CPS-5301 Final Project

Software Requirement Specification

Going Anywhere APP - Group 3

Cheng Bao(1335784), Xinin Qiu(1335785), Dinggen Zhang(1336574)

# Introduction

The software provides a platform for users who want to travel by air and want to find all possible choices they have in one place. The software will gather flight information from all avaliable airlines, give users a list of tickets that they can buy according to their search settings, and complete the processes of purchasing, changing, refunding and boarding of their tickets.

## Project Information

* Project Name: Going Anywhere APP
* Category: Airline Ticket Management

## Group Information

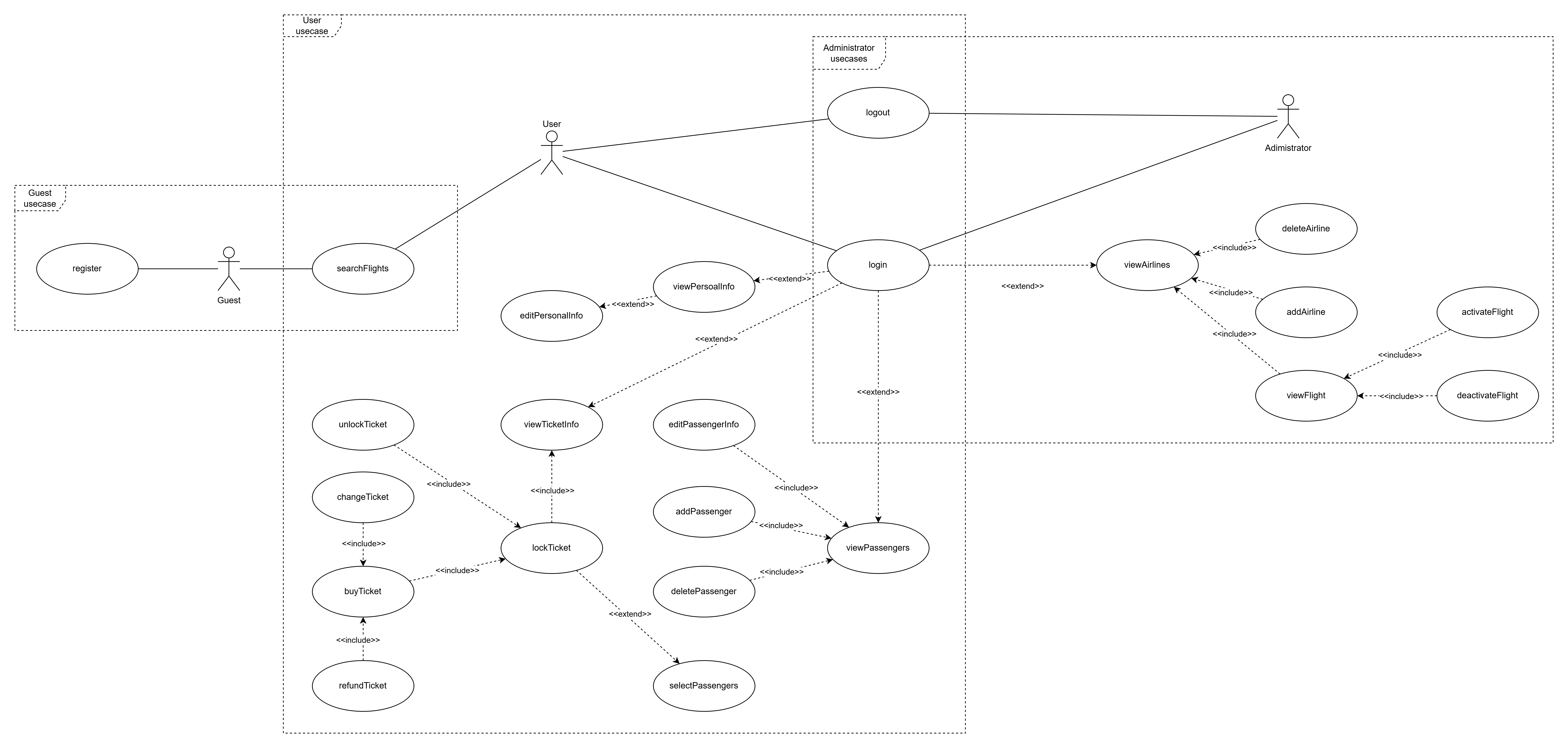
* Group Name: Old Folks
* Group Member:
  + Cheng Bao, as the Product Manager
  + Xinxin Qiu, as the Development Manager
  + Dinggen Zhang, as the Architector & Testing Manager

# Actors & Usecases

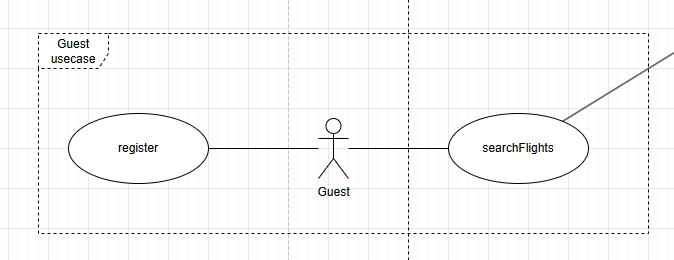
|  |  |
| --- | --- |
| Actors | Usecases |
| Guest | searchFlights  register |
| User | login  logout  viewPersonalInfo  editPersonalInfo  viewPassengers  addPassenger  editPassengerInfo  deletePassenger  searchFlights  viewTicketInfo  lockTicket  unlockTicket  selectPassenger  buyTicket  refundTicket  changeTicket |
| Administrator | login  logout  addAirline  viewAirlines  viewFlight  activateFlight  deactivateFlight  deleteAirline |

# Usecase Diagram

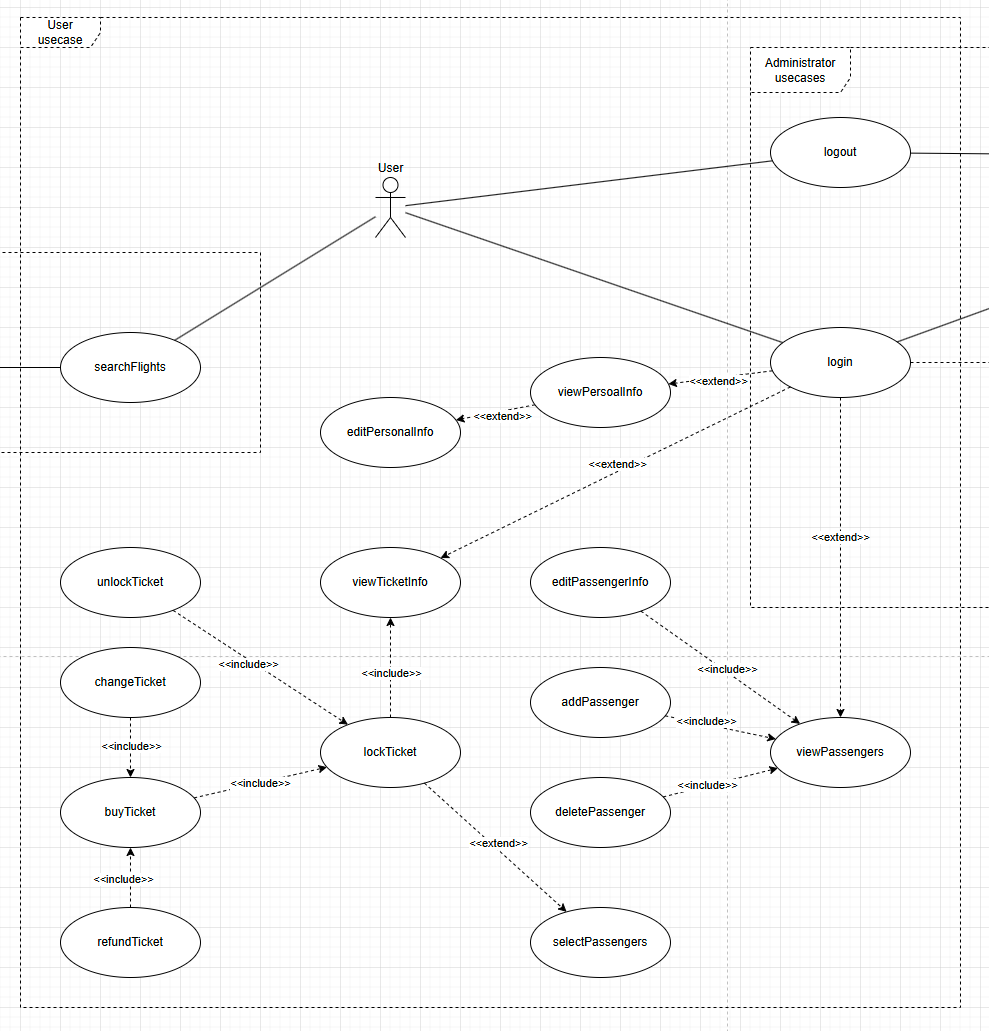
The overview



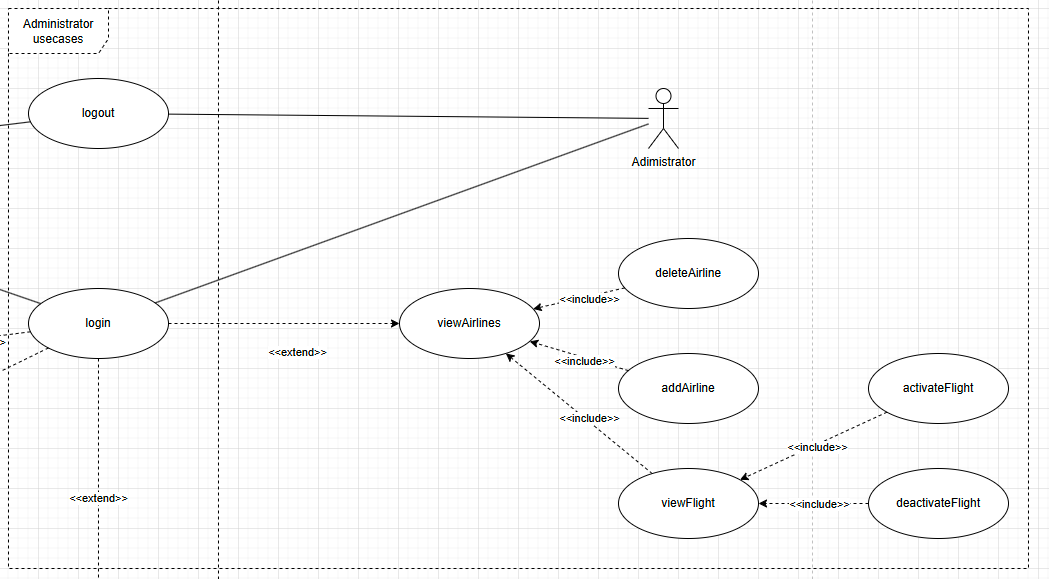
The guest part



The user part



The administrator part



# User Requirements

The software shall have following requirements combined with serveral modules:

## Account Management

1.1. The software shall allow a guest to register as user.

1.2. The software shall allow a user or administrator to login.

1.3. The software shall allow a user of administrator to logout.

1.4. The software shall allow a user to view his/her personal information.

1.5. The software shall allow a user to edit his/her personal information.

1.6. The software shall allow a user to view his/her passenger list.

1.7. The software shall allow a user to add a passenger into his/her own passenger list.

1.8. The software shall allow a user to delete a passenger from his/her own passenger list.

1.9. The software shall allow a user to edit the information of passenger in his/her own passenger list.

## Ticket Management

2.1. The software shall allow a guest or user to search avaliable flights based on his/her searching query.

2.2. The software shall allow a user to view the information of the chosen flight.

2.3. The software shall allow a user to view avaliable tickets from chosen filght.

2.4. The software shall allow a user to lock the selected ticket.

2.5. The software shall allow a user to unlock the locked ticket.

2.5. The software shall allow a user to select passenger for the locked ticket.

2.6. The software shall allow a user to buy the locked ticket.

2.7. The software shall allow a user to change the ticket he/she bought, the change time of each ticket shall limit to one, the changed ticket shall not be changed again.

2.8. The software shall allow a user to refund the ticket he/she bought, once confirm, the refund shall not be undo.

2.9. The software shall allow a user to view the process status of the selected ticket.

## Airline/Flight Management

3.1. The software shall allow an administrator to view the list of connected airlines.

3.2. The software shall allow an administrator to add an airline into the system, and the system shall gather the information of flight and tickets from this airline. All the flight gathered from airline shall inital as deactived firstly.

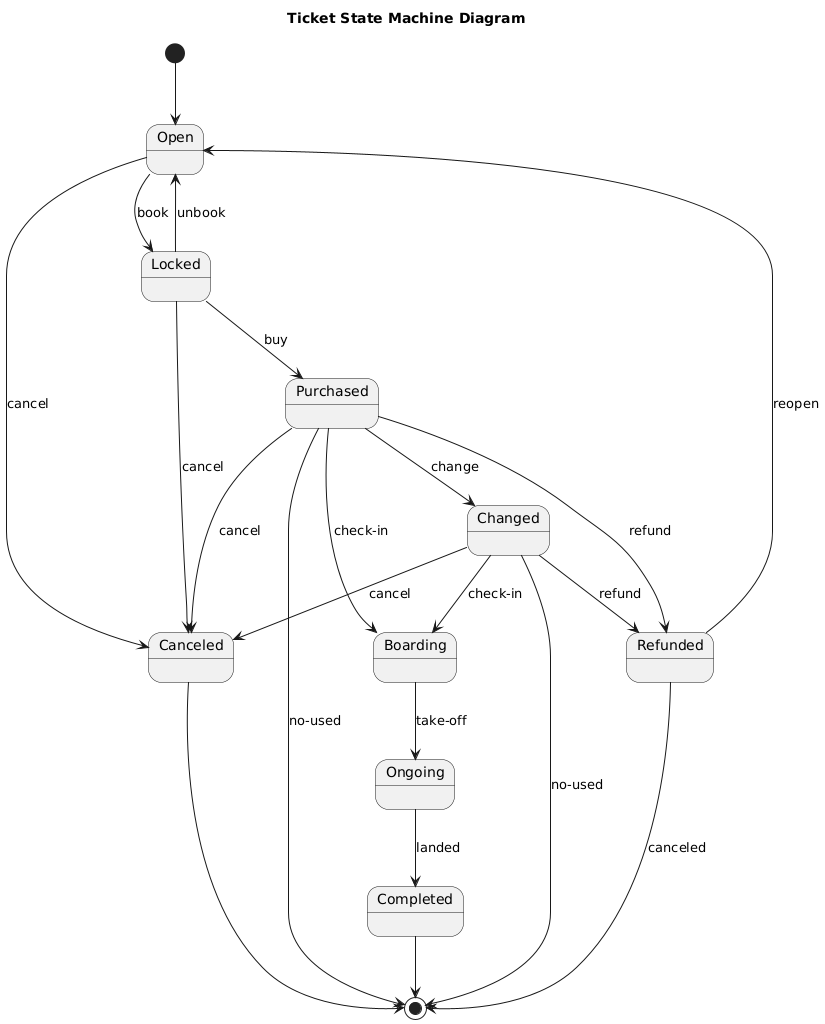
3.3. The software shall allow an administrator to view the information of all the flights of the selected airline, as long as the detail of the selected flight.

3.4. The software shall allow an administrator to activate the selected flight, all tickets related this flight will be open.

3.5. The software shall allow an administrator to deactivate the selected flight, all tickets related this flight will be canceled.

3.6. The software shall allow an administrator to delete an airline, an airline is able to be deleted only if all the flights of this airline are deactivated.

# Ticket State Machine Diagram



# System Requirements

## 1. Account Management Module

### 1.1 Account Registration

* The system shall provide a registration interface for guests to provide necessary information (e.g., username, password, email, etc.).
* The system shall validate the provided information and ensure that no duplicate usernames or emails exist in the database.
* Upon successful validation, the system shall store the user information in the database and generate a unique user ID.

### 1.2 User Login and Logout

* The system shall provide a user login portal where users can input their credentials (username and password).
* The system shall authenticate the user by comparing the input credentials with the stored data in the database.
* Upon successful login, the system shall maintain a session for the user.
* The system shall provide a logout functionality that terminates the user session.

### 1.3 Administrator Login and Logout

* The system shall provide a separate admin login portal where administrators can input their credentials.
* The system shall authenticate the administrator by comparing the input credentials with the stored data in the database.
* Upon successful login, the system shall maintain a session for the administrator.
* The system shall provide a logout functionality that terminates the administrator session.

### 1.4 View Personal Information

* The system shall allow logged-in users to view their personal information, including but not limited to username, email, and registered details.
* The system shall fetch the user’s personal information from the database and display it in a readable format.

### 1.5 Edit Personal Information

* The system shall allow logged-in users to edit their personal information.
* The system shall validate the edited information and update the database accordingly.
* The system shall notify the user of the success or failure of the update operation.

### 1.6 View Passenger List

* The system shall allow logged-in users to view their passenger list, which includes details of all associated passengers.
* The system shall fetch the passenger list from the database and display it in a readable format.

### 1.7 Add Passenger

* The system shall allow logged-in users to add a new passenger to their passenger list.
* The system shall validate the passenger information and store it in the database.
* The system shall notify the user of the success or failure of the addition operation.

### 1.8 Delete Passenger

* The system shall allow logged-in users to delete a passenger from their passenger list.
* The system shall remove the passenger’s information from the database.
* The system shall notify the user of the success or failure of the deletion operation.

### 1.9 Edit Passenger Information

* The system shall allow logged-in users to edit the information of a passenger in their passenger list.
* The system shall validate the edited information and update the database accordingly.
* The system shall notify the user of the success or failure of the update operation.

## 2. Ticket Management Module

### 2.1 Guest Flight Search

* The system shall provide a search interface where guests can input search criteria (e.g., departure city, arrival city, date, etc.) to search for available flights.
* The system shall display basic information about the available flights, but detailed information will be restricted until the guest registers and logs in.

### 2.2 User Flight Search

* The system shall provide a search interface where logged-in users can input search criteria to search for available flights.
* The system shall display detailed information about the available flights, including all relevant details such as flight times, prices, and seat availability.

### 2.3 View Available Tickets

* The system shall allow logged-in users to view available tickets for a selected flight.
* The system shall display detailed information about each ticket, including price, seat class, and any associated restrictions.

### 2.4 Lock and Unlock Ticket

* The system shall allow logged-in users to lock a selected ticket to prevent others from purchasing it.
* The system shall allow the user to unlock the ticket, making it available for purchase by others again.
* The system shall maintain the lock status in the database.

### 2.5 Select Passenger for Ticket

* The system shall allow logged-in users to select a passenger from their passenger list for the locked ticket.
* The system shall validate the selection and update the ticket’s passenger information in the database.

### 2.6 Purchase Ticket

* The system shall allow logged-in users to purchase the locked ticket.
* The system shall process the payment and update the ticket status to “purchased” in the database.
* The system shall notify the user of the success or failure of the purchase operation.

### 2.7 Change Ticket

* The system shall allow logged-in users to change the ticket they have purchased, with the restriction that each ticket can only be changed once.
* The system shall update the ticket status to “changed” and mark it as non-changeable in the database.
* The system shall notify the user of the success or failure of the change operation.

### 2.8 Refund Ticket

* The system shall allow logged-in users to refund the ticket they have purchased.
* The system shall process the refund and update the ticket status to “refunded” in the database.
* The system shall notify the user of the success or failure of the refund operation.

### 2.9 View Ticket Information and Status

* The system shall allow logged-in users to view the detailed information and status of a selected ticket.
* The system shall display the ticket status, which includes the following options, the state machine diagram show above:

1. Open: Ticket is open for sale or unlocked.
2. Locked: Ticket is locked by a user but not purchased yet.
3. Purchased: Ticket is already bought.
4. Canceled: Ticket is canceled by airline or administrator.
5. Changed: Ticket is changed by user.
6. Refunded: Ticket is refunded by user.
7. Boarding: Ticket is checked in by user.
8. On-going: Flight of the ticket is already take-off.
9. Completed: Flight of the ticket is landed, and the ticket is completed.

## 3. Airline/Flight Management Module

### 3.1 View Connected Airlines

* The system shall allow administrators to view the list of airlines that are connected to the system.
* The system shall fetch the list of connected airlines from the database and display it in a readable format.

### 3.2 Add Airline

* The system shall allow administrators to add a new airline to the system.
* The system shall gather flight and ticket information from the new airline and store it in the database.
* All flights gathered from the new airline shall be initially set to “deactivated” status.

### 3.3 View Airline Flights

* The system shall allow administrators to view all flights associated with a selected airline.
* The system shall display detailed information about each flight, including flight times, routes, and status.
* The status of a flight shall include the following options:
  + Deactivated: Flight is not open for users.
  + Activated: Flight is open for users.

### 3.4 Activate Flight

* The system shall allow administrators to activate a selected flight of the selected airline.
* The system shall update the flight status to “activated” and set all related tickets to “open” status in the database.

### 3.5 Deactivate Flight

* The system shall allow administrators to deactivate a selected flight of the selected airline.
* The system shall update the flight status to “deactivated” and set all related tickets to “canceled” status in the database.

### 3.6 Delete Airline

* The system shall allow administrators to delete an airline from the system, provided that all flights associated with the airline are deactivated.
* The system shall remove the airline and all associated flights and tickets from the database.
* The system shall notify the administrator of the success or failure of the deletion operation.

# Methods inputs, outputs & passing parameters

See the detail of the Class Diagram

# Class Diagram

