CS 102 Project

ÖZYEĞİN ÜNİVERSİTESİ

Project submission Date: 27 May 2023 (Time: 12 Midnight) via LMS

Project Demo Date: 31 May 2023

Airport Management Systems

Background

An airport management system (AMS) is a software solution used to manage and optimize airport operations, including flight operations, passenger processing, baggage handling, and other airport services. An AMS typically integrates multiple modules to provide a comprehensive solution for airport management. Some of the key modules in an AMS include but are not limited to:

- **Flight Management**: This module manages flight schedules, tracks flight movements, and provides real-time information on flight status. It also facilitates communication between airlines, ground handlers, and air traffic control.
- **Passenger Management**: This module manages passenger processing, including check-in, security screening, boarding, and baggage handling. It provides real-time information on passenger flow and facilitates communication with passengers.
- **Plane Management**: This module manages all the operations related to airplanes and resources related to plane such as runways allocation, hanger where plane need to be parked and other multiple things. It helps optimize the use of these resources and tracks their availability and usage.
- **Baggage Management**: This module tracks baggage from check-in to arrival, ensuring that it is properly routed and delivered to the correct destination. It also manages baggage handling equipment and provides real-time information on baggage status.

Overall, an AMS helps optimize airport operations and improve the passenger experience by providing real-time information and facilitating communication between stakeholders. It is an essential tool for modern airports and helps ensure safe, efficient, and profitable airport operations.

In this project we are going to implement AMS and its following key modules as shown in fig 1.

- Passenger Management System
- Plane Management System
- Flight Management System
- Tickets

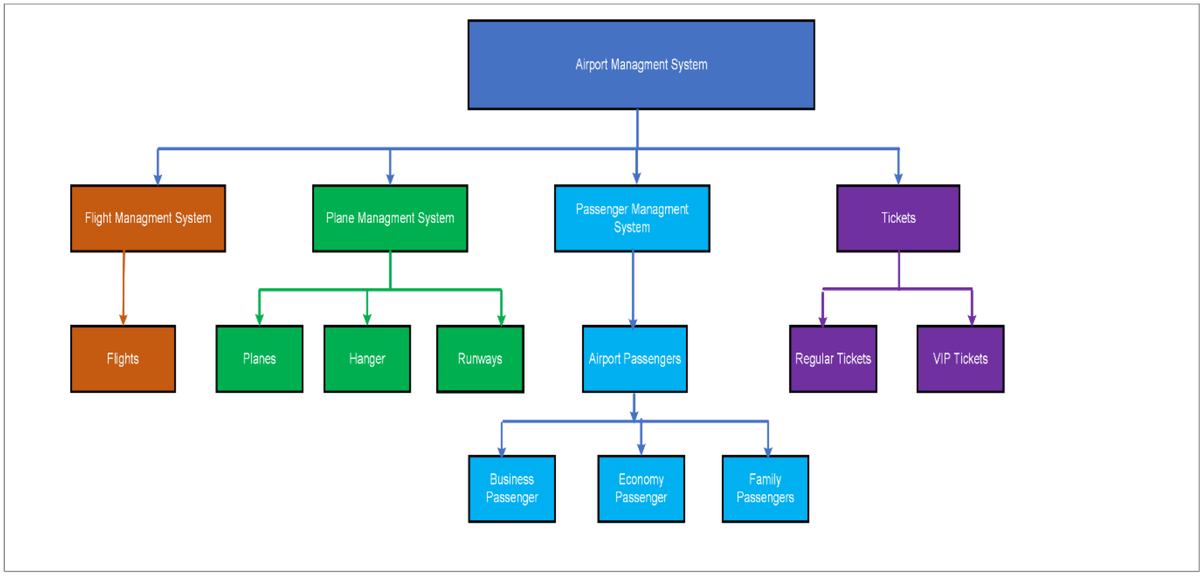


Figure 1: Airport Management System

Flight Management System contain the following functions/modules

- Generate Flights
- Add flights details
- Remove flights
- Current flights
- Manage Tickets
- Buy ticket and cancel ticket for a particular flight

Plane Management System contain the following functions/modules

- Add plane
- Remove plane
- Add and remove hanger

- Add and remove runways
- Reserve and reset hanger
- Reserve and reset runways

Passenger Management System contain the following functions/modules

- Add passenger details (name, sur name)
- Passenger type (Business, Economy or family)
- Add luggage details
- Add date of birth

An Ticket class need to be created which contain the following details:-

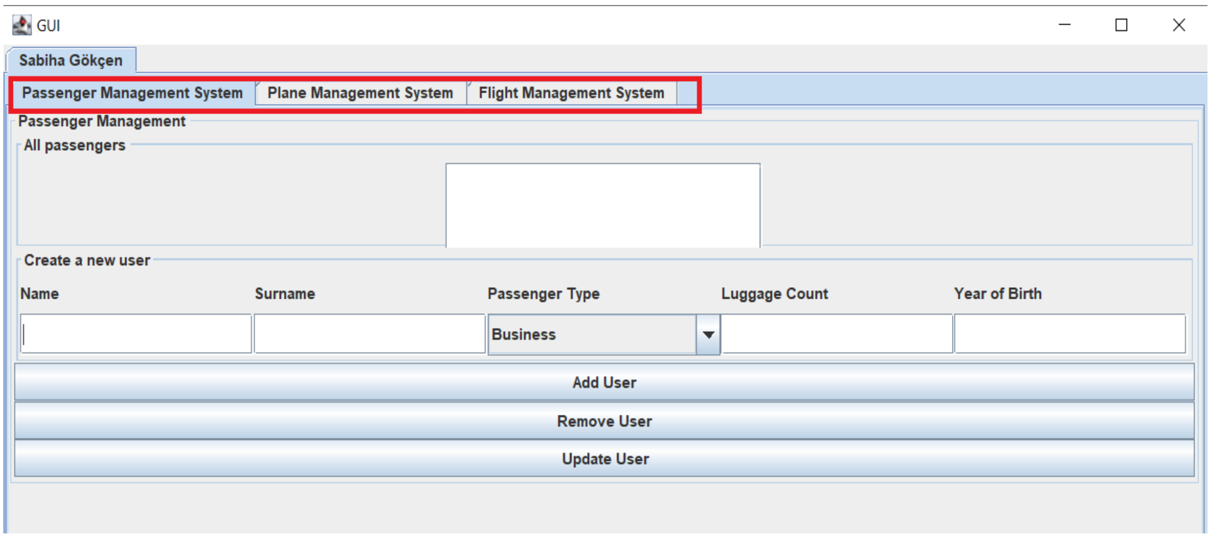
- Flight details
- Passenger details
- Seat allocation
- Base price calculation
- Regular ticket and VIP ticket
- VIP tickets can only be purchased by business class. Please add some discount ratio as well for different categories of class.

1. Students are expected to deploy all the concepts in this project that are learned in the class i-e abstract class, interface concept, inheritance and polymorphism concept, exception handling techniques.
2. Students must utilize the abstract and inheritance feature as the project contains multiple subclasses. It should be implemented properly so that it can inherit properties and methods from another subclass (superclass or base class)
3. Students must utilize the interface feature as well. Since multiple classes can implement the same interface, so that your code can work with any object that implements that interface.
4. Students must utilize exception handling for invalid data entries. As an example, few screen shots are shown in the figures as well that pop up a window when an invalid entry is added in their respective filed. You can think of multiple scenarios by yourself for the whole project.
5. UI must include **a MVC pattern** GUI and listeners.
6. Students must utilize array list for storing any kind of information.
7. Demo session is mandatory to be attended (31 May 2023), otherwise it will be

graded 0 from the project (It does not matter if you submit your project or not)

Step 1: Create a GUI (MVC based pattern) which contains the: -

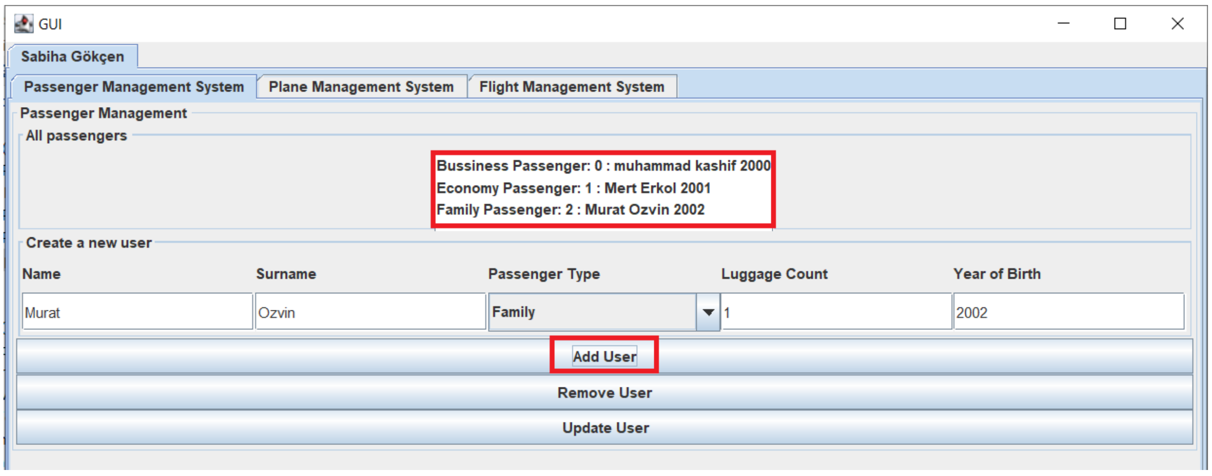
- Passenger management system
- Plane management system
- Flight management system



Step 2: For the Passenger management system, the following functionalities need to be achieved in this gui pattern.

- User can be able to **add** passenger details by entering his/her first name, last name, date of birth and luggage details.
- User can be able to select **Passenger type** i-e Business class, Economy class and Family class.
- User can be able to delete or **remove** passengers info
- User can be able to **update** passenger details as well

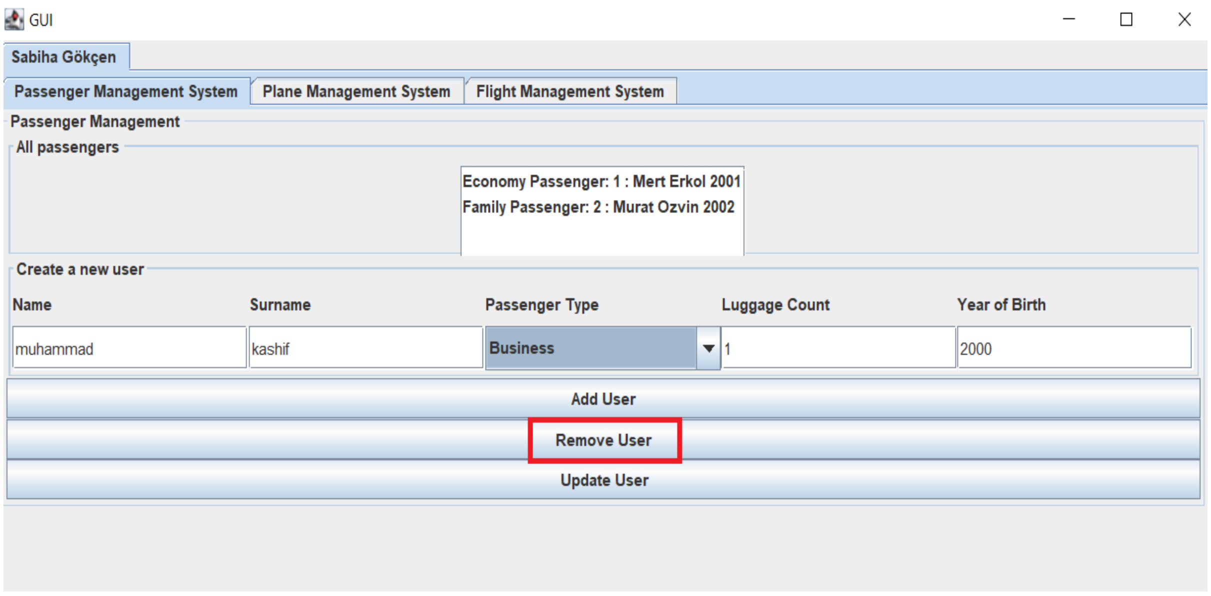
For example, to add the passenger details, we need to enter Name, Surname, passenger type, luggage count and date of birth



Bussiness Passenger: 0 : muhammad kashif 2000
Economy Passenger: 1 : Mert Erkol 2001
Family Passenger: 2 : Murat Ozvin 2002

Name	Surname	Passenger Type	Luggage Count	Year of Birth
Murat	Ozvin	Family	1	2002

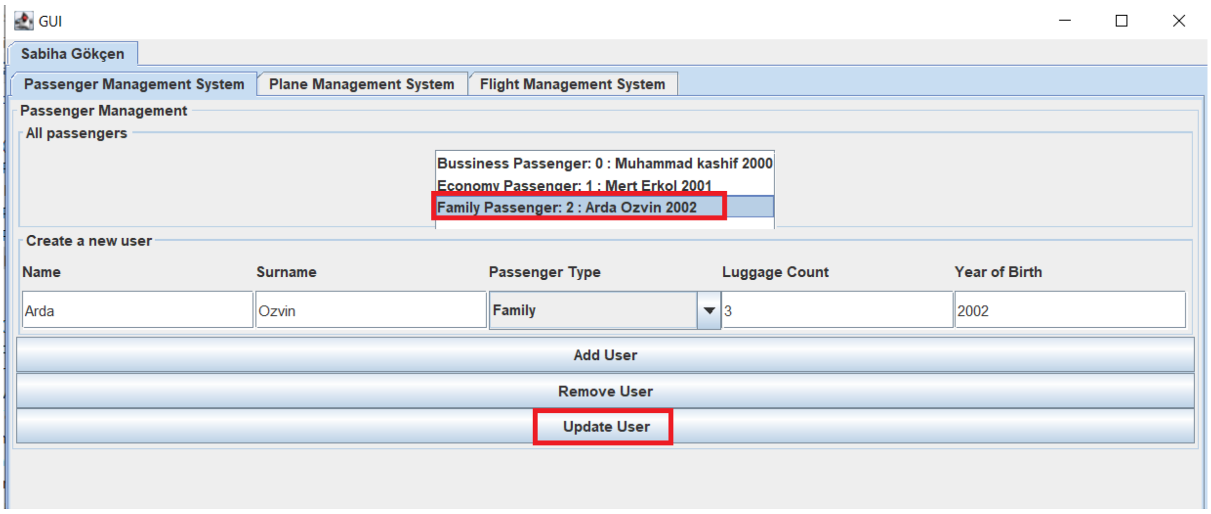
To remove a particular user, we can remove the passenger by specifying the attribute and that particular passenger will be removed.



Economy Passenger: 1 : Mert Erkol 2001
Family Passenger: 2 : Murat Ozvin 2002

Name	Surname	Passenger Type	Luggage Count	Year of Birth
muhammad	kashif	Business	1	2000

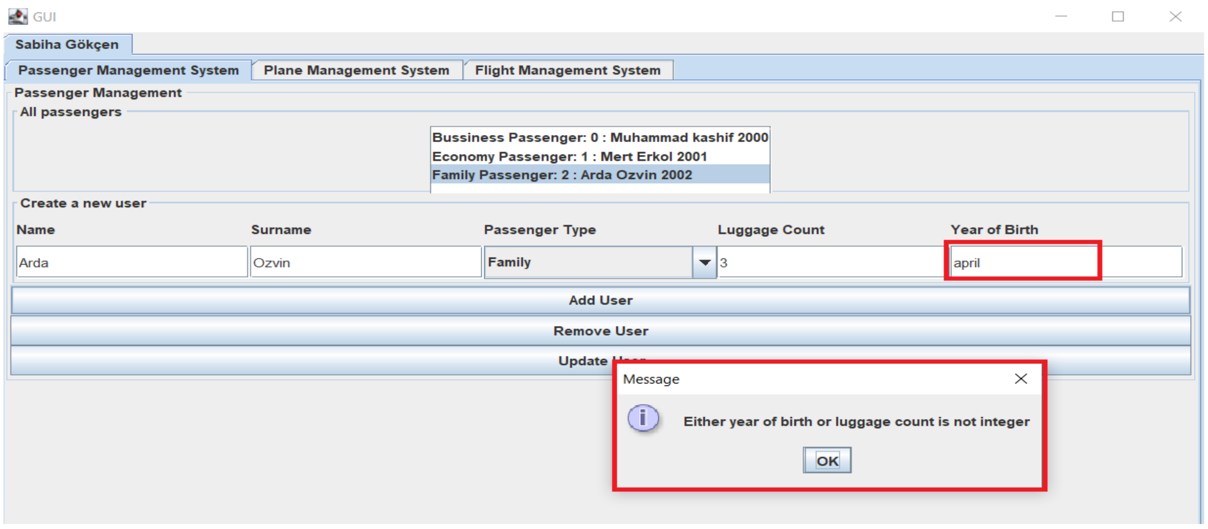
To update a user i-e to change the information of passenger.



Bussiness Passenger: 0 : Muhammad kashif 2000
Economy Passenger: 1 : Mert Erkol 2001
Family Passenger: 2 : Arda Ozvin 2002

Name	Surname	Passenger Type	Luggage Count	Year of Birth
Arda	Ozvin	Family	3	2002

If any invalid entry is entered it should pop a window error message (Exception error handling)



Bussiness Passenger: 0 : Muhammad kashif 2000
Economy Passenger: 1 : Mert Erkol 2001
Family Passenger: 2 : Arda Ozvin 2002

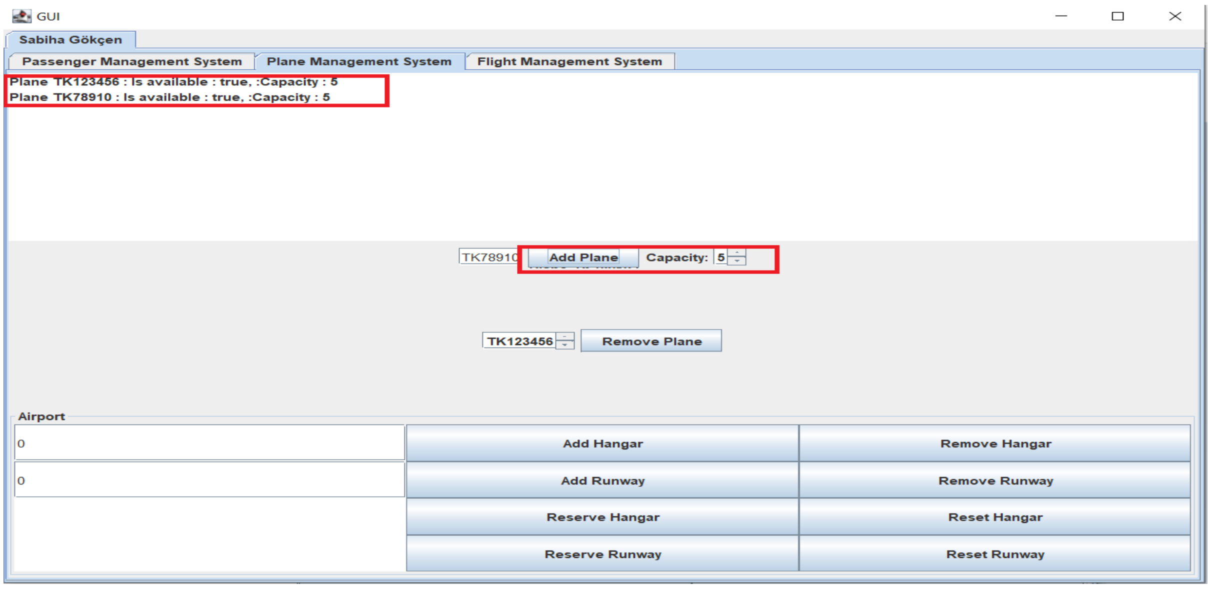
Name	Surname	Passenger Type	Luggage Count	Year of Birth
Arda	Ozvin	Family	3	april

Either year of birth or luggage count is not integer

Step 3: For the Plane management system, the following functionalities need to be achieved in this gui pattern.

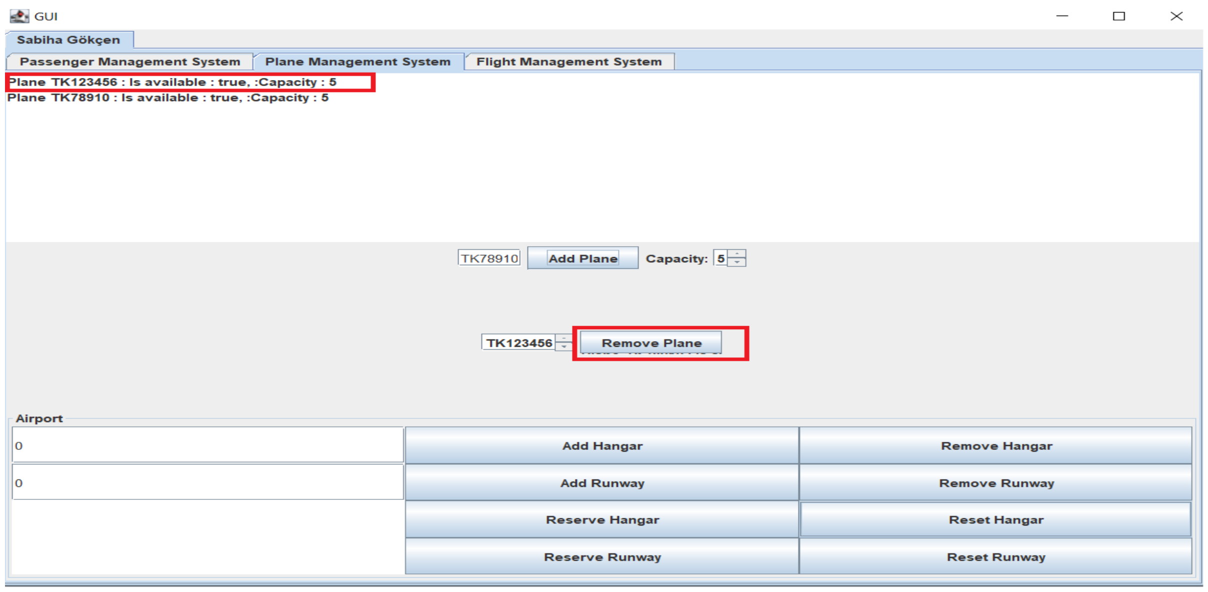
- User can be able to **add** plane details
- User can be able to **delete** or **remove** plane details
- User can be able to **add and remove** hangers details
- User can be able to **reserve and reset** hangers for any plane added
- User can be able to **add and remove** runways details
- User can be able to **reserve and reset** runways for any plane added

For example, to add the plane, we need to enter its plane details



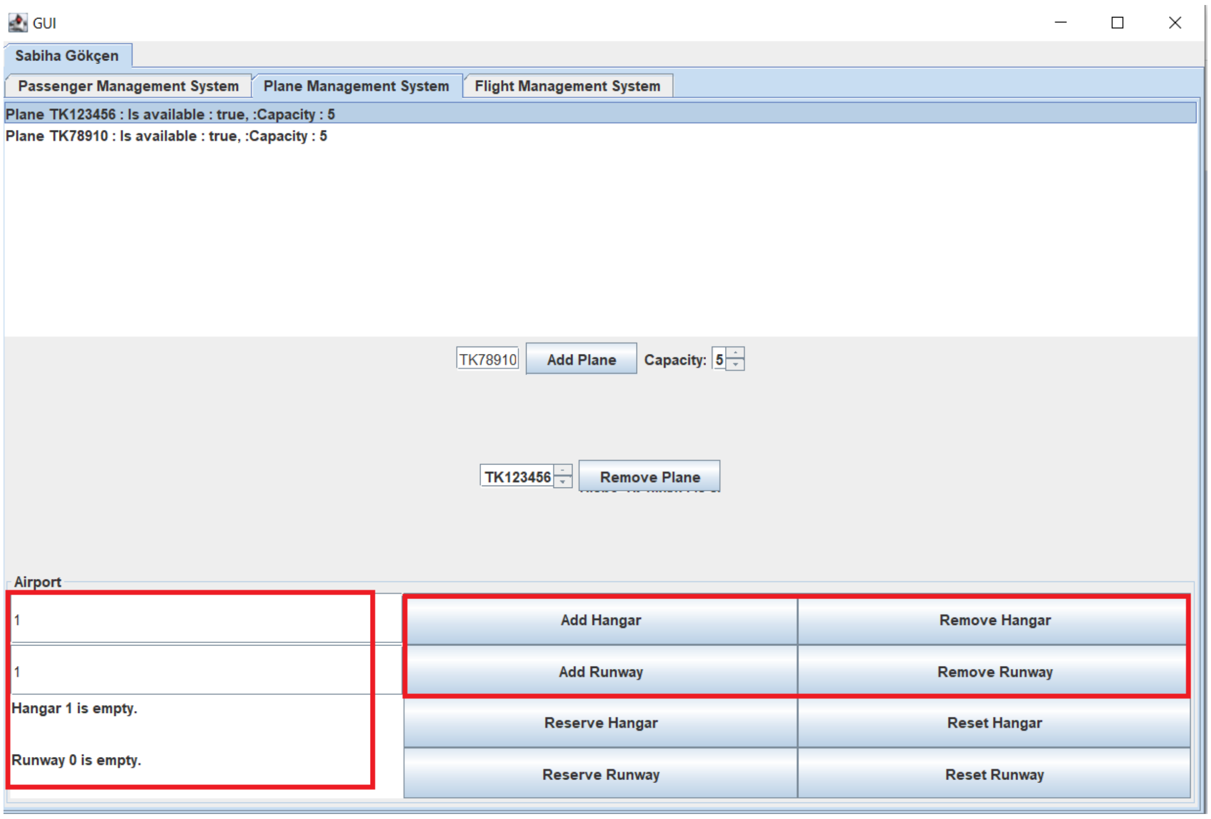
```
Plane TK123456 : Is available : true, :Capacity : 5
Plane TK78910 : Is available : true, :Capacity : 5
```

A particular plane details can be deleted as well



```
Plane TK123456 : Is available : true, :Capacity : 5
Plane TK78910 : Is available : true, :Capacity : 5
```

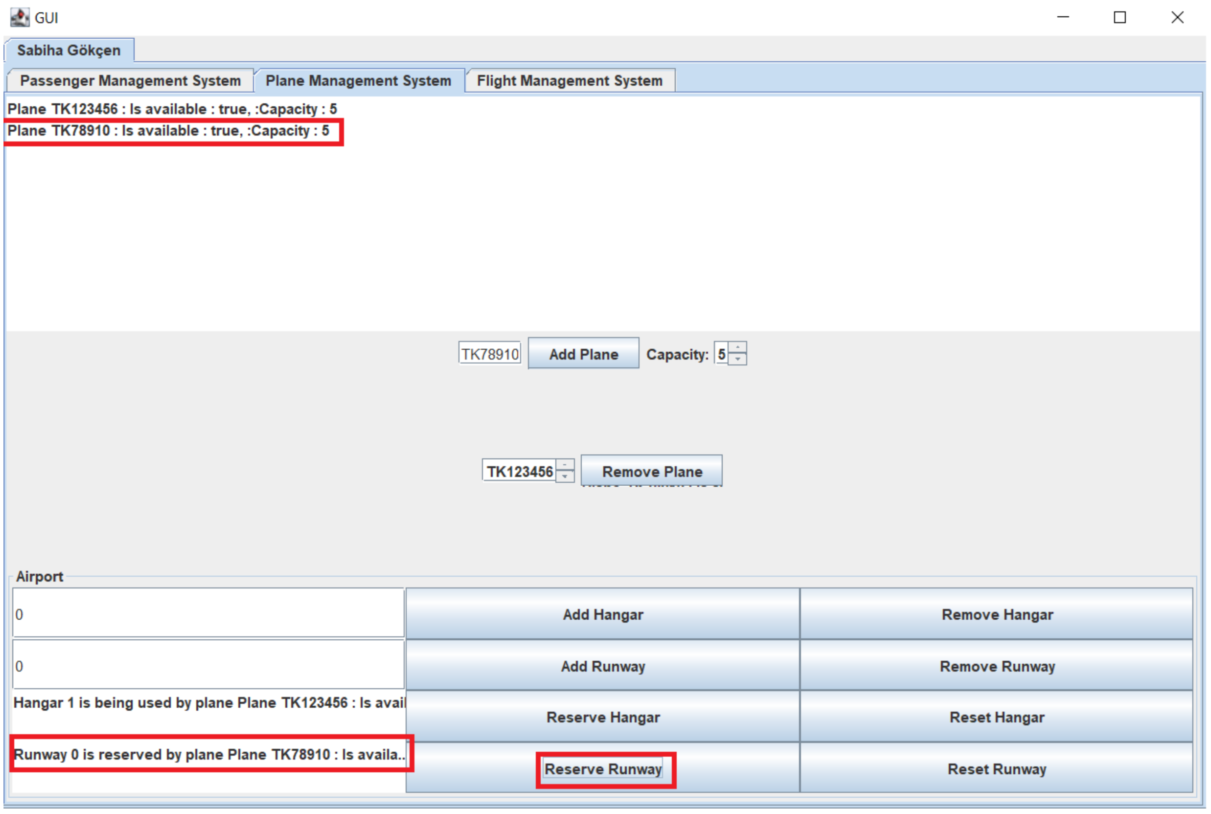
Similarly to add the hanger and runway we need to mention its airport id and we can remove the hanger and runway as well



```
Plane TK123456 : Is available : true, :Capacity : 5
Plane TK78910 : Is available : true, :Capacity : 5
```

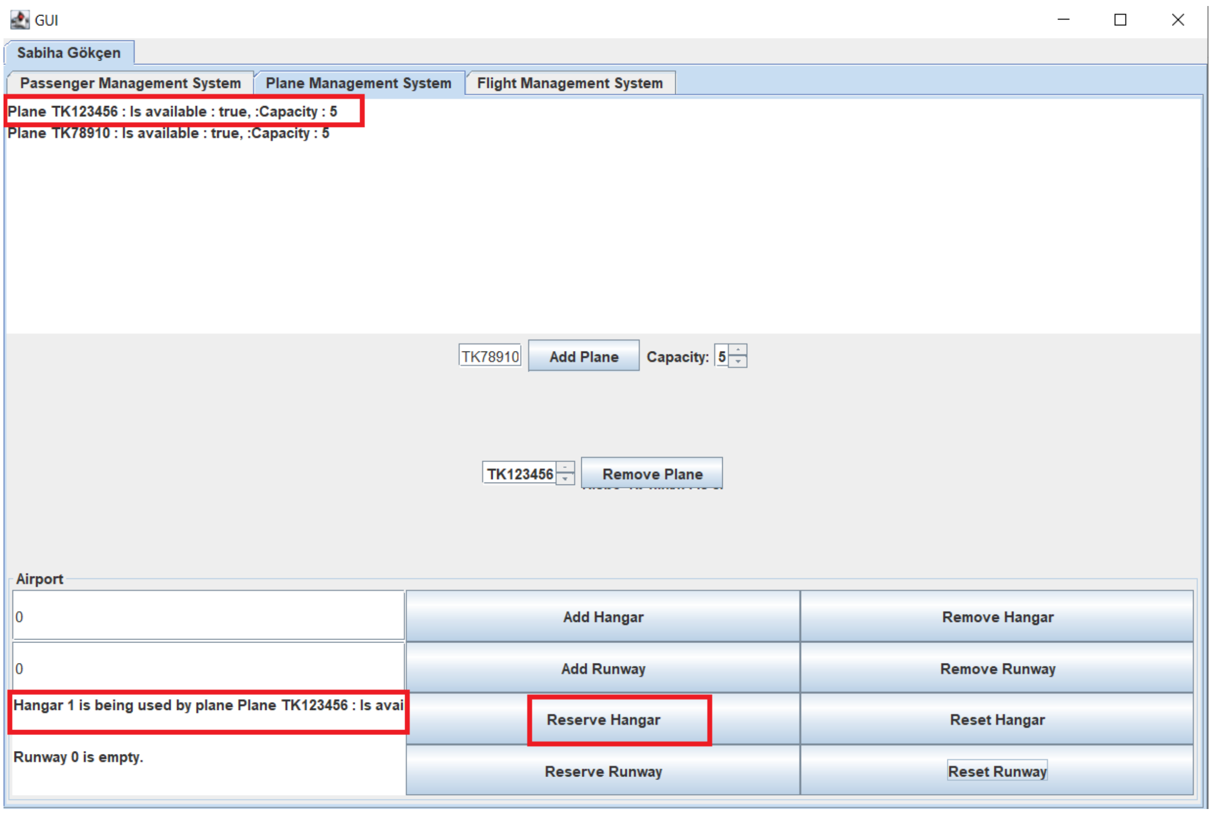
```
1
1
Hangar 1 is empty.
Runway 0 is empty.
```

We can reserve the hanger and runway for the particular plane that is added previously



```
Plane TK123456 : Is available : true, :Capacity : 5
Plane TK78910 : Is available : true, :Capacity : 5
```

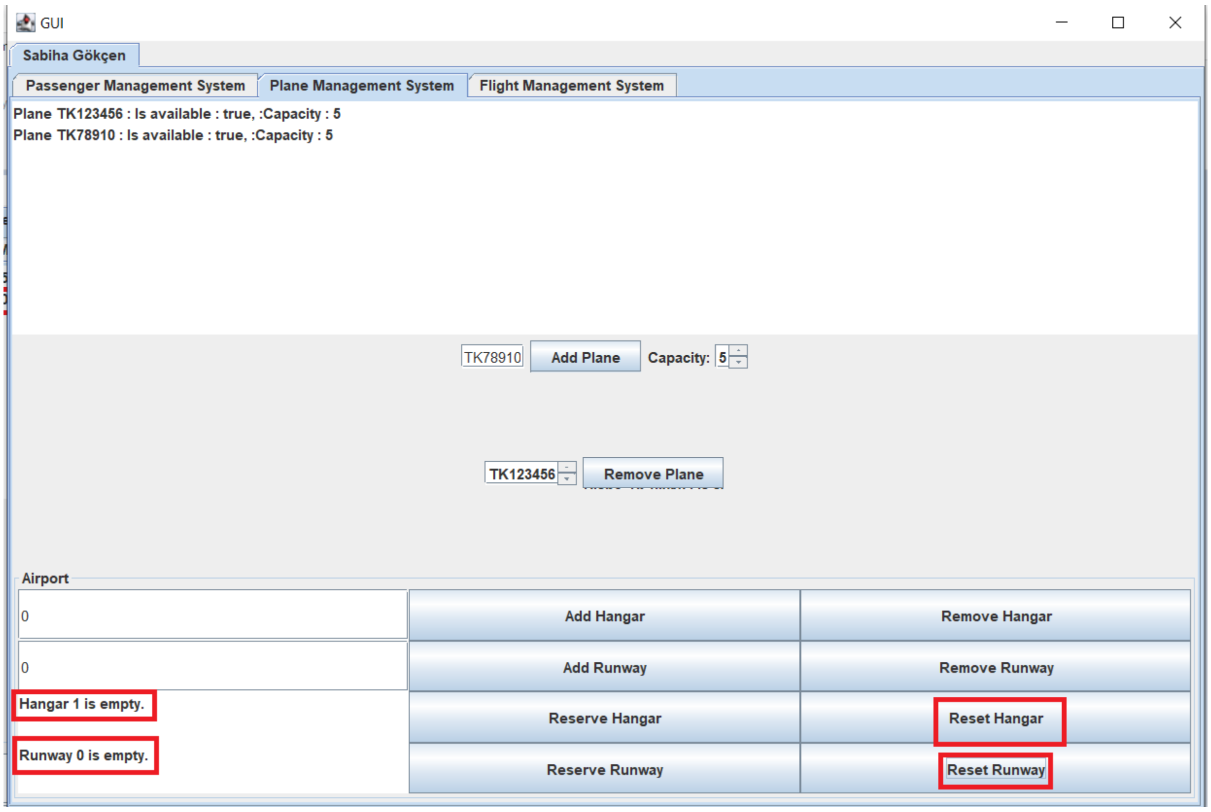
```
0
0
Hangar 1 is being used by plane Plane TK123456 : Is avail
Runway 0 is reserved by plane Plane TK78910 : Is availa...
```



Plane TK123456 : Is available : true, :Capacity : 5
Plane TK78910 : Is available : true, :Capacity : 5

Hangar 1 is being used by plane Plane TK123456 : Is avai
Runway 0 is empty.

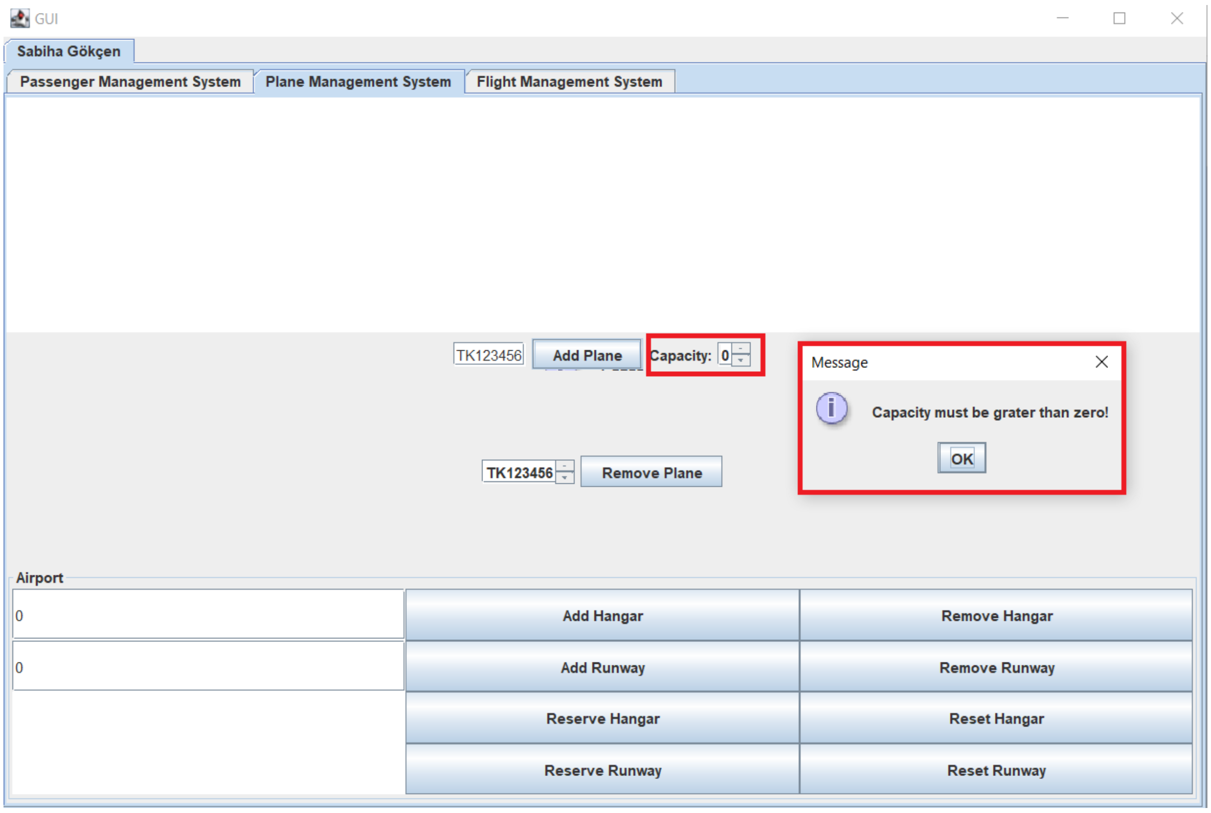
Similarly we can reset the hanger and runway if it is not being used by any plane



Plane TK123456 : Is available : true, :Capacity : 5
Plane TK78910 : Is available : true, :Capacity : 5

Hangar 1 is empty.
Runway 0 is empty.

For an invalid entries , error must be pop up as well

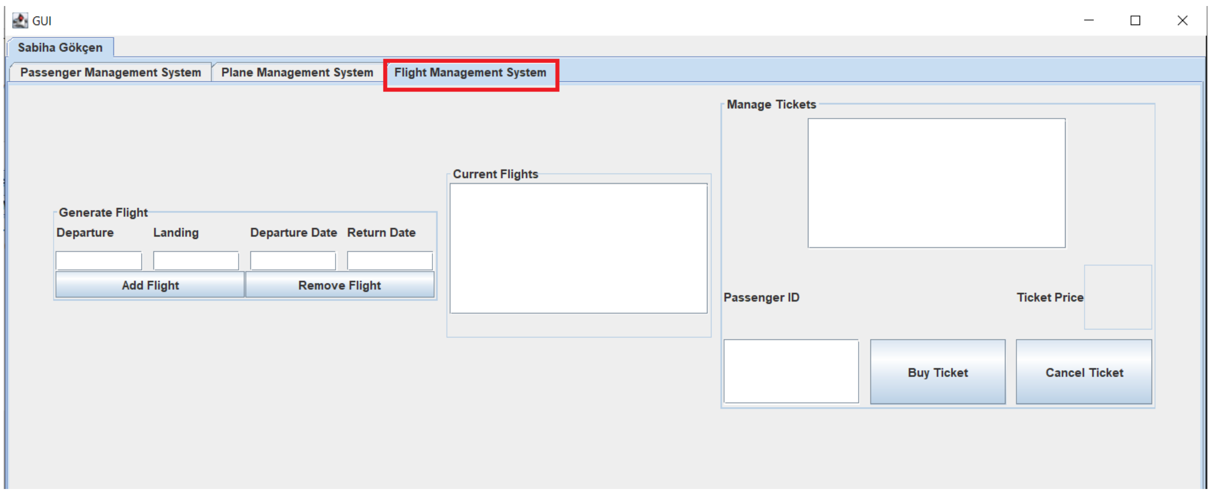


Message

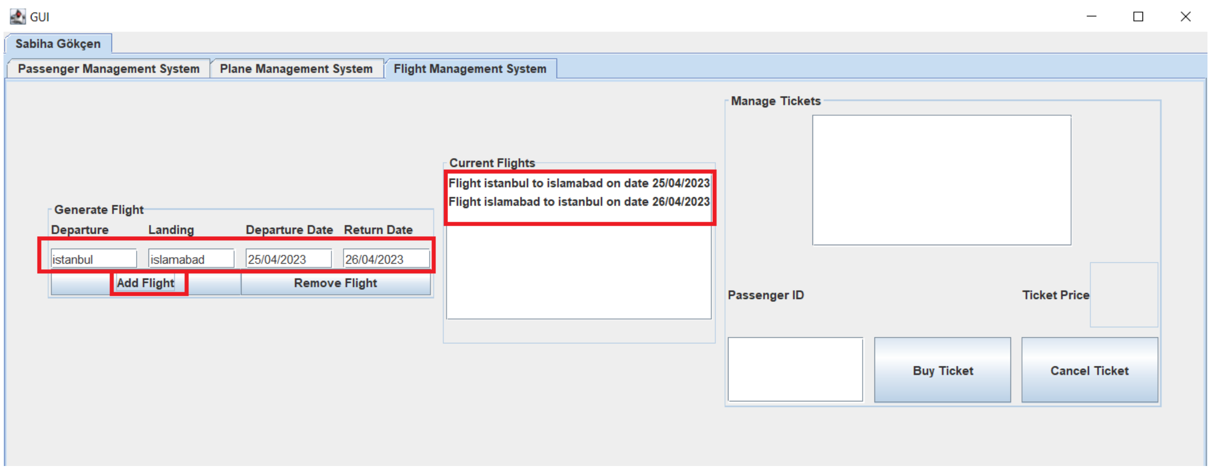
Capacity must be grater than zero!

Step 4: For the Flight management system, the following functionalities need to be achieved in this gui pattern.

- User can be able to **generate** flight details by entering departure and landing location information, departure and return date, and capacity information of the plane.
- User can be able to **add or remove** flights.
- User can be able to manage tickets by i-e buying a ticket for a particular passenger that is created earlier. Users can allocate a seat number as well.
- User can cancel a ticket for particular passenger as well.



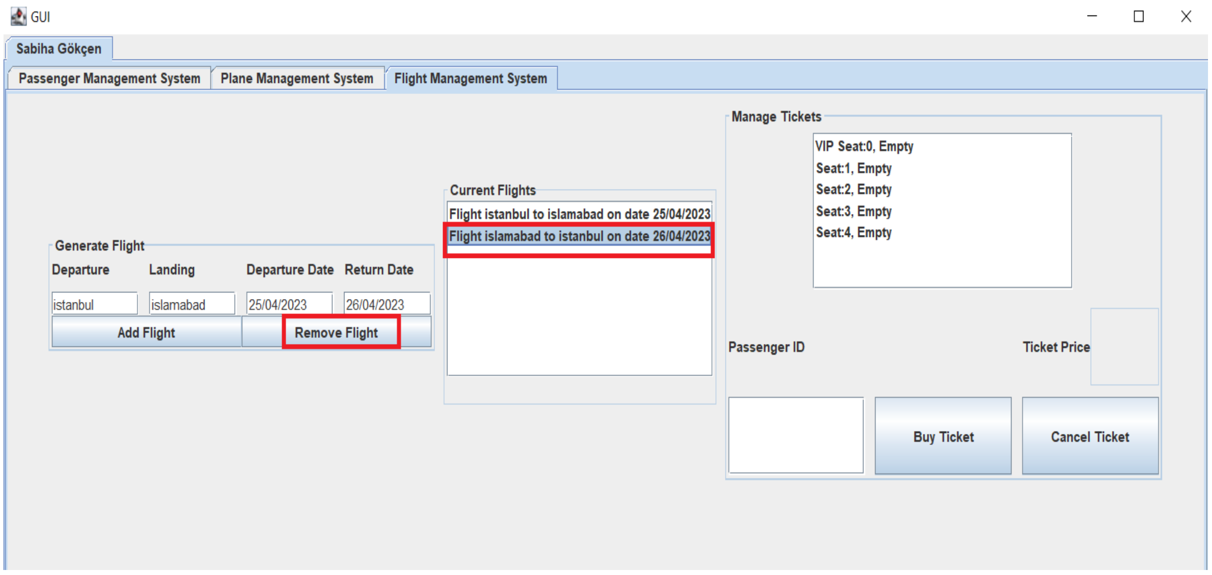
For example, to add a flight, we need to enter its departure and landing location, departure and return date



Departure: istanbul | Landing: islamabad | Departure Date: 25/04/2023 | Return Date: 26/04/2023

Flight istanbul to islamabad on date 25/04/2023
Flight islamabad to istanbul on date 26/04/2023

A flight can be removed as well by selecting a particular flight that is added earlier



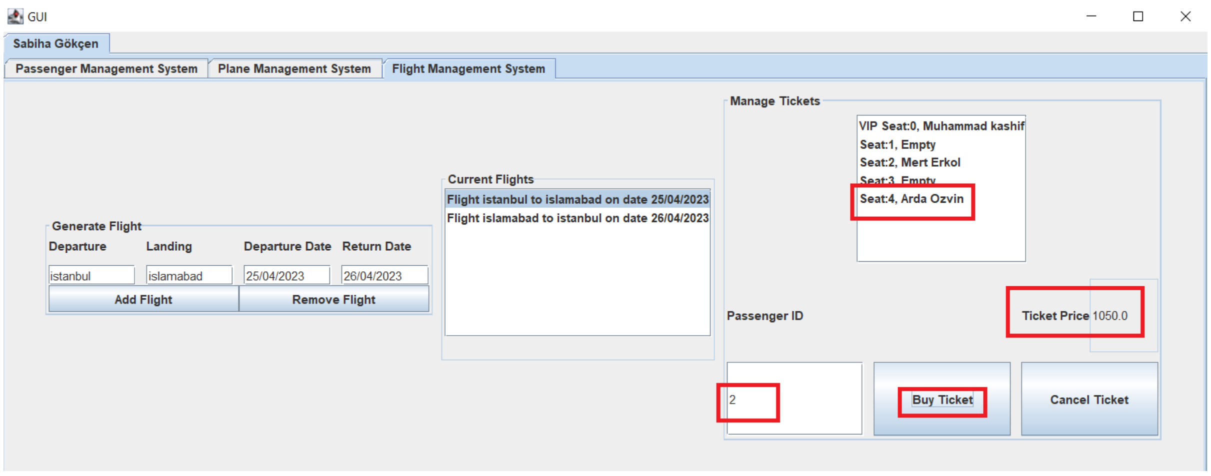
Departure: istanbul | Landing: islamabad | Departure Date: 25/04/2023 | Return Date: 26/04/2023

Flight istanbul to islamabad on date 25/04/2023
Flight islamabad to istanbul on date 26/04/2023

VIP Seat:0, Empty
Seat:1, Empty
Seat:2, Empty
Seat:3, Empty
Seat:4, Empty

Ticket can be purchased or cancelled for a particular passenger that is created earlier by mentioning his/her id , and seat number and seat number allocated to him. Ticket price will vary depending upon the nature of ticket that is booked at the time of booking

For example VIP tickets can only be purchased by business class. Please add some discount ratio as well for different categories of class. So that business class ticket will be most expensive one, then economy class and least expensive ticket will be for family class


GUI

Sabiha Gökçen

Passenger Management System | Plane Management System | Flight Management System

Generate Flight

Departure	Landing	Departure Date	Return Date
istanbul	islamabad	25/04/2023	26/04/2023

Add Flight | Remove Flight

Current Flights

Flight istanbul to islamabad on date 25/04/2023
Flight islamabad to istanbul on date 26/04/2023

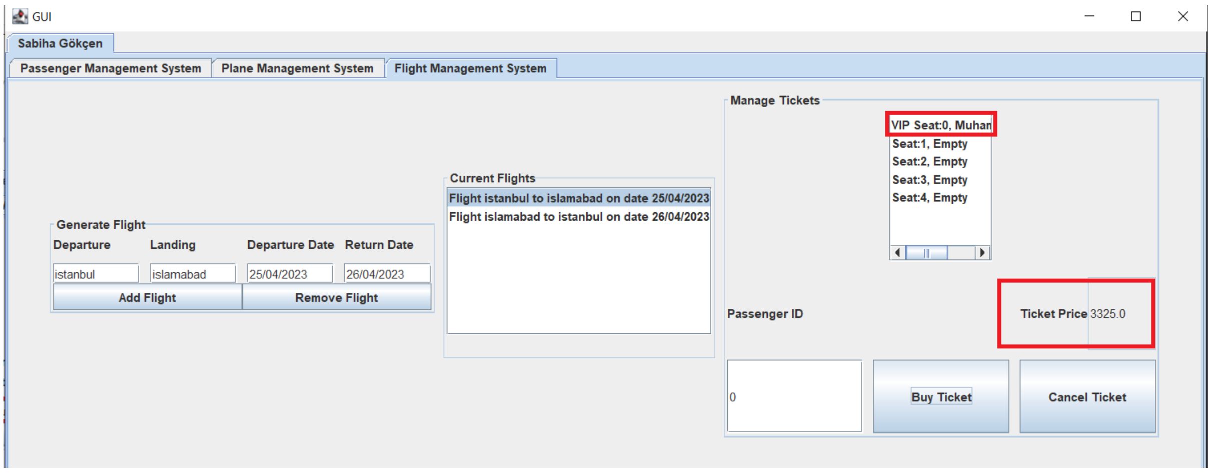
Manage Tickets

VIP Seat:0, Muhammad kashif
Seat:1, Empty
Seat:2, Mert Erkol
Seat:3, Empty
Seat:4, Arda Ozvin

Passenger ID: 2

Ticket Price 1050.0

Buy Ticket | Cancel Ticket


GUI

Sabiha Gökçen

Passenger Management System | Plane Management System | Flight Management System

Generate Flight

Departure	Landing	Departure Date	Return Date
istanbul	islamabad	25/04/2023	26/04/2023

Add Flight | Remove Flight

Current Flights

Flight istanbul to islamabad on date 25/04/2023
Flight islamabad to istanbul on date 26/04/2023

Manage Tickets

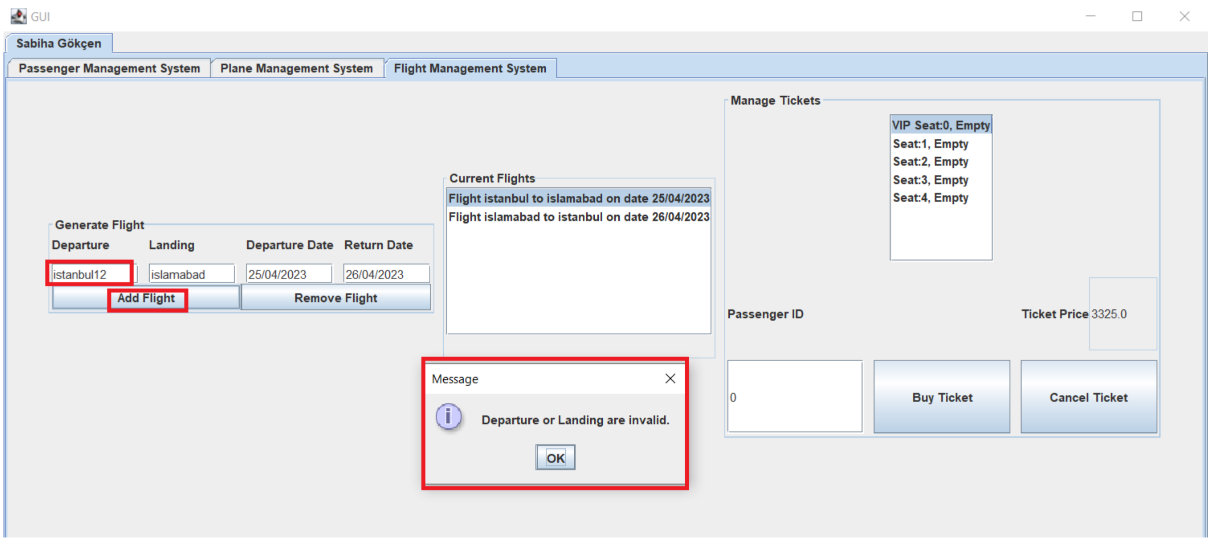
VIP Seat:0, Muhan
Seat:1, Empty
Seat:2, Empty
Seat:3, Empty
Seat:4, Empty

Passenger ID: 0

Ticket Price 3325.0

Buy Ticket | Cancel Ticket

To handle an invalid entry, pop up message need to be displayed


GUI

Sabiha Gökçen

Passenger Management System | Plane Management System | Flight Management System

Generate Flight

Departure	Landing	Departure Date	Return Date
istanbul12	islamabad	25/04/2023	26/04/2023

Add Flight | Remove Flight

Current Flights

Flight istanbul to islamabad on date 25/04/2023
Flight islamabad to istanbul on date 26/04/2023

Manage Tickets

VIP Seat:0, Empty
Seat:1, Empty
Seat:2, Empty
Seat:3, Empty
Seat:4, Empty

Passenger ID: 0

Ticket Price 3325.0

Buy Ticket | Cancel Ticket

Message

Departure or Landing are invalid.

OK

GUI Model View and Controller

Your GUI model will use the same features as shown in above screenshots for reference. You will implement the code using the model, view and controller part. Your program needs to fulfill all the functionalities above as shown in the screenshot

Submission

You will submit the homework via LMS system. You should submit the following java files related to your model

Please make sure to submit java files only, other wise your project will not be graded

1. It must contain the following java files :

   - airport. Java
   - airportmanagemensystem.java
   - airportpassenger.java
   - businesspassenge.java
   - Economypassenger.java
   - Familypassenger.java
   - flight.java
   - flightmanagementssytem.java
   - hanger.java
   - passengermanagementsystem.java
   - plane.java
   - palnemangementssytem.java
   - Regulaticket.java
   - runway.java
   - Ticket.java
   - VIPticket.java

2. MVC files must contain
   - Model.java
   - View.java
   - Controller.java