

---

# Task 3: Project Source on GitHub

Project Report



Ca' Foscari  
University  
of Venice

Jan 21, 2024

CM0639-1

Mazhar Hussain (893479)

---

# Features

## 1. Data Modeling

In order to create a comprehensive flight ticket management system, I designed the data model. This involved defining essential entities such as Flight, Passenger, and Ticket, and establishing relationships between them. Key constraints were implemented, ensuring that each ticket is associated with exactly one flight, and passengers can have multiple tickets. I captured crucial information, including flight numbers, passenger names, and seat numbers, to form the foundation of the system.

## 2. Ticket Reservation

To provide users with a seamless ticket reservation experience, I implemented a function that allows them to reserve a ticket for a specific flight. Users can input their preferred flight details, including flight number, date, and seat preference. The system then checks the availability of the specified seat, reserves it for the user, and delivers a confirmation message on screen containing the reservation details.

## 3. Ticket Cancellation

Empowering users with the ability to cancel their ticket reservations was a key feature. Users can cancel the reservation by clicking cancel button in front of their reservation on booking history page, system updates the seat availability status, and displays a cancellation confirmation message to the user, ensuring a straightforward and user-friendly cancellation process.

## 4. Flight Information Display

To enhance user experience, I implemented a feature to display detailed information about specific flights. Users can input the flight number to retrieve information such as departure time, arrival time, available seats, and other relevant details. This feature aims to provide users with a comprehensive overview of their chosen flights.

## 5. Flight Search

Facilitating users in finding the most suitable flights, I implemented a user-friendly flight search feature. This includes a well-designed interface for users to search for available flights based on criteria such as destination, or flight number. The system displays pertinent information, such as seat availability, departure times, and prices, streamlining the flight selection process.

## 6. Seat Upgrade

Enhancing the overall travel experience, users can now upgrade their seats through a carefully developed feature. By providing reservation details, users can check the availability of seat upgrades for their current seat. If an upgrade is possible, users are presented with options, and upon confirmation, the reservation is updated with the new seat information.

---

## 7. Booking History

For user convenience, a feature was created to display a comprehensive booking history. Upon providing account credentials, users can access a list of all past and upcoming flight reservations. The displayed information includes relevant details such as flight number, date, and seat, allowing users to track and manage their travel history effortlessly, they can also cancel a flight from here.

## 8. User Authentication

Security is paramount, and therefore, I integrated a robust user authentication system. Users can create accounts with unique usernames and passwords. A login function verifies user credentials, ensuring that only authenticated users can reserve or cancel tickets, providing a secure environment for their interactions with the system.

## Admin Specific Features

Admin can also perform all the actions mentioned above, in addition the admin can also create new flights in the system and confirm the tickets for the users

## GitHub Repo:

<https://github.com/mazhartejani/airlinesreservation>

## Installation Instructions:

1. Clone the repository:

```
git clone https://github.com/mazhartejani/airlinesreservation.git
```

2. Navigate to the project directory:

```
cd airlinesreservation
```

3. Navigate to the Docker directory:

```
cd docker
```

4. Make the setup script executable:

```
chmod +x envsetup.sh
```

---

5. Start the Docker containers:

```
docker-compose up -d
```

Wait until all containers are in the running state.

6. Run the setup script:

```
bash envsetup.sh
```

This script will set up databases and install all the dependencies.

7. Application will start at port:80

```
http://localhost:80
```

## Login Credentials:

### 1. Passenger:

email: mazhar@test.com

password: 12345678

### 2. Admin:

email: admin@test.com

password: 12345678