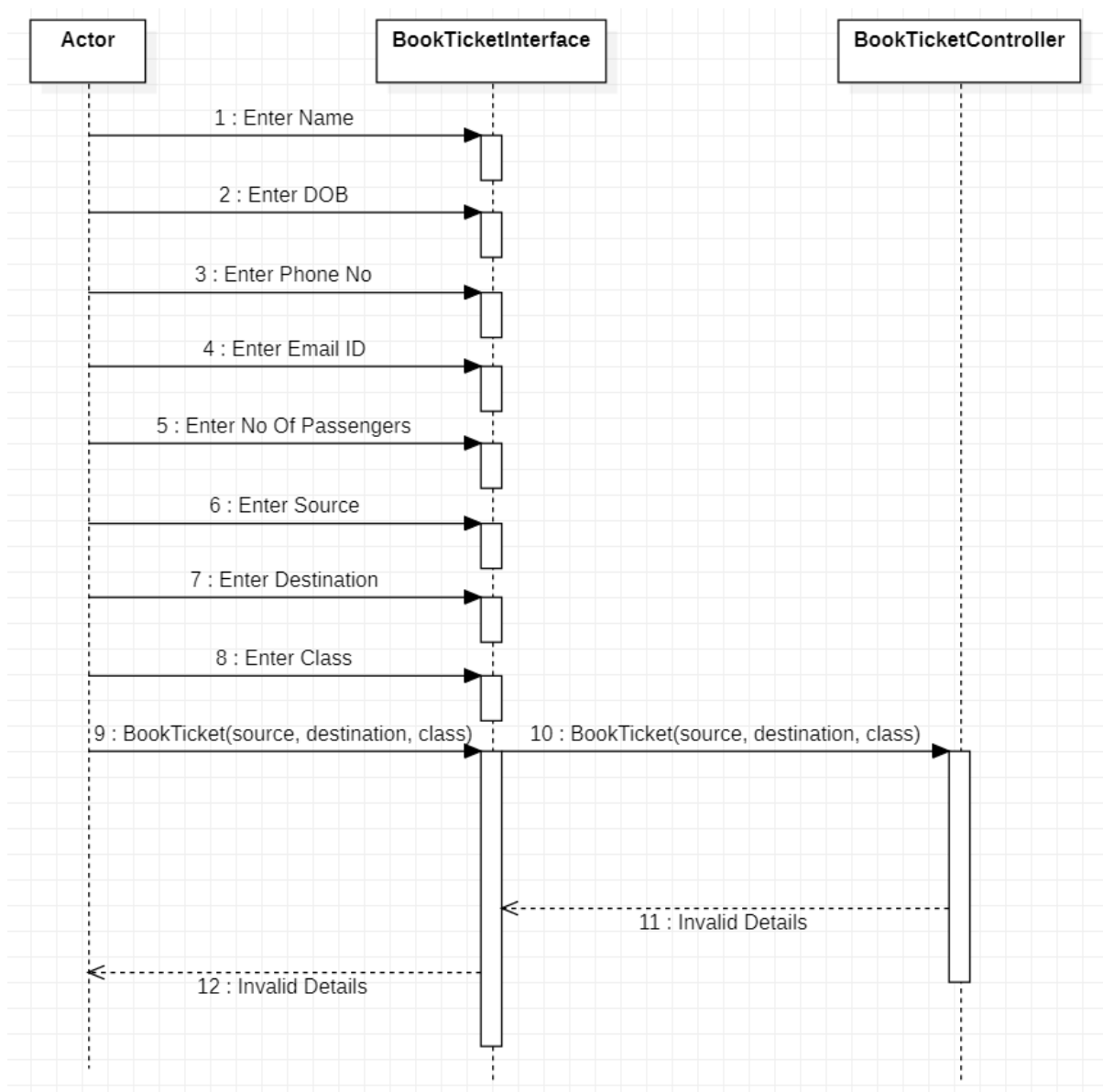


## 3.2. BOOK FLIGHT TICKET

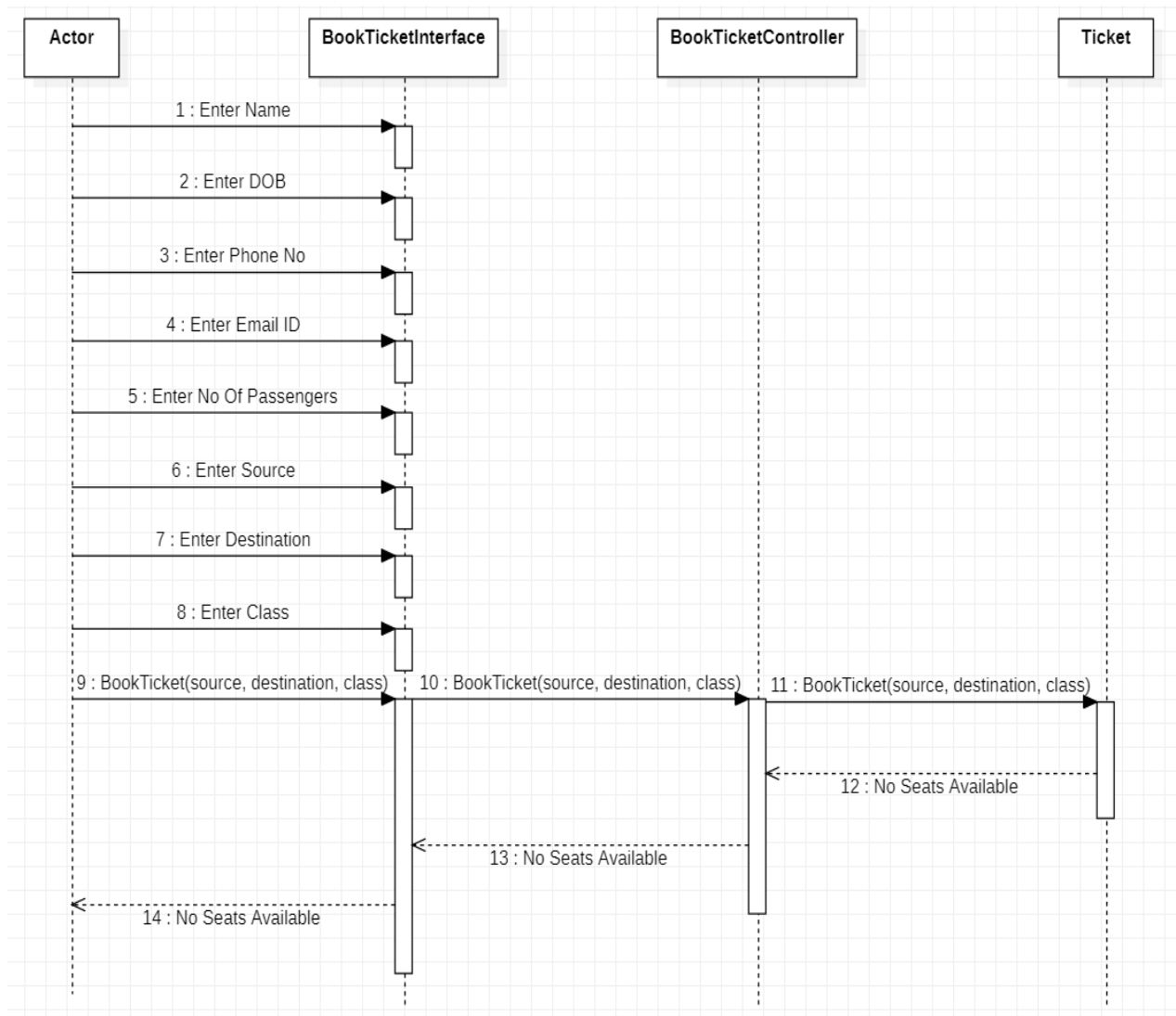
### 3.2.1. BASIC FLOW



### 3.2.2. ALTERNATE FLOW 1: INVALID DETAILS

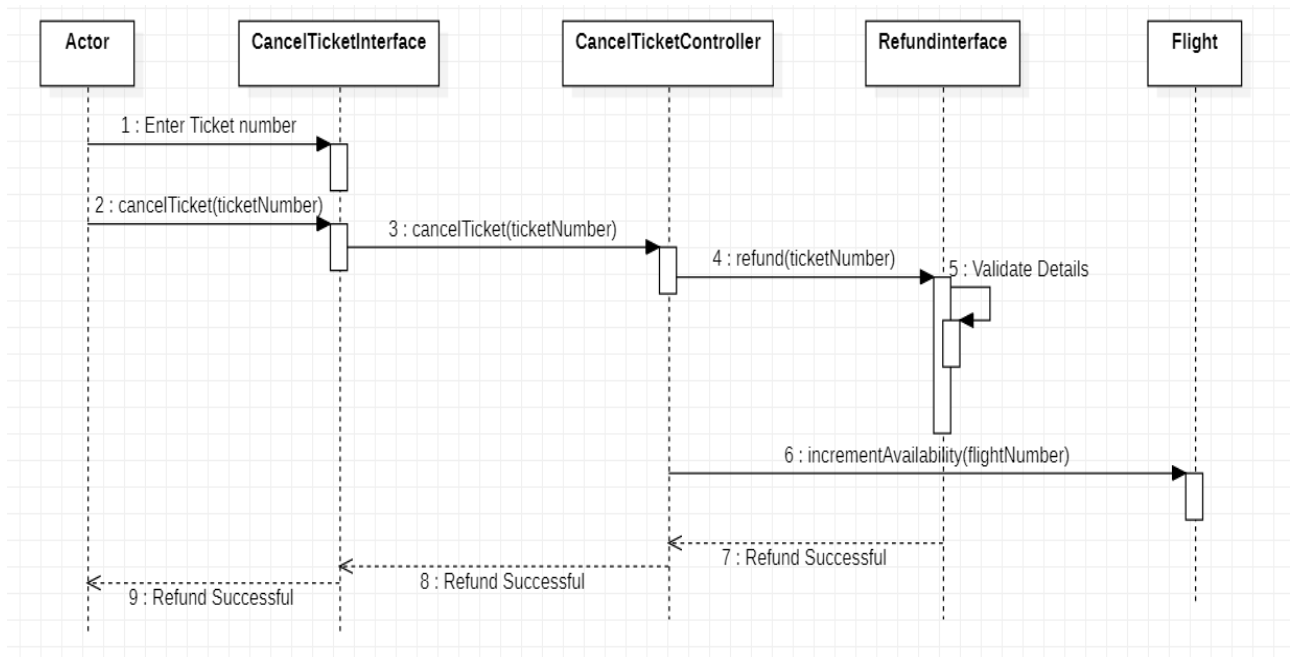


### 3.2.3. ALTERNATIVE FLOW 2: NO SEATS AVAILABLE

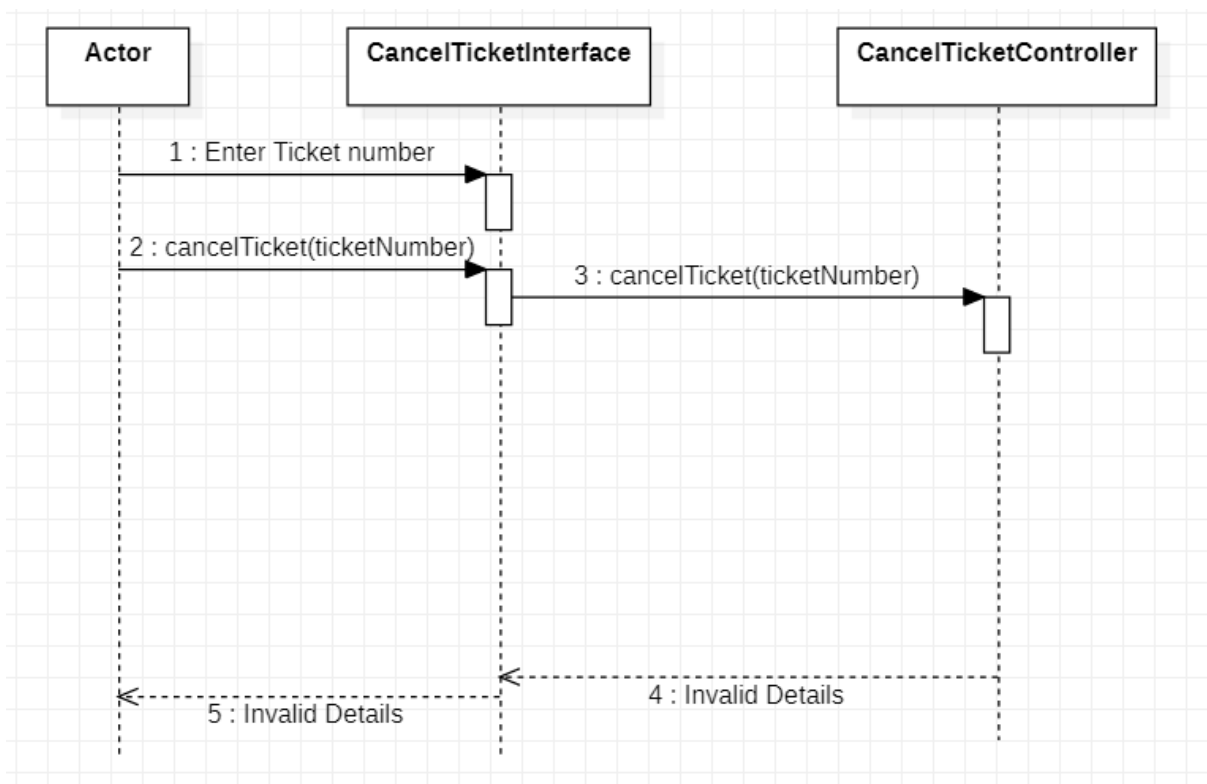


### 3.3. CANCEL TICKET

#### 3.3.1. BASIC FLOW

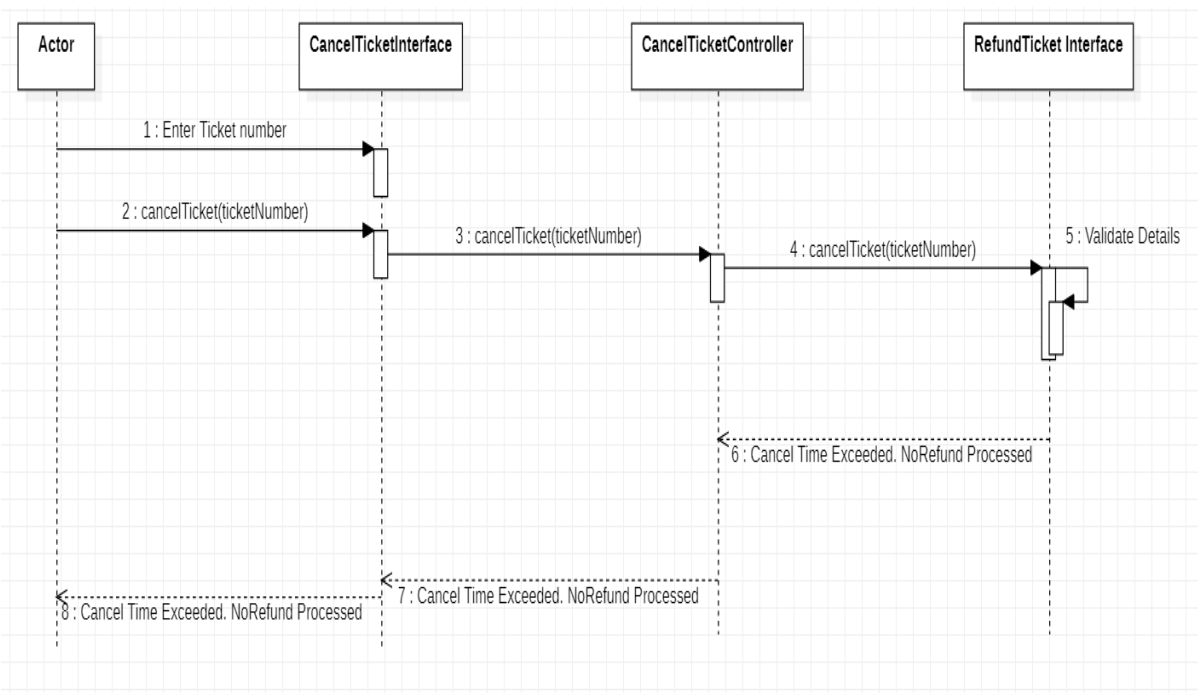


#### 3.3.2. ALTERNATE FLOW 1: INVALID DETAILS



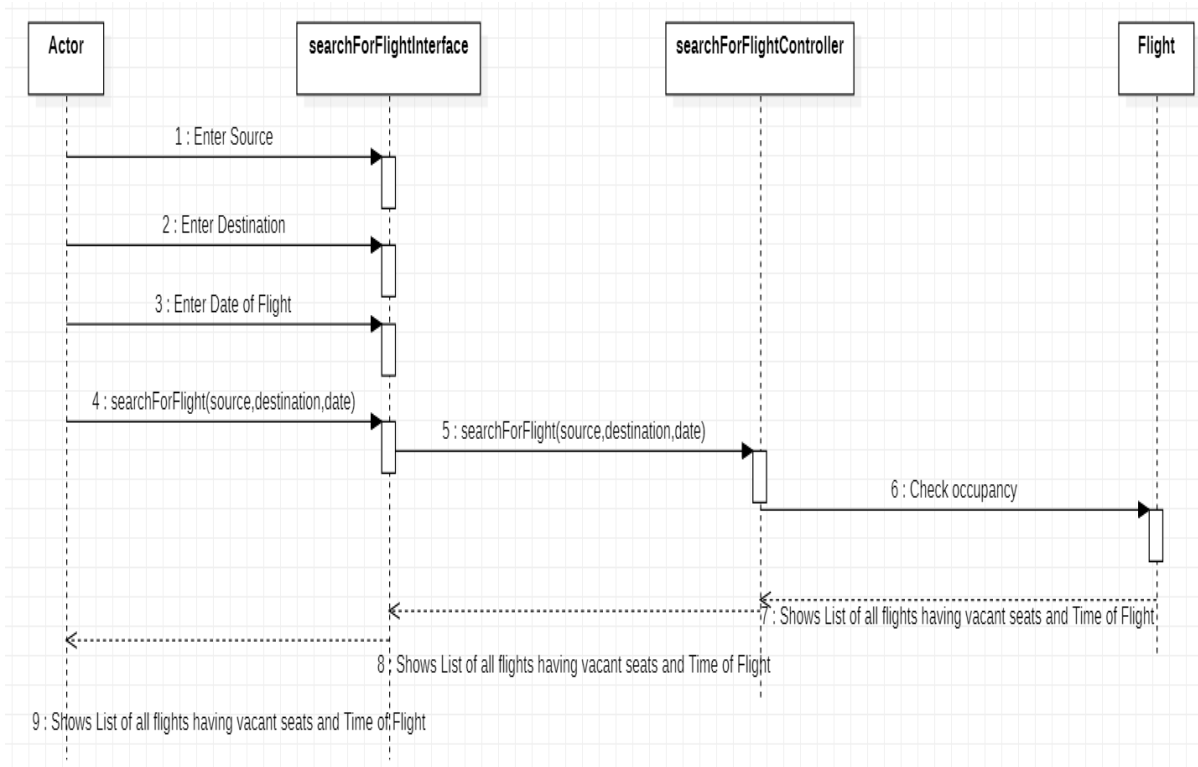


### 3.3.3. ALTERNATIVE FLOW 2: TIME LIMIT EXCEEDED

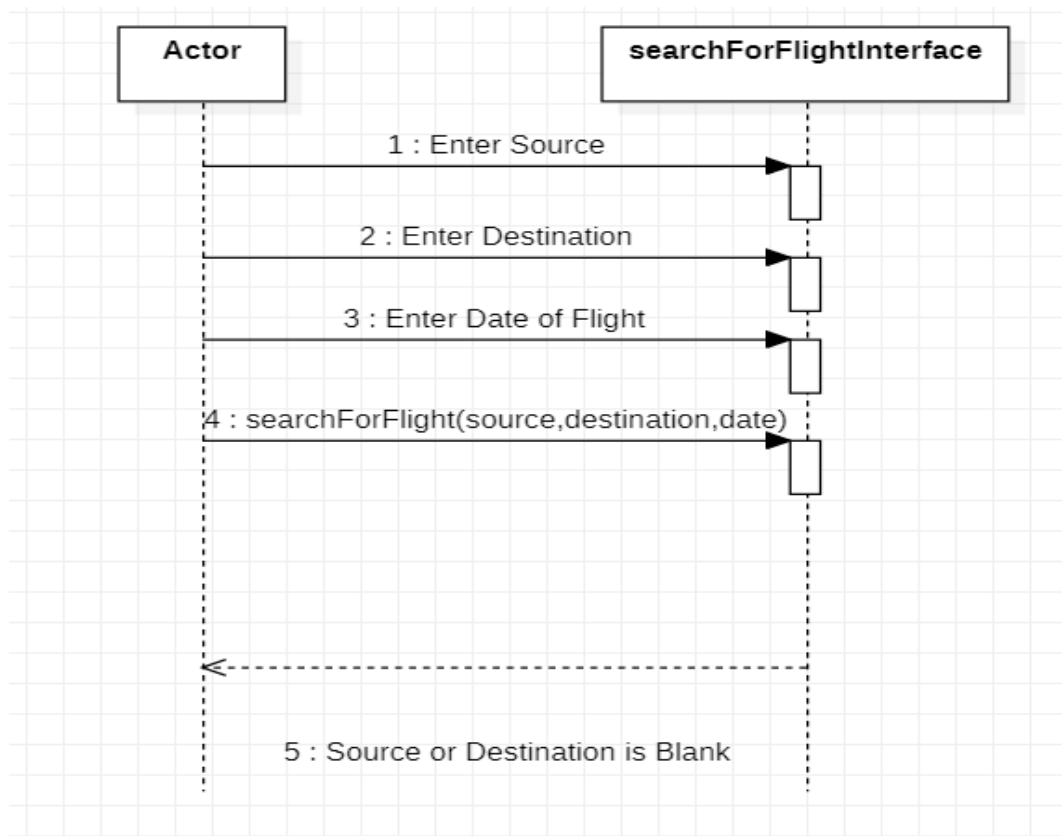


## 3.4. SEARCH FOR FLIGHTS

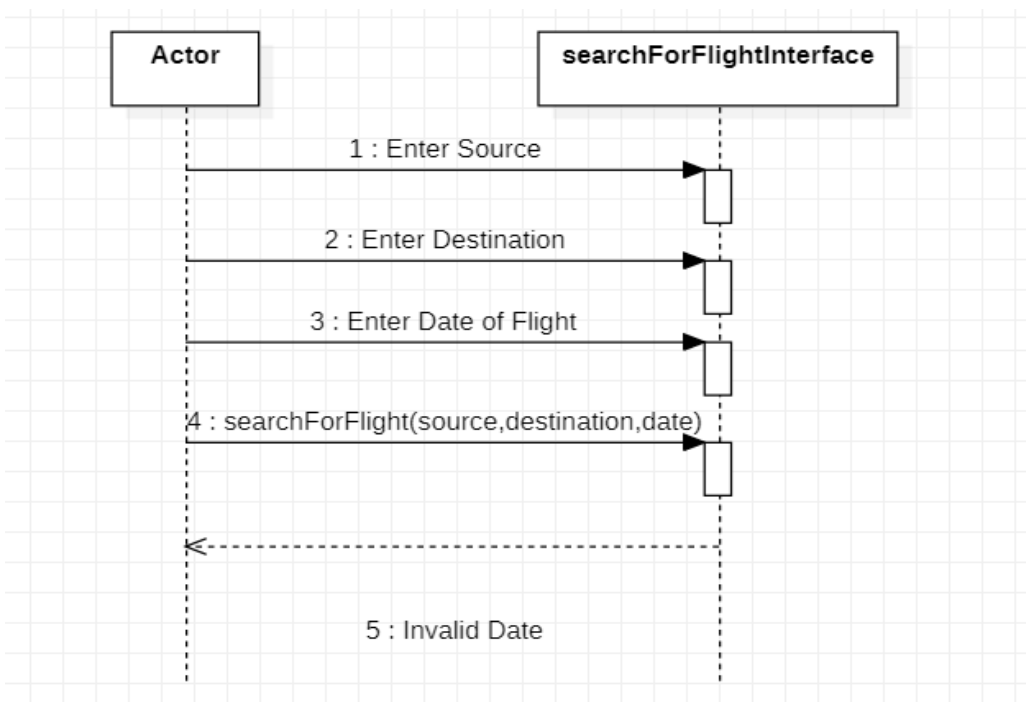
### 3.4.1. BASIC FLOW



### 3.4.2. ALTERNATE FLOW 1: BLANK DETAILS

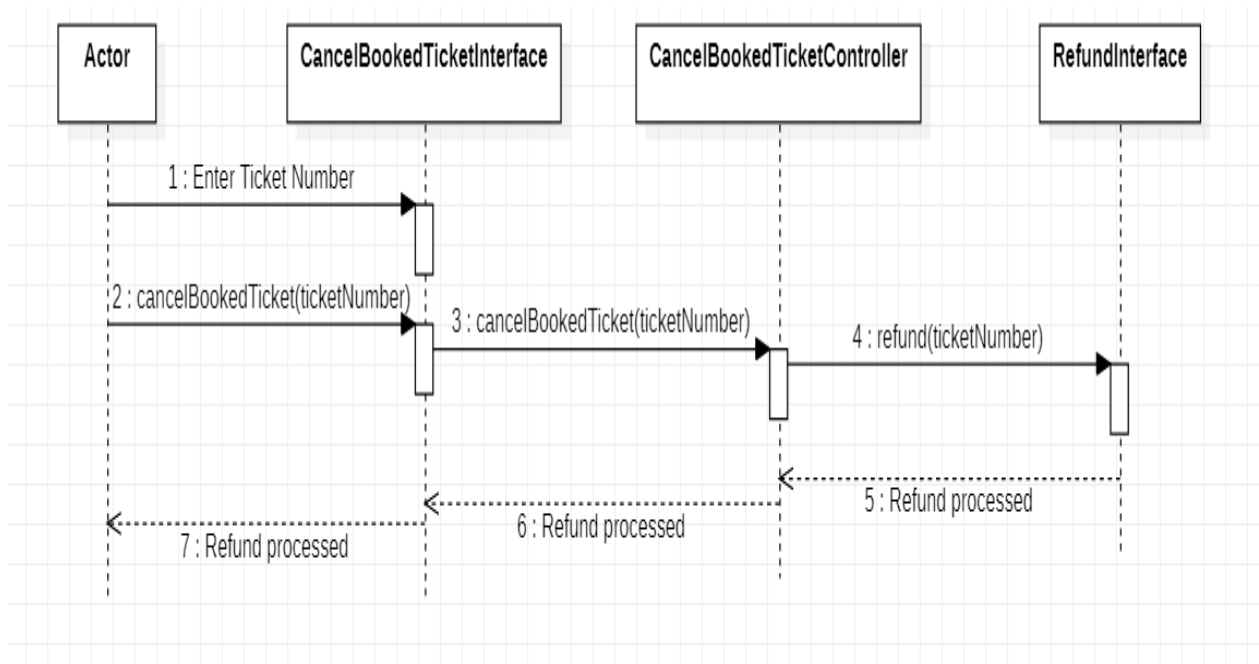


### 3.4.3. ALTERNATIVE FLOW 2: INVALID DATE

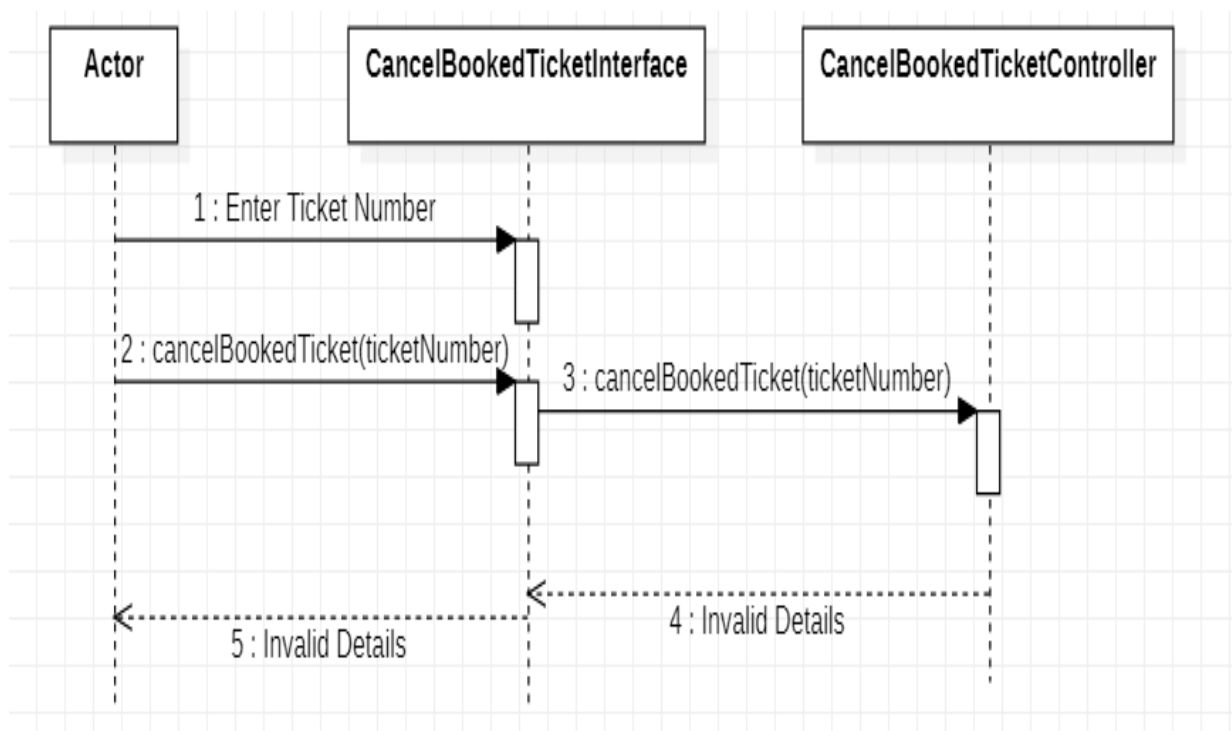


### 3.5. REFUND

#### 3.5.1. BASIC FLOW

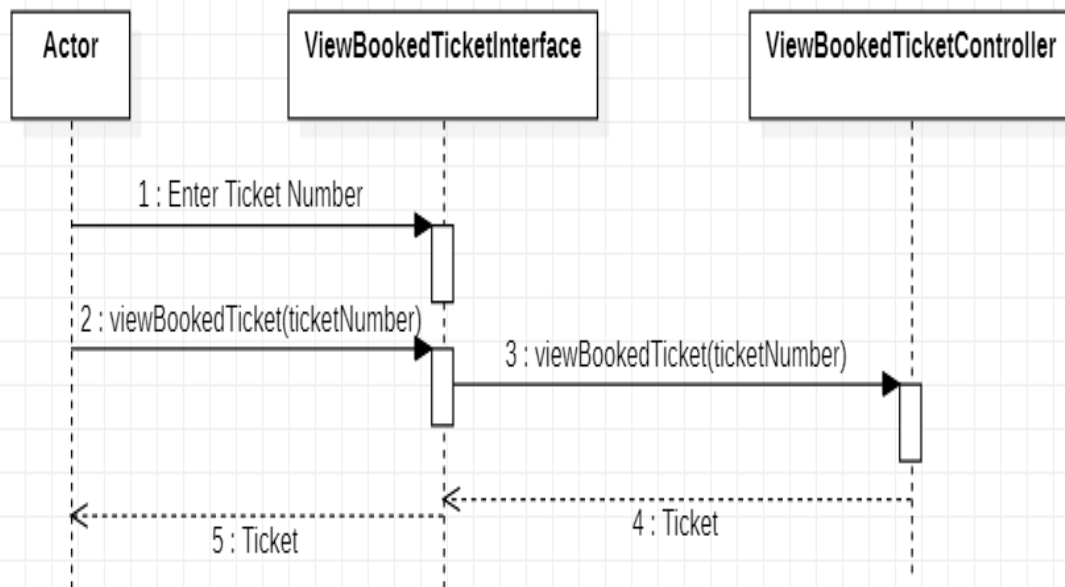


#### 3.5.2. ALTERNATE FLOW: INVALID DETAILS

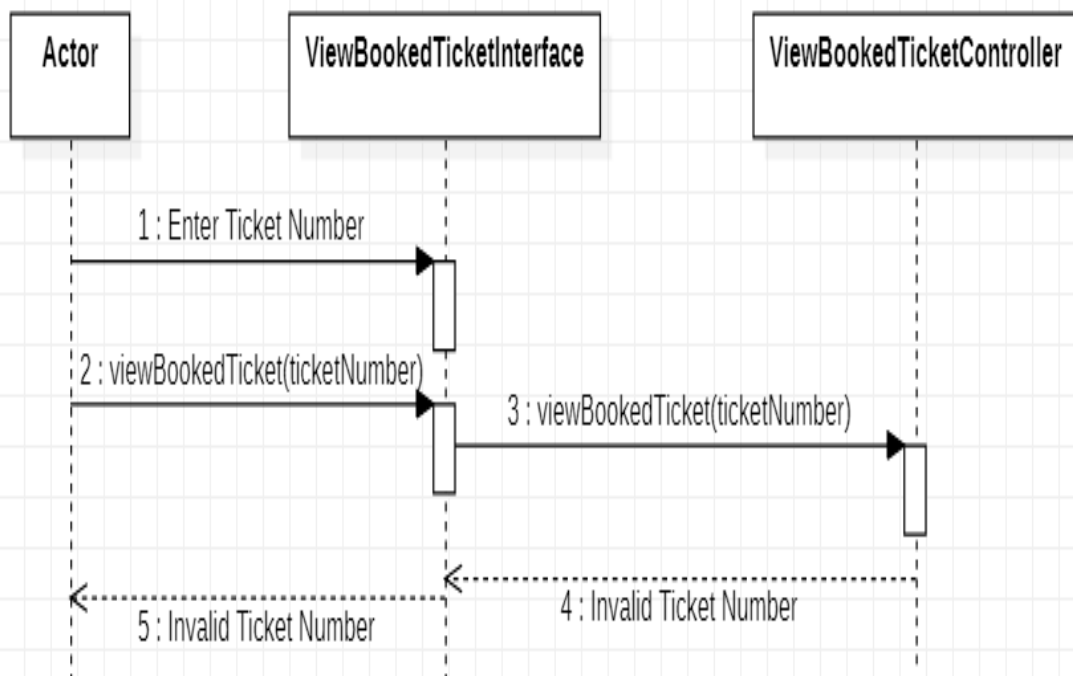


### 3.6. VIEW BOOKED TICKET

#### 3.6.1. BASIC FLOW

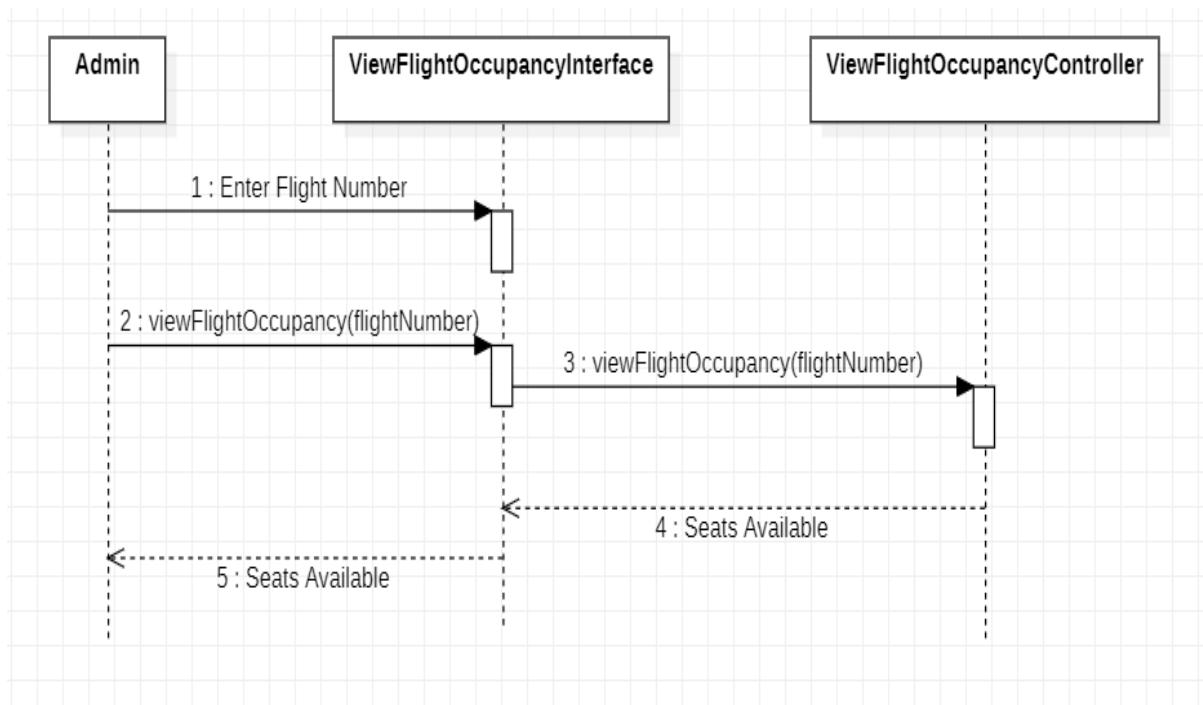


#### 3.6.2. ALTERNATE FLOW

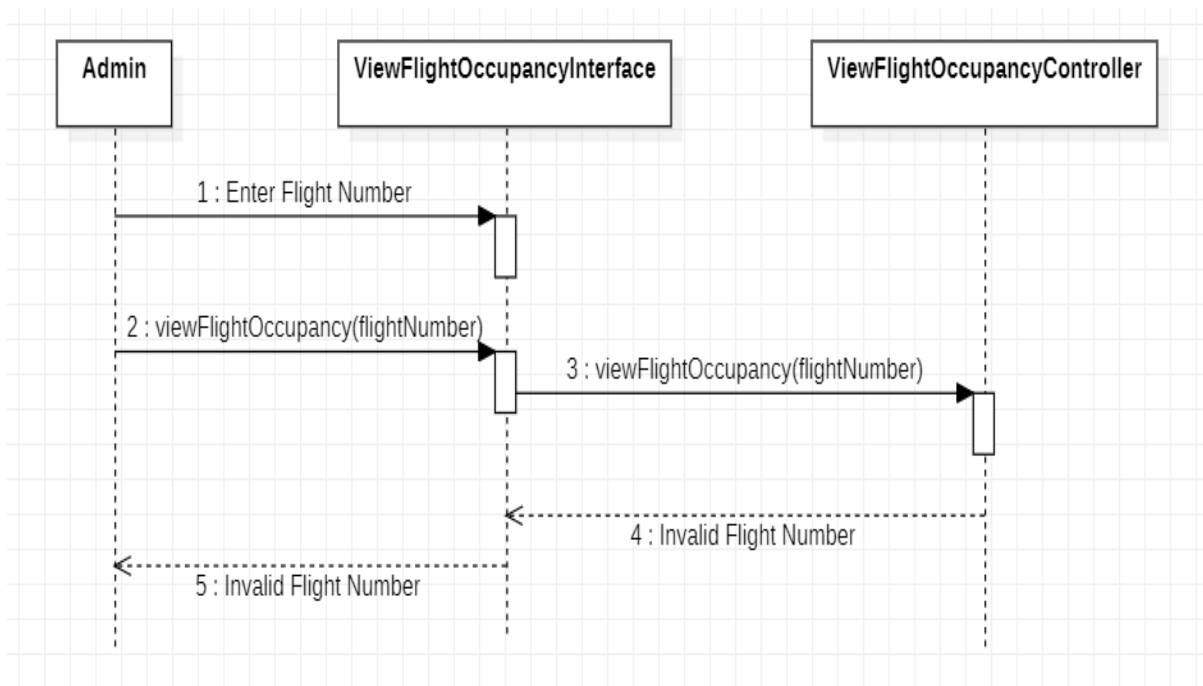


### 3.7. VIEW FLIGHT OCCUPANCY

#### 3.7.1. BASIC FLOW

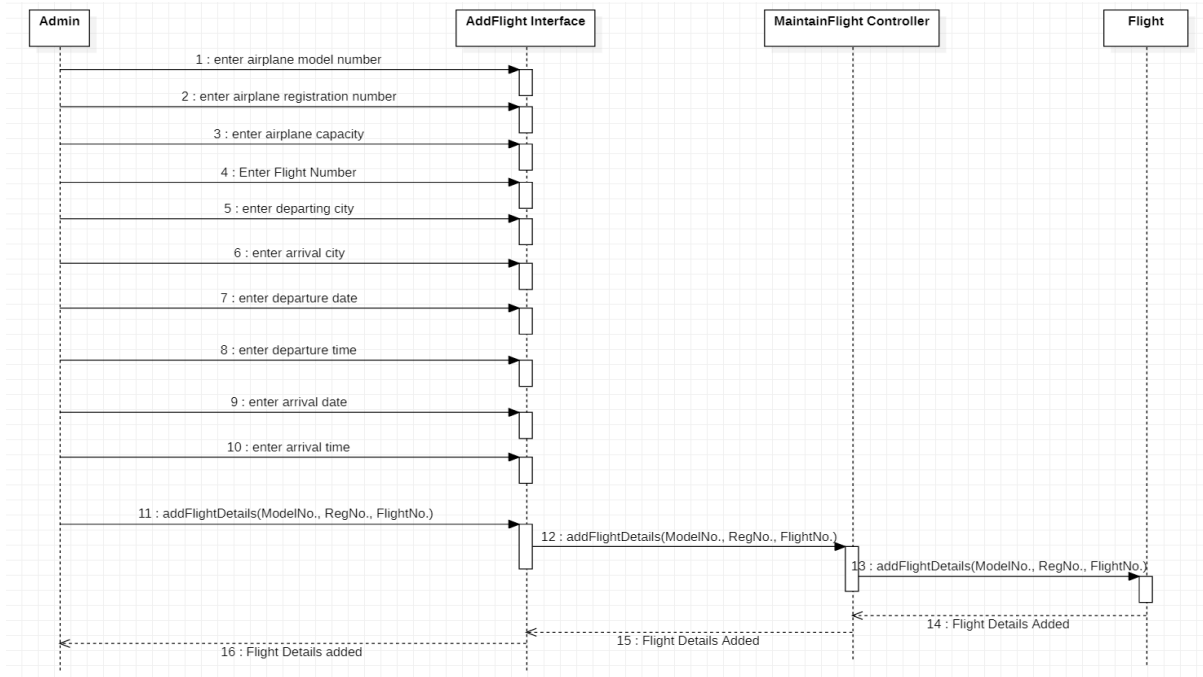


#### 3.7.2. ALTERNATE FLOW: INVALID FLIGHT NUMBER

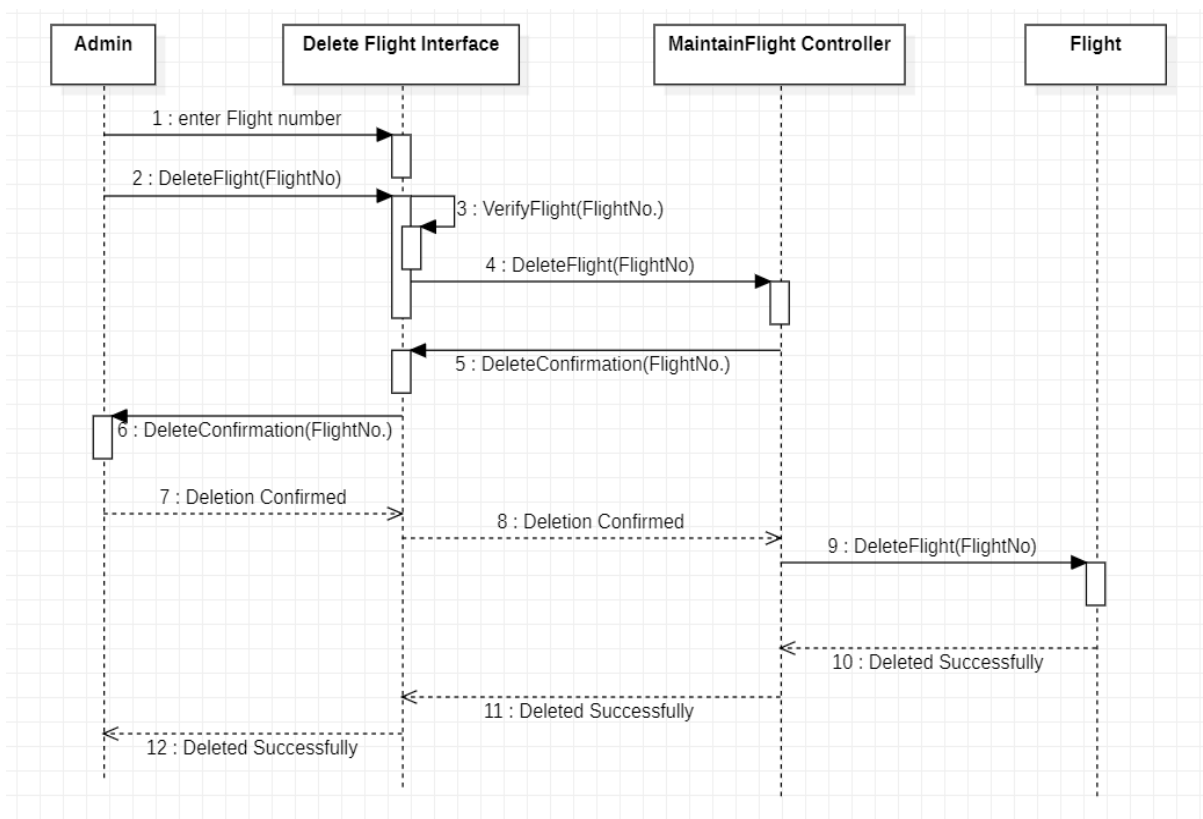


### 3.8. MAINTAIN FLIGHT DETAILS

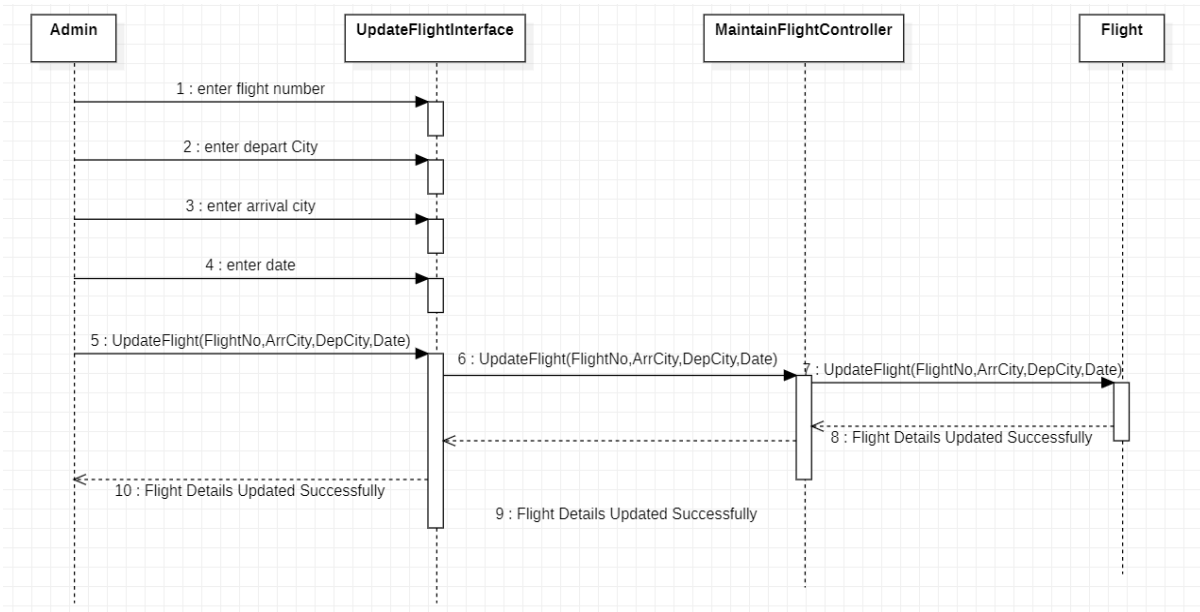
#### 3.8.1. BASIC FLOW 1: ADD FLIGHT



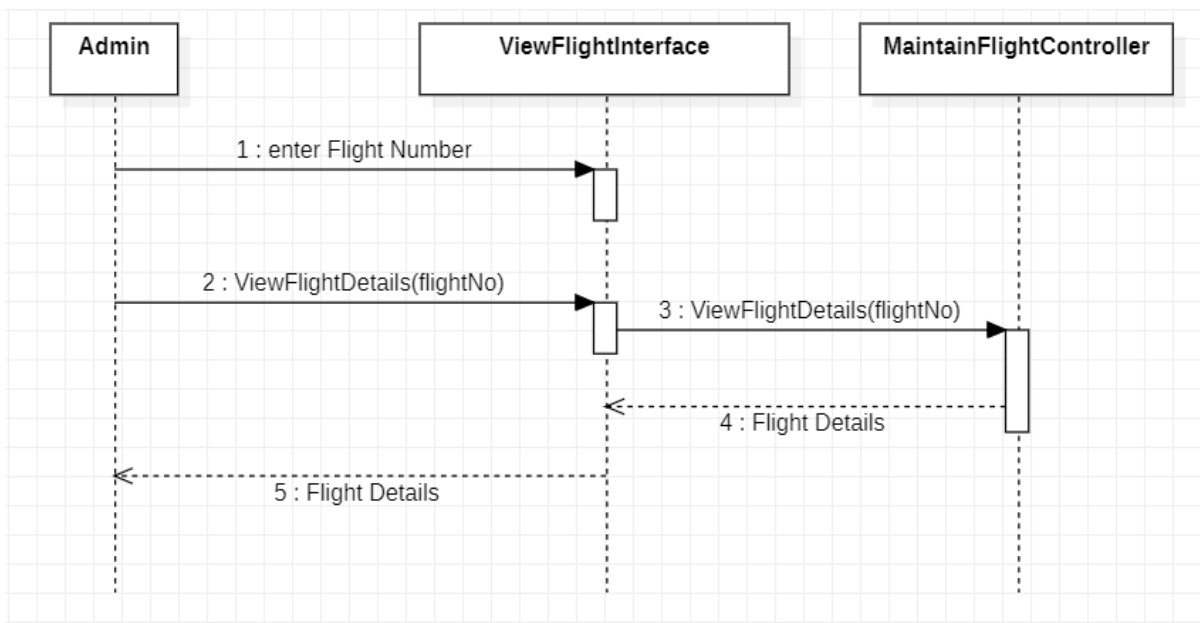
#### 3.8.2. BASIC FLOW 2: DELETE FLIGHT



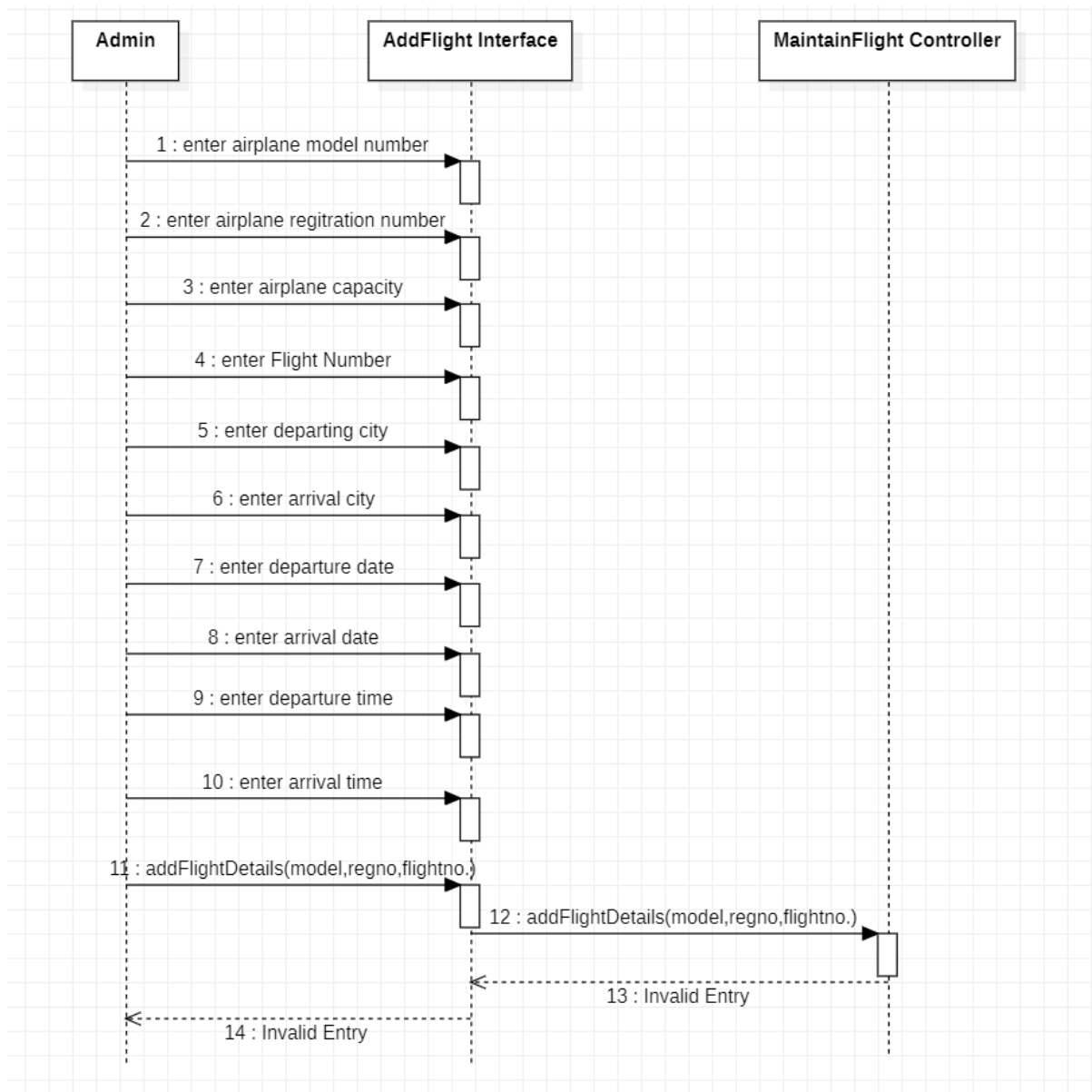
### 3.8.3. BASIC FLOW 3: UPDATE FLIGHT



### 3.8.4. BASIC FLOW 4: VIEW FLIGHT

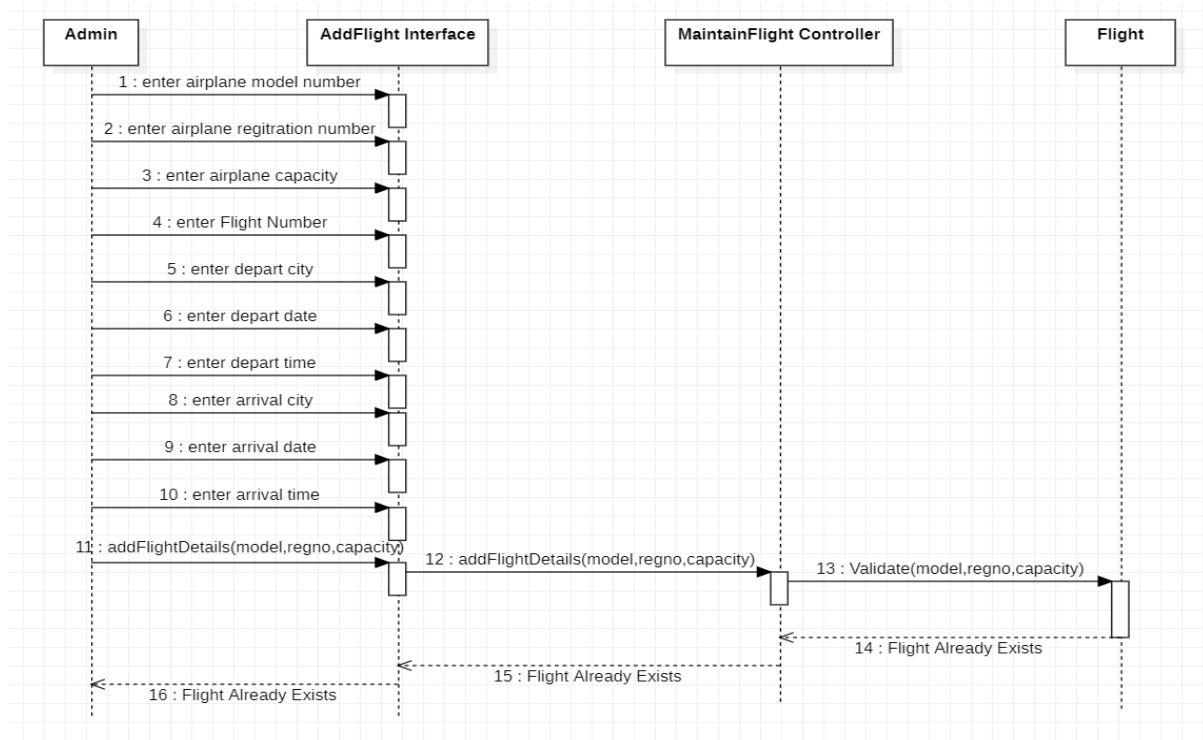


### 3.8.5. ALTERNATE FLOW 1: INVALID ENTRY

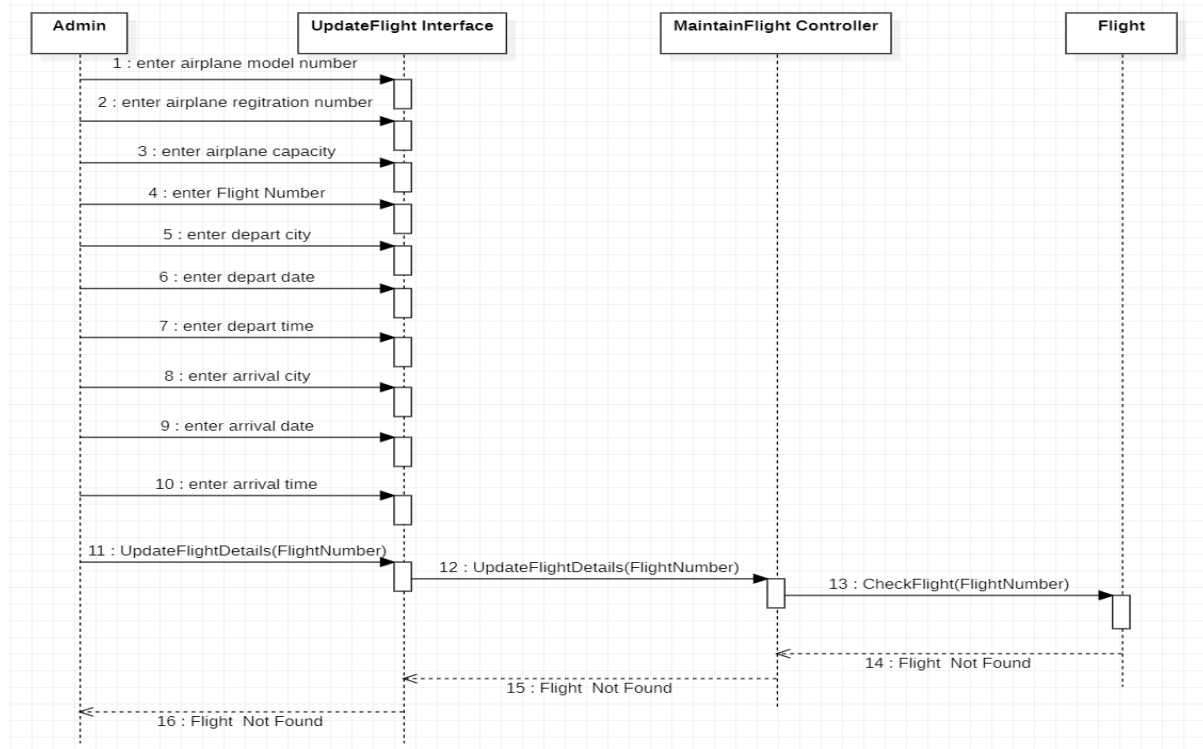




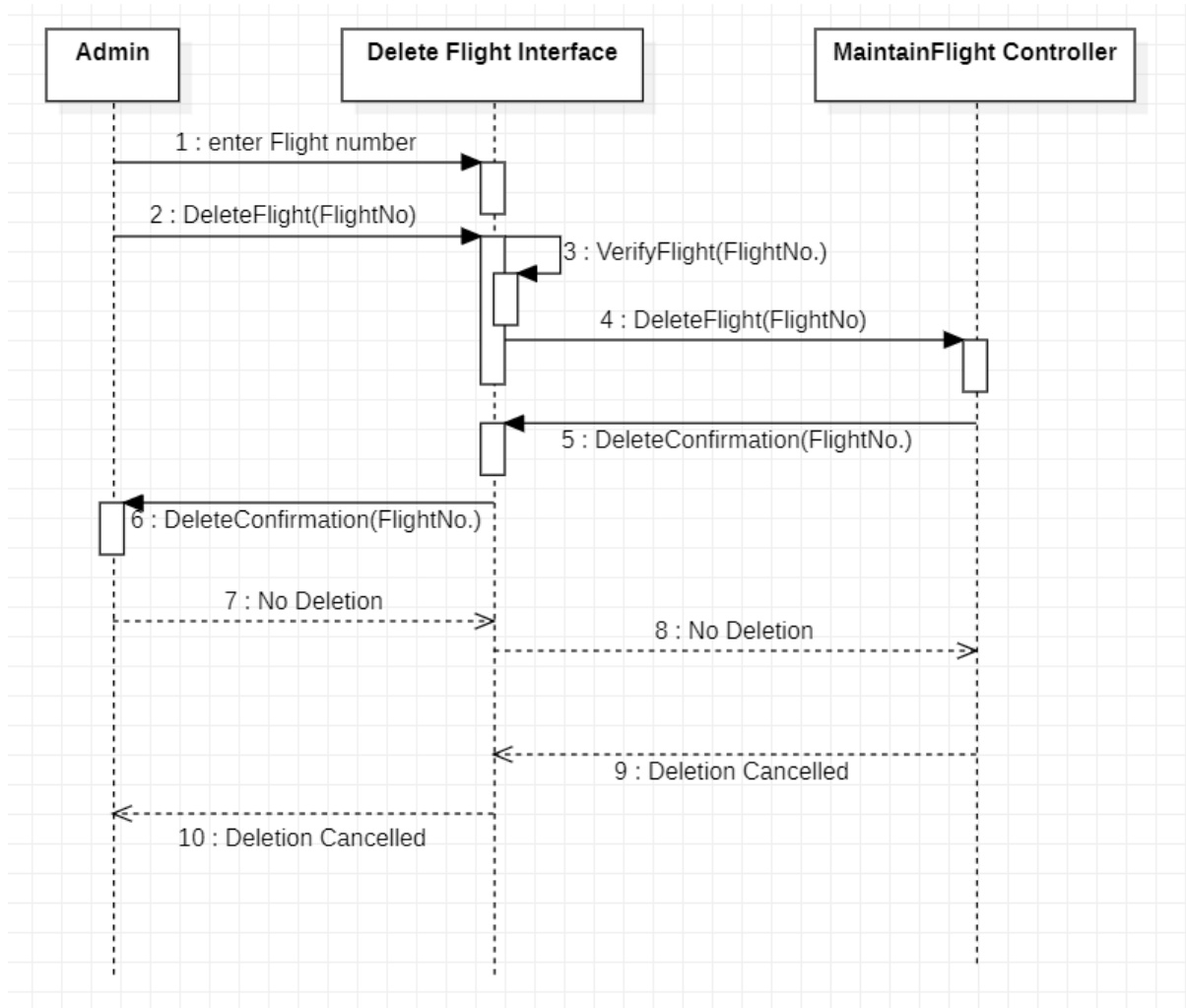
### 3.8.6. ALTERNATIVE FLOW 2: FLIGHT ALREADY EXISTS



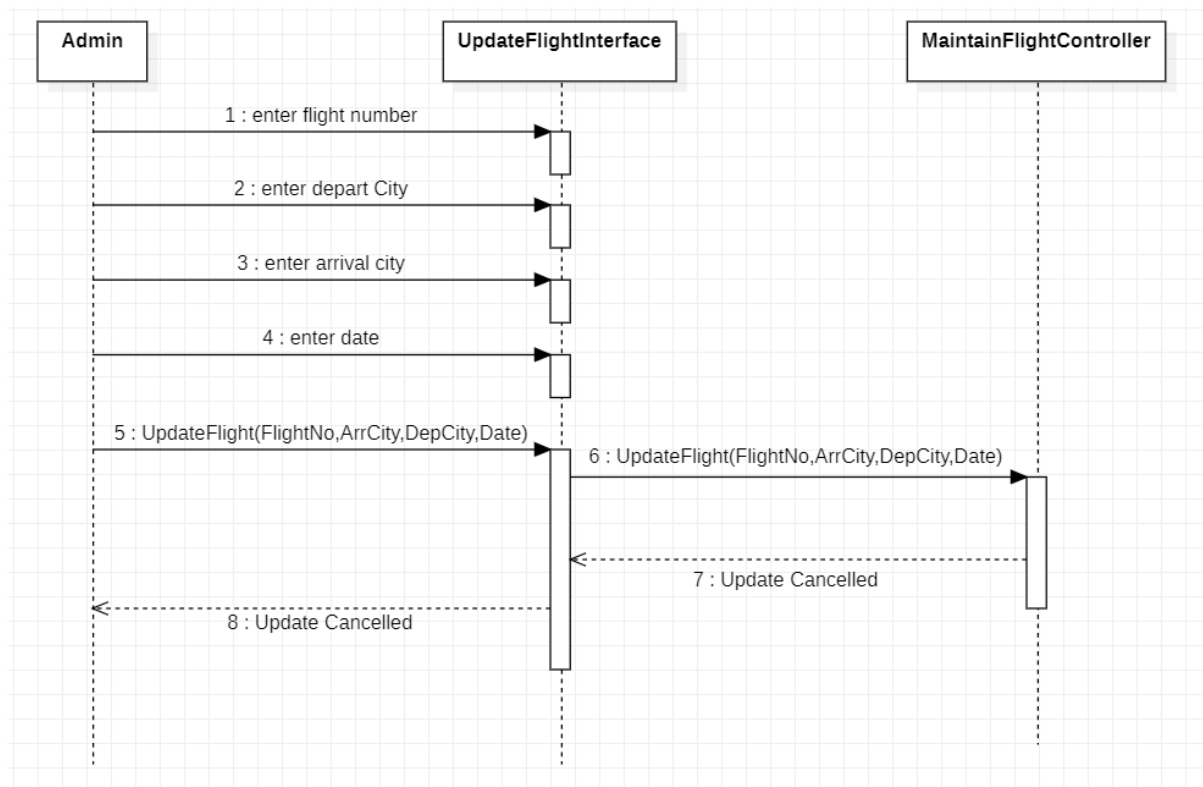
### 3.8.7. ALTERNATIVE FLOW 3: FLIGHT NOT FOUND



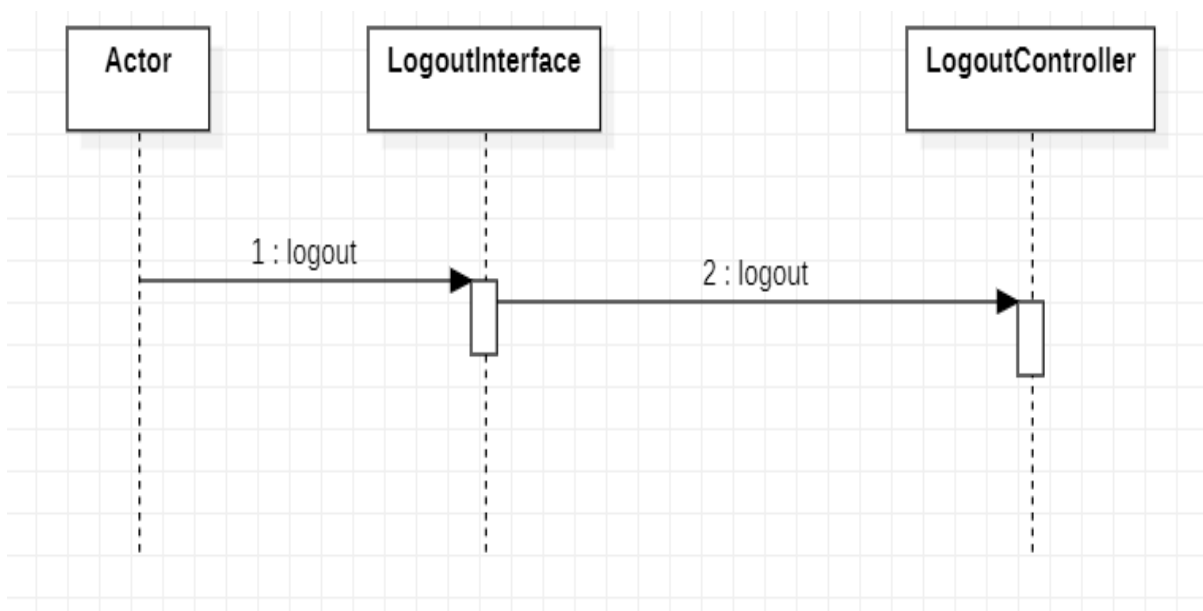
### 3.8.8. ALTERNATIVE FLOW 4: DELETE CANCELLED



### 3.8.9. ALTERNATIVE FLOW 5: UPDATE CANCELLED

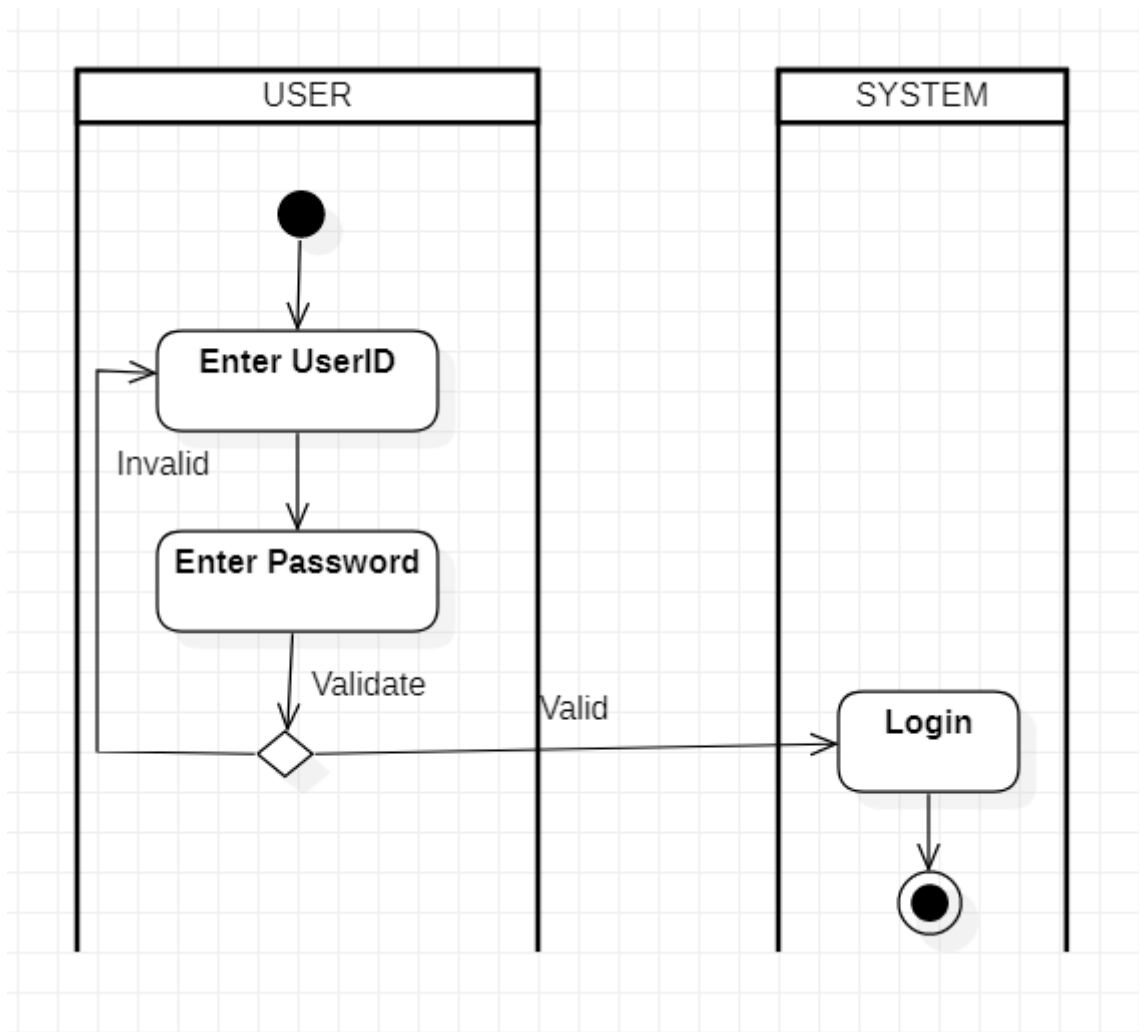


### 3.8.10. ALTERNATIVE FLOW 6: LOGOUT

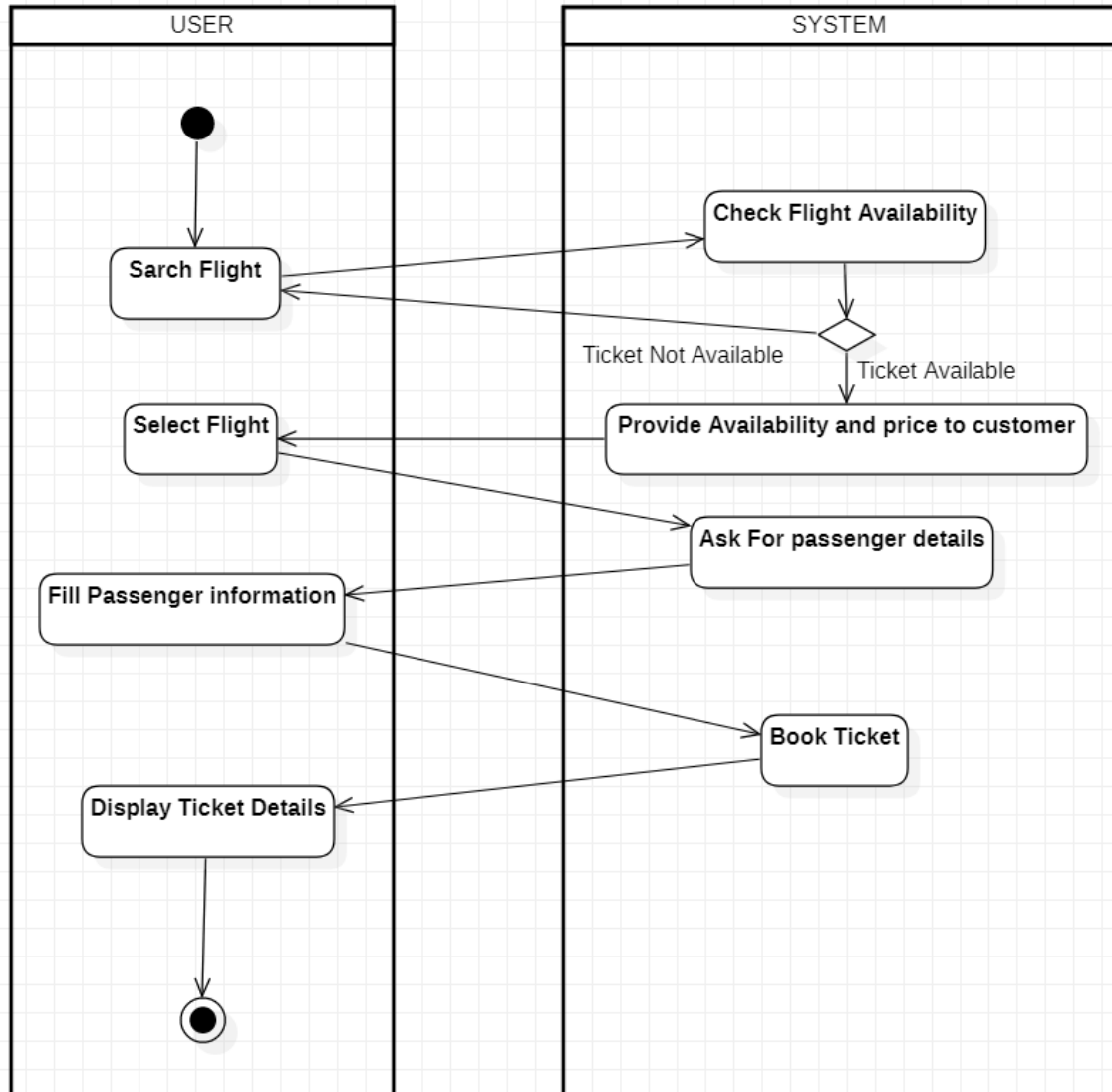


## 4. ACTIVITY DIAGRAM

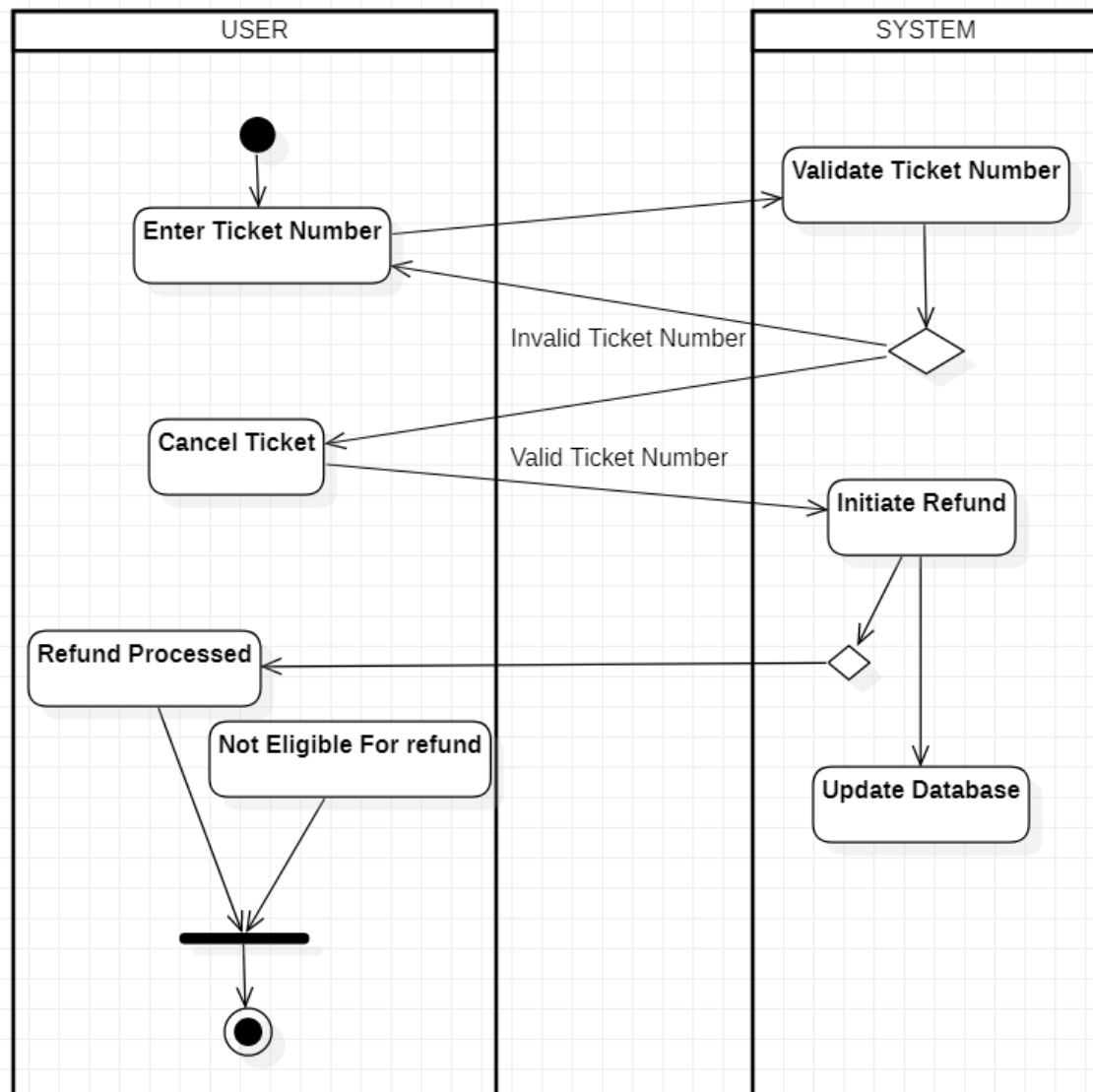
### 4.1. LOGIN



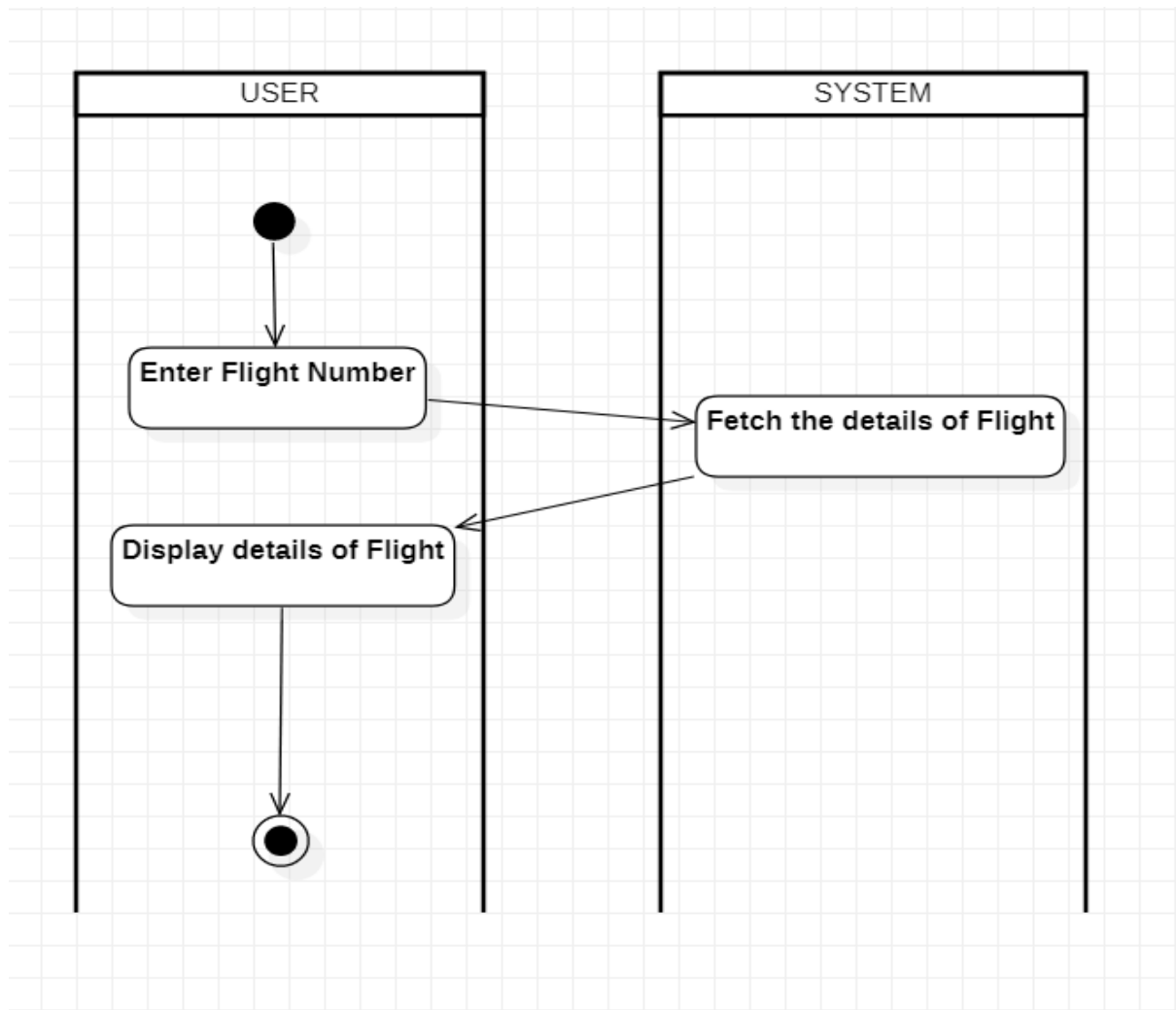
## 4.2. BOOK TICKET



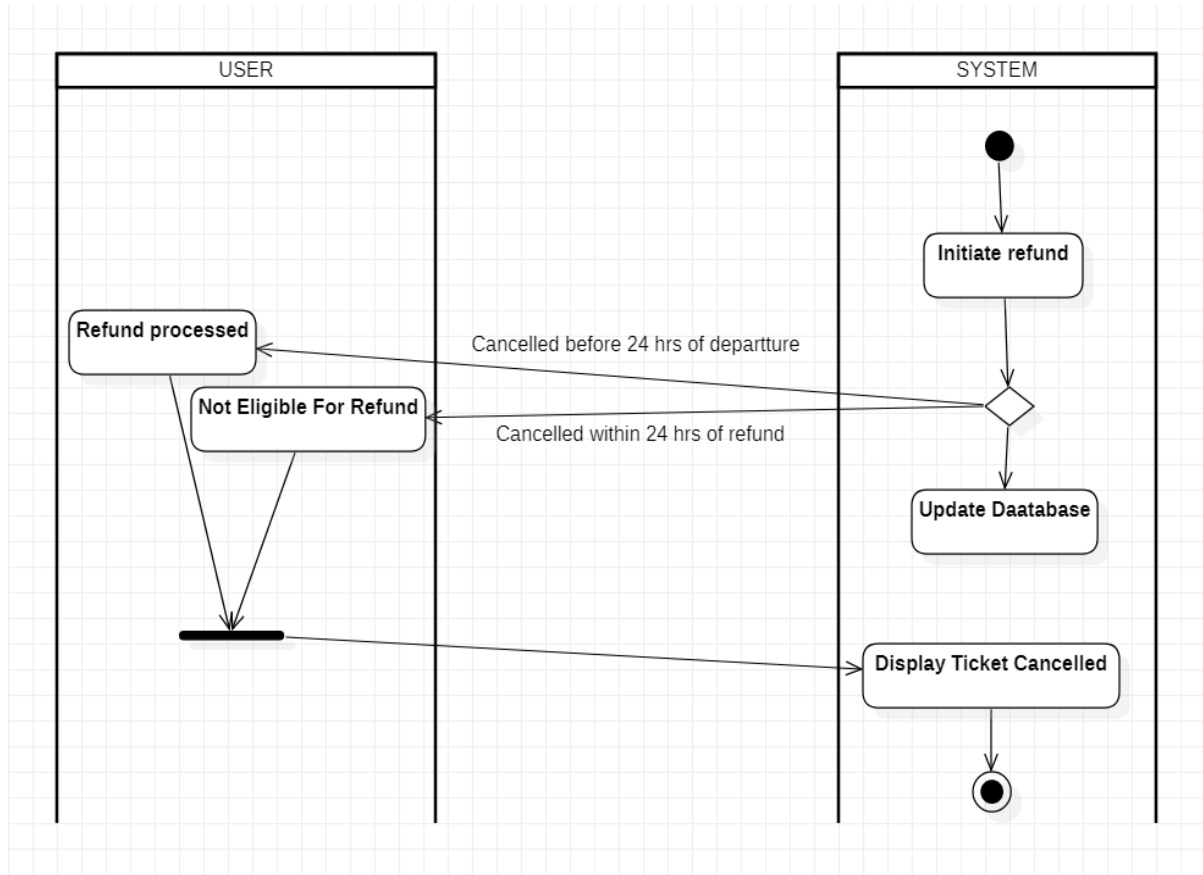
### 4.3. CANCEL TICKET



#### 4.4. VIEW FLIGHT OCCUPANCY

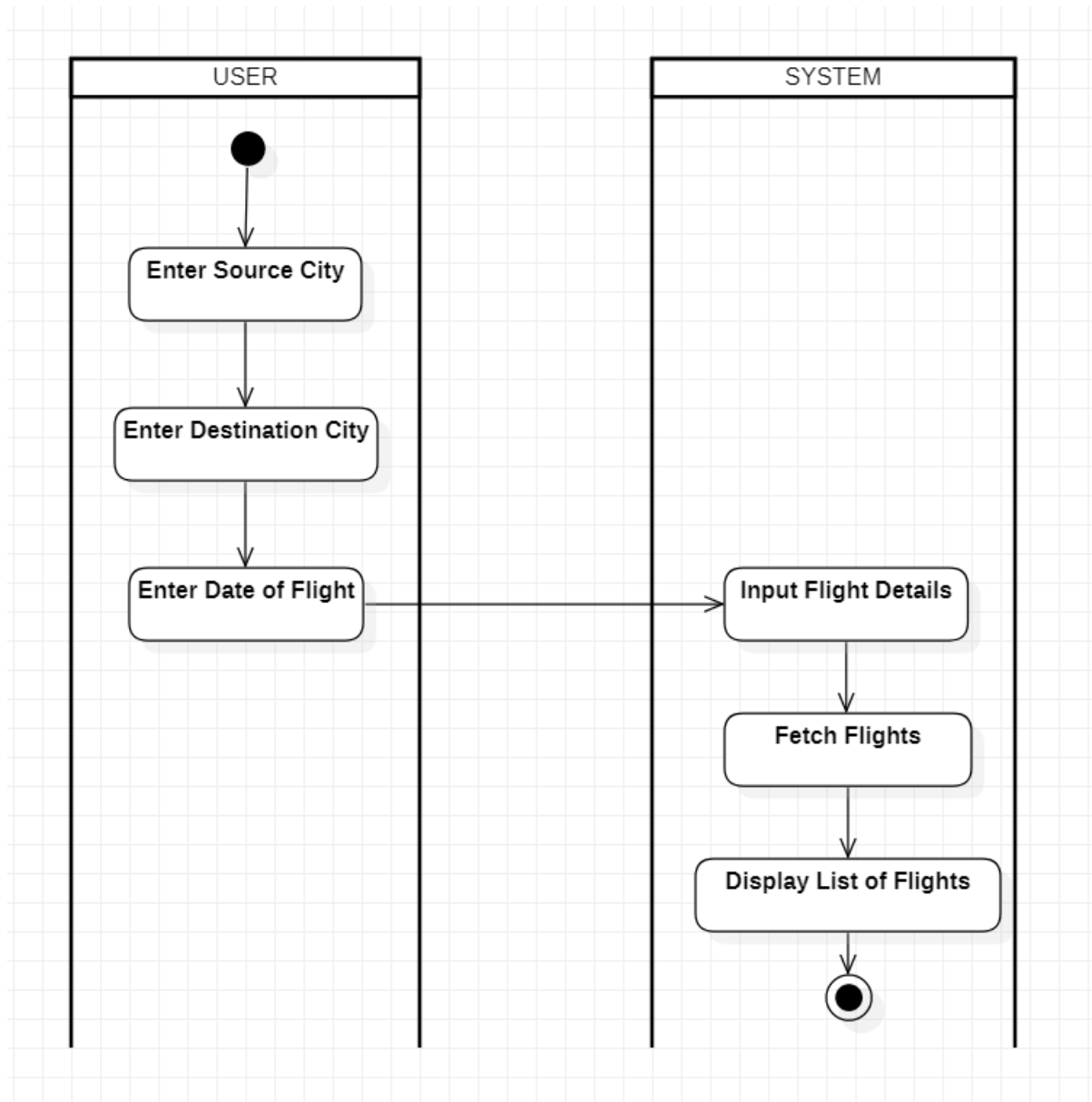


## 4.5. REFUND

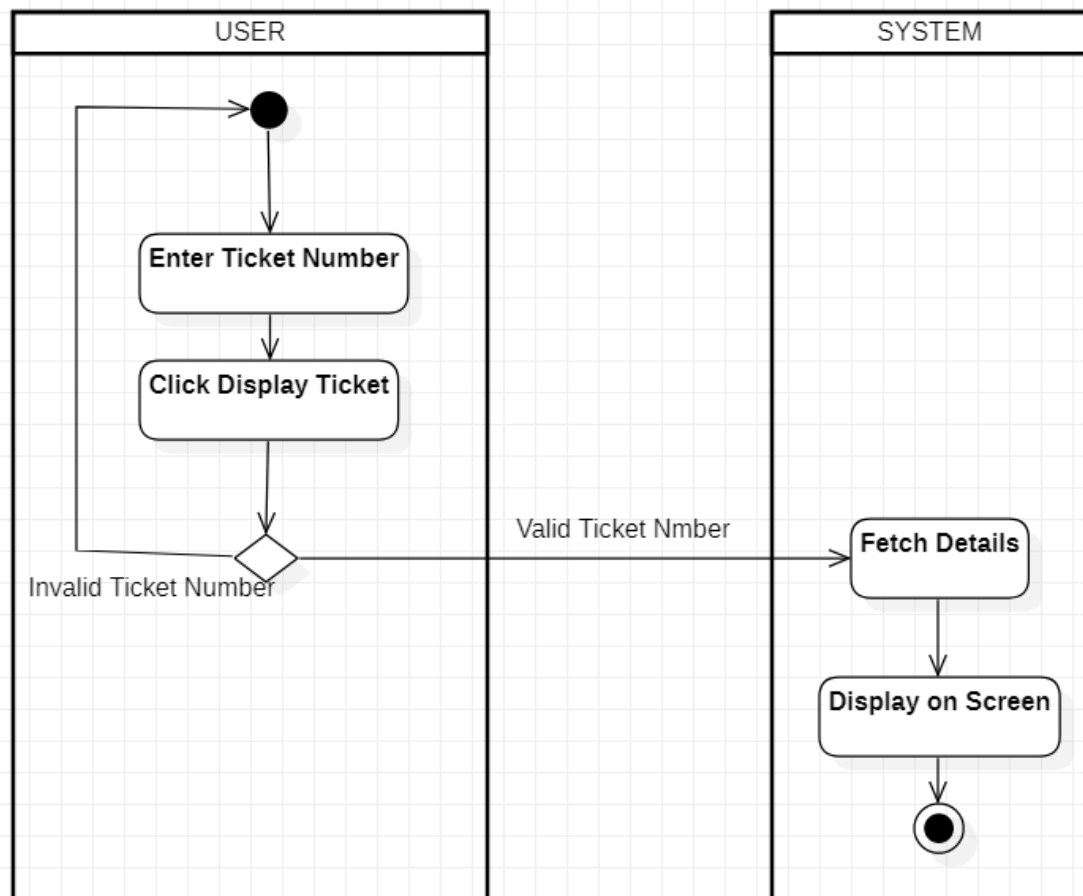




## 4.6. SEARCH FOR FLIGHTS

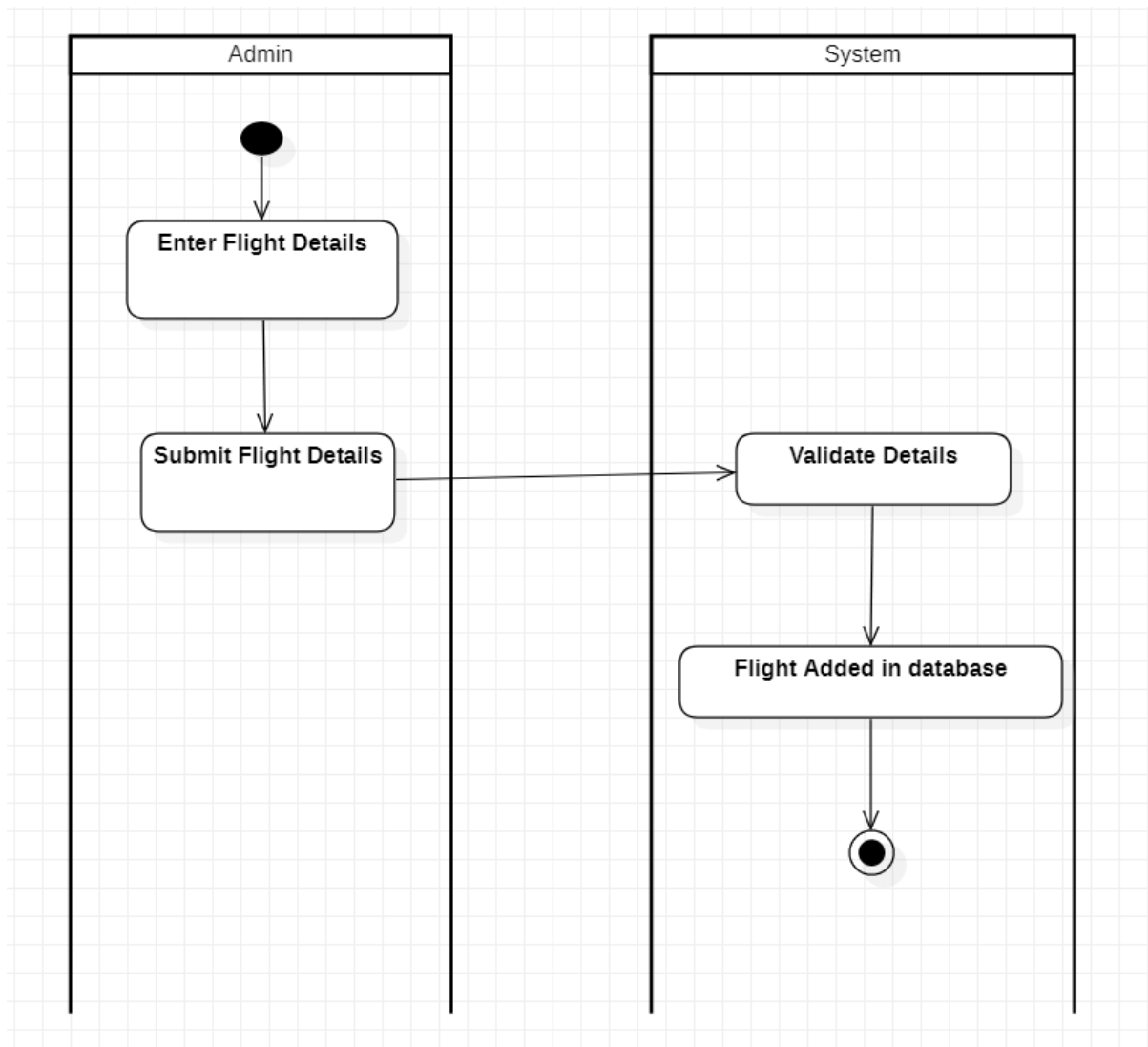


## 4.7. VIEW BOOKED TICKET

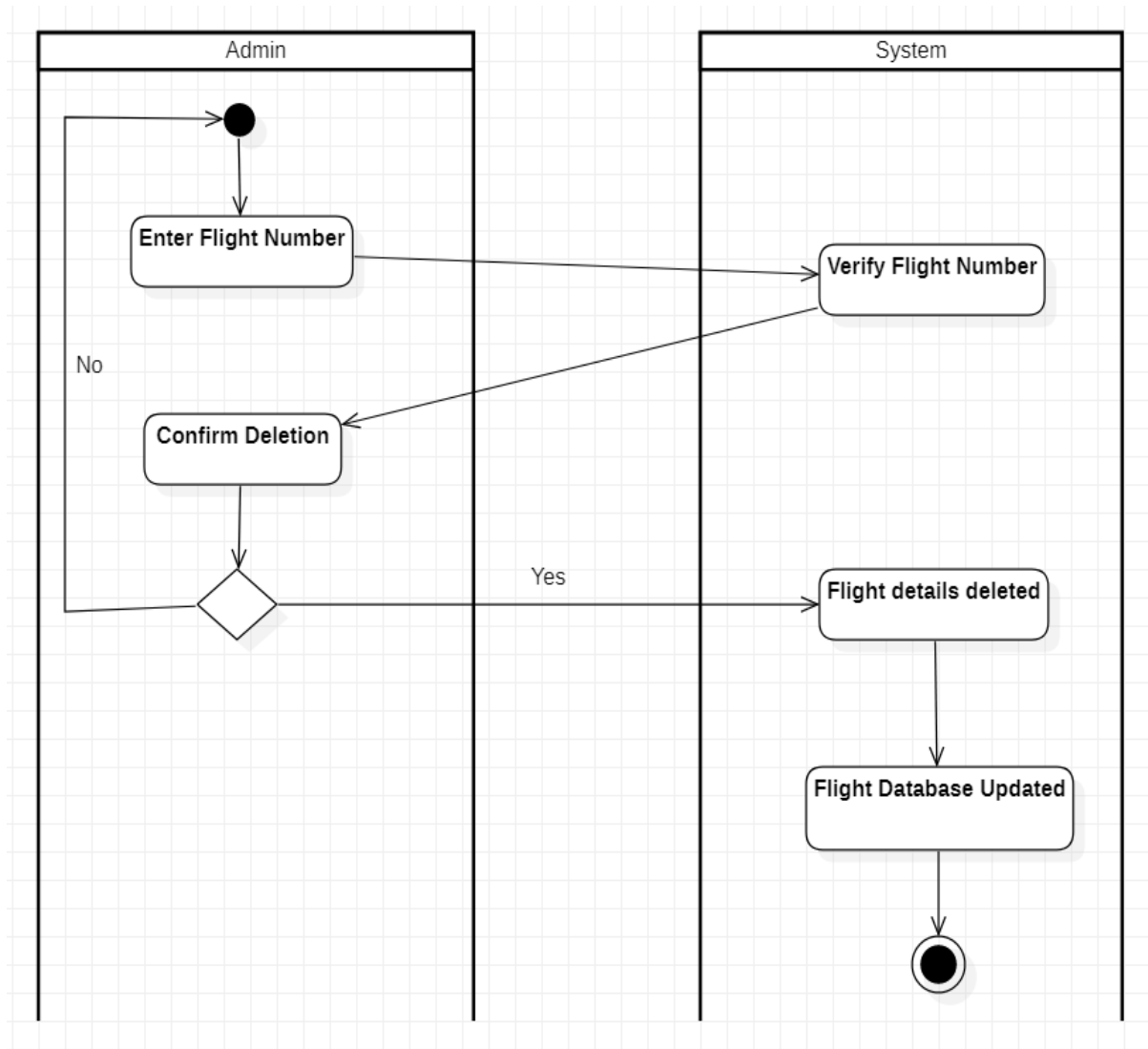


## 4.8. MAINTAIN FLIGHT DETAILS

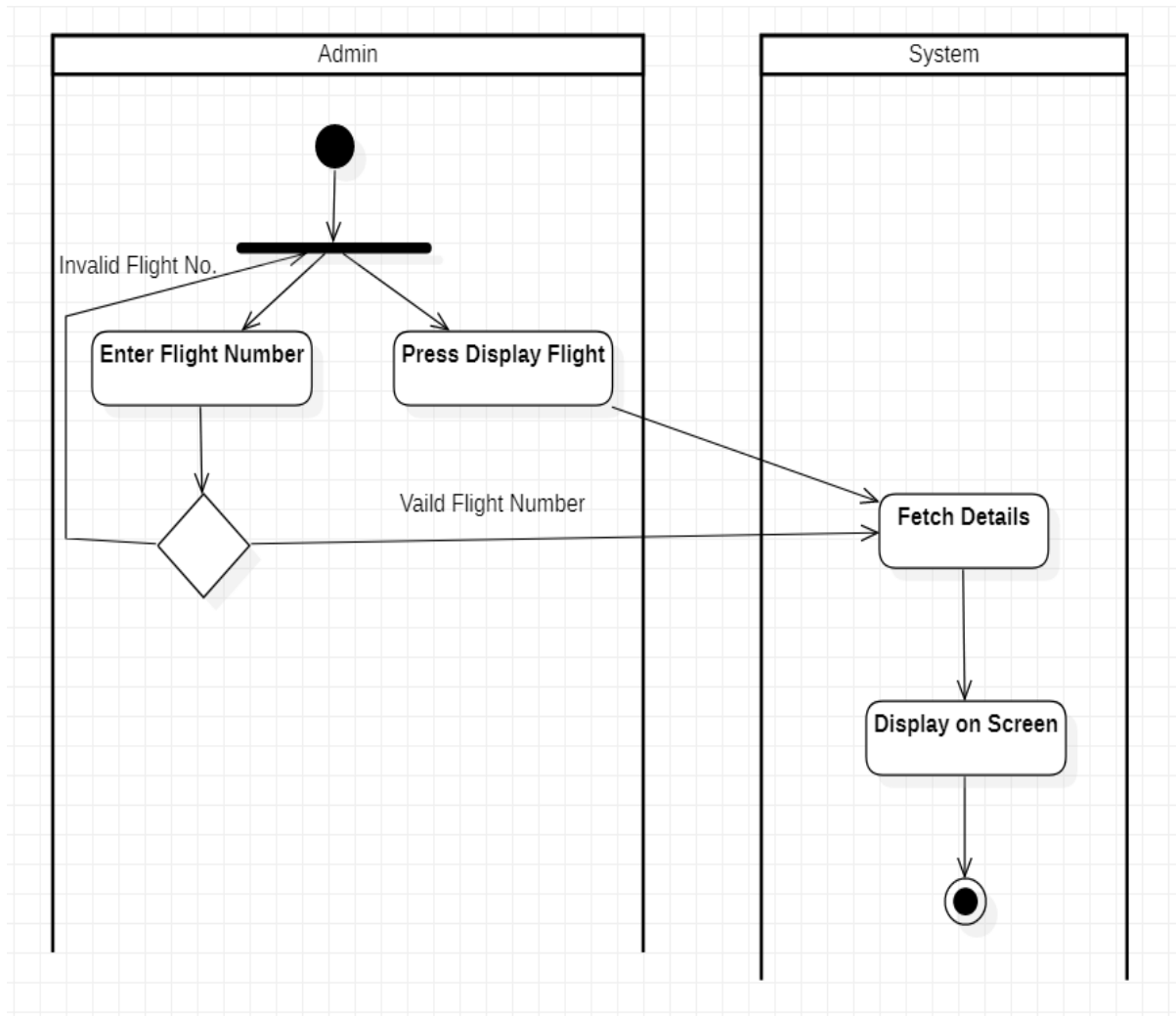
### 4.8.1. ADD FLIGHT



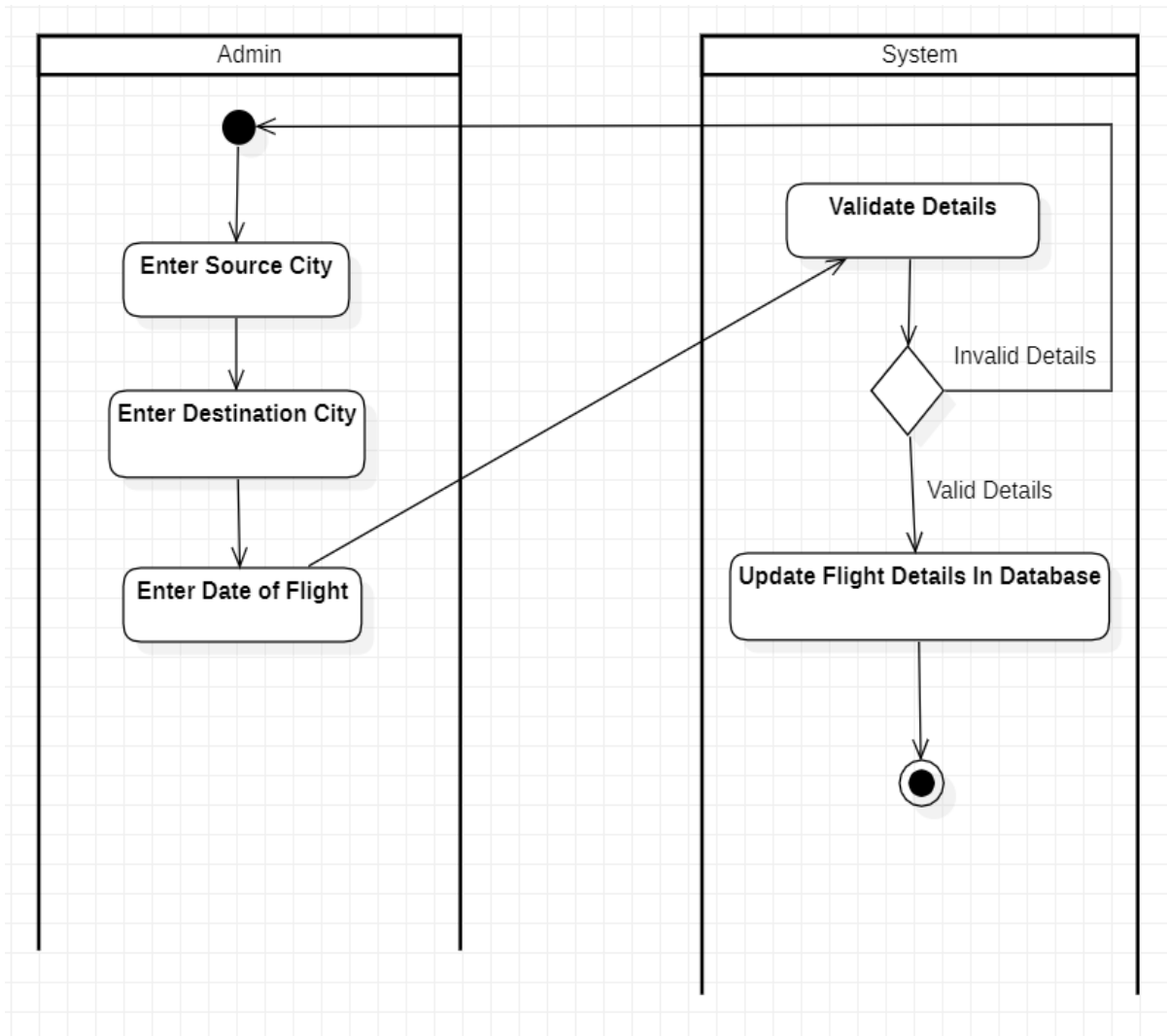
#### 4.8.2. DELETE FLIGHT



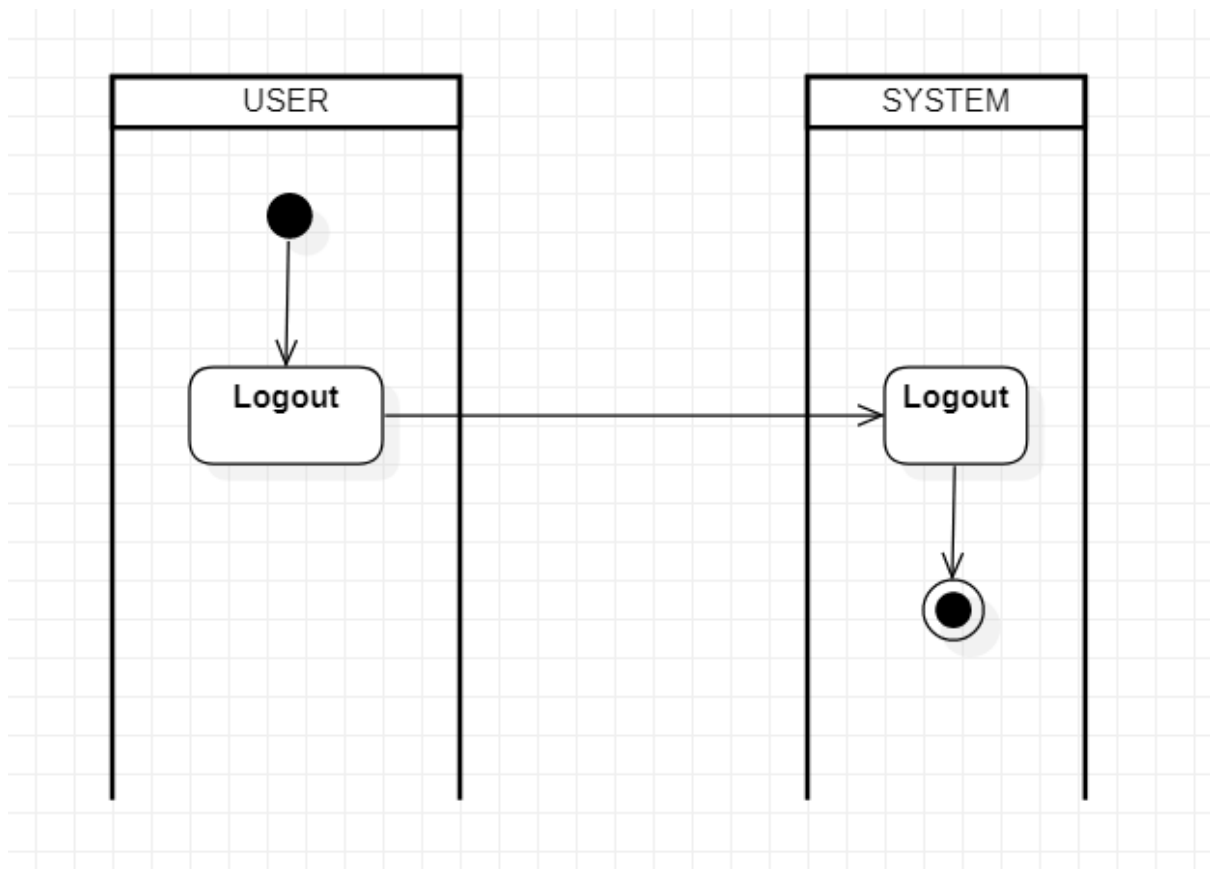
### 4.8.3. VIEW FLIGHT



#### 4.8.4. UPDATE FLIGHT

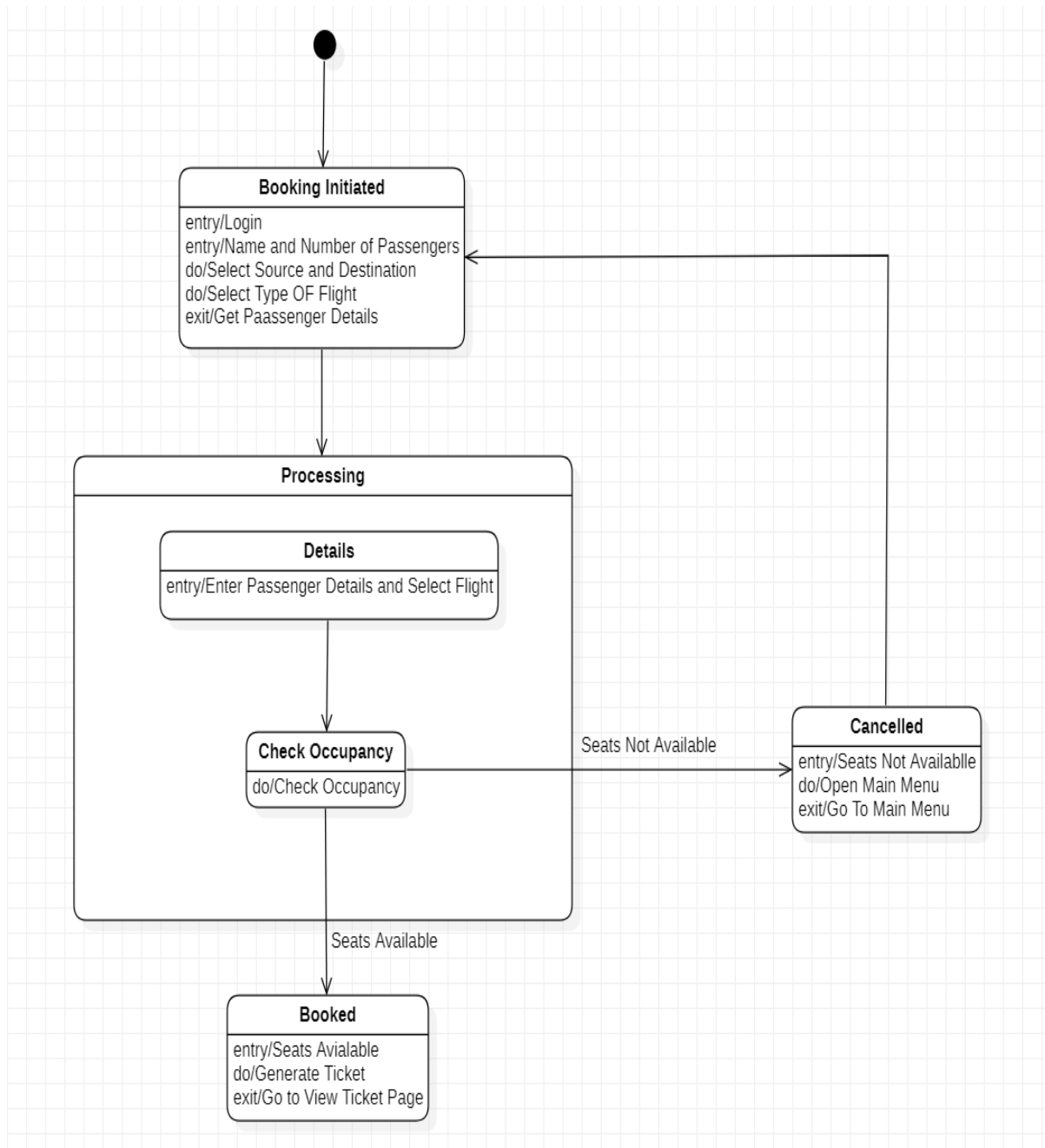


## 4.9. LOGOUT



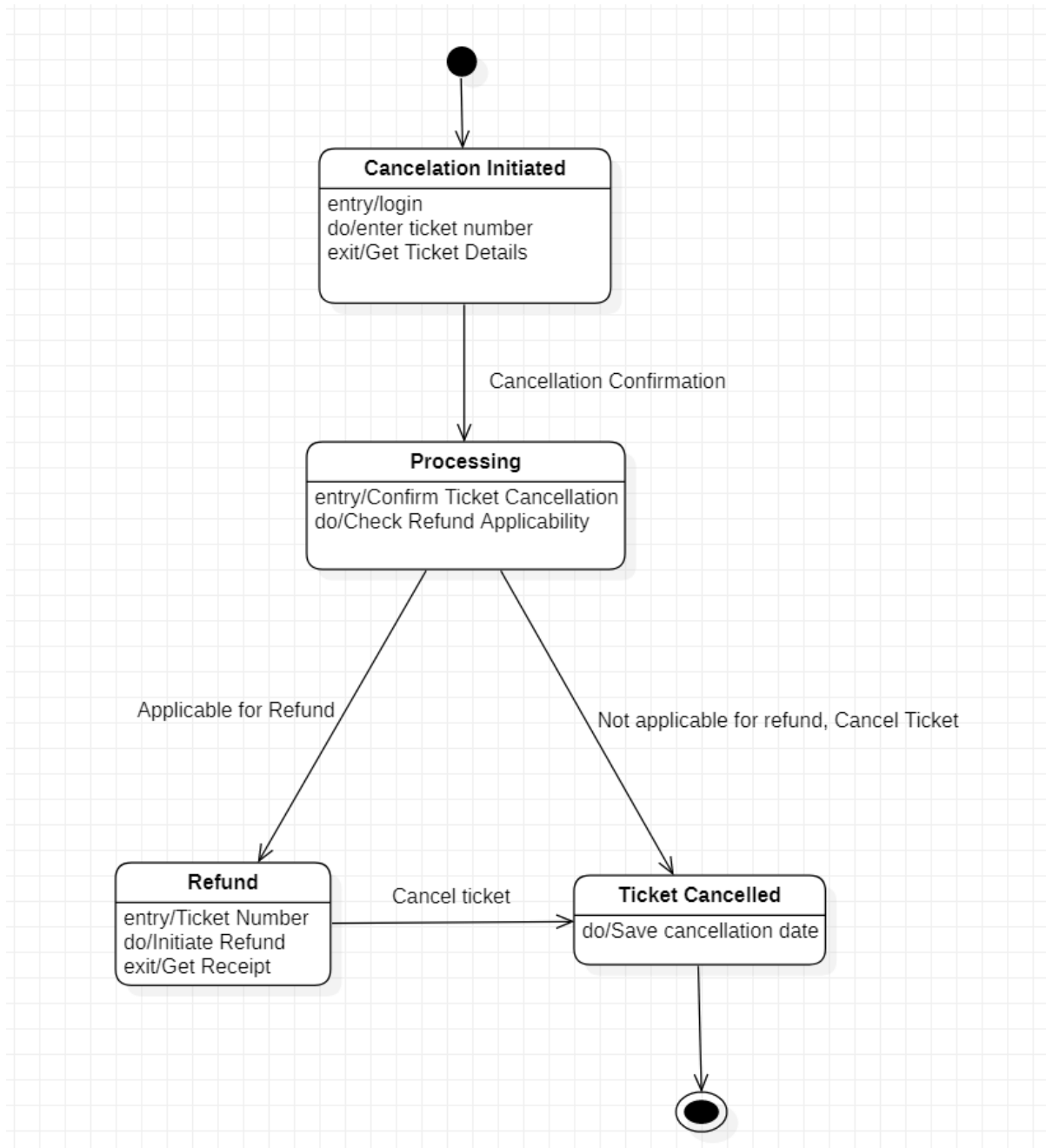
## 5. STATE CHART DIAGRAM

### 5.1. BOOK A TICKET





## 5.2. CANCEL TICKET



## 6. TEST CASE MATRIX

### 6.1. LOGIN

Test Case ID	Scenario name and Description	INPUT 1 USER ID	INPUT 2 PASSWORD	Expected Output	Remarks
TC <sub>1</sub>	Scenario 1 – Login	Valid	Valid	User is allowed to Login	-
TC <sub>2</sub>	Scenario 2 – Login Alternative flow: Invalid entry	Invalid	Valid	User ID is invalid	USER ID is not in specified format.
TC <sub>3</sub>		Invalid	Valid	User ID is Invalid	User ID does not exist in database.
TC <sub>4</sub>		Valid	Invalid	Password is Invalid	Password is not in the specified format.
TC <sub>5</sub>		Valid	Invalid	Password is Invalid	Password does not exist in database.
TC <sub>6</sub>		Invalid	Invalid	User ID and Password are invalid	Login ID and password are not in the specified format.
TC <sub>7</sub>	Scenario 3 – Login Alternative flow: User Exits	Valid/ Invalid Input	Valid/ Invalid Input	User Comes out of the system	-

## 6.2. SEARCH FOR FLIGHT

Test Case ID	Scenario Name and Description	INPUT 1 Source City	INPUT 2 Destination City	INPUT 3 Date of Flight	Expected Output	Remarks
TC <sub>1</sub>	Scenario 1- Search for Flights	Valid	Valid	Valid	Flights Searched Successfully	-
TC <sub>2</sub>	Scenario 2- Search for Flights Alternative Flow: Invalid Entry	Invalid/ Blank	Valid/ Invalid	Valid/ Invalid	Source City is Invalid	Source City entered is invalid
TC <sub>3</sub>		Valid	Invalid/ Blank	Valid/ Invalid	Destination City is Invalid	Destination City entered is Invalid
TC <sub>4</sub>		Valid	Valid	Invalid/ Blank	Date of Flight is invalid	Date of Flight entered is Invalid
TC <sub>5</sub>	Scenario 3- Search for Flights Alternative Flow: User Exits	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	User comes out of the system	-

### 6.3. BOOK TICKET (Condition: Flights and seats are Available)

Test Case ID	Scenario Name and Description	INPUT 1 Name	INPUT 2 Number of Passengers	INPUT 3 Source	INPUT 4 Destination	INPUT 5 Type of Flight	Expected Output	Remarks
TC <sub>1</sub>	Scenario 1- Book Ticket	Valid	Valid	Valid	Valid	Valid	Ticket Booked Successful	-
TC <sub>2</sub>	Scenario 2- Book Ticket Alternate Flow: Invalid Details	Invalid/ Blank	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Invalid Name	Name is not in the specified format
TC <sub>3</sub>		Valid	Invalid/ Blank	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Invalid Number of Passengers	Number of passengers is not in the specified format
TC <sub>4</sub>		Valid	Valid	Invalid/ Blank	Valid/ Invalid	Valid/ Invalid	Invalid Source	Source is not in the specified format
TC <sub>5</sub>		Valid	Valid	Valid	Invalid/ Blank	Valid/ Invalid	Invalid Destination	Destination is not in the specified format
TC <sub>6</sub>		Valid	Valid	Valid	Valid	Invalid/ Blank	Invalid Type of Flight	Type of Flight is not in the specified format
TC <sub>7</sub>	Scenario 3- Book Ticket Alternate Flow: User Exits	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	Valid/ Invalid	User comes out of the system	-

## 6.4. VIEW FLIGHT OCCUPANCY

Test case ID	Scenario Name and Description	INPUT 1 Enter Flight Number	Expected Output	Remarks
TC <sub>1</sub>	Scenario 1- View Flight Occupancy	Valid	Flight Occupancy Shown	-
TC <sub>2</sub>	Scenario 2- View Flight Occupancy- Alternative Flow: Invalid Details	Invalid	Flight Number is Invalid	Flight Number entered is Invalid
TC <sub>3</sub>	Scenario 3- User Exits	Valid/ Invalid	User comes out of the system	-

---

## 6.5. VIEW BOOKED TICKET (Condition: Tickets are booked)

Test case ID	Scenario Name and Description	INPUT 1 Enter Ticket Number	Expected Output	Remarks
TC <sub>1</sub>	Scenario 1- View Booked Ticket	Valid	Booked Ticket Details are shown	-
TC <sub>2</sub>	Scenario 2- View Booked Ticket- Alternate Flow: Invalid Details	Invalid	Ticket number is Invalid	Ticket Number entered is Invalid
TC <sub>3</sub>	Scenario 3- View Booked Ticket- Alternate Flow: User Exits	Valid/ Invalid	User comes out of the system	-

---

**6.6. CANCEL BOOKED TICKET (Condition: Tickets are booked and Refund condition is satisfied)**

<b>Test Case ID</b>	<b>Scenario Name and Description</b>	<b>INPUT 1 Ticket Number</b>	<b>INPUT 2 User ID</b>	<b>Expected Output</b>	<b>Remarks</b>
TC <sub>1</sub>	Scenario 1- Cancel Booked Ticket	Valid	Valid	Tickets Cancelled	-
TC <sub>2</sub>	Scenario 2- Cancel Book Ticket Alternate Flow: Invalid Details	Invalid/ Blank	Valid/ Invalid	Ticket Number Invalid	Ticket Number entered is Invalid
TC <sub>3</sub>		Valid	Invalid/ Blank	User ID Invalid	User ID entered is Invalid
TC <sub>4</sub>	Scenario 3- Cancel Booked Ticket Alternate Flow: User Exits	Valid/ Invalid	Valid/ Invalid	User comes out of the System	-