A blue and white logo

Description automatically generated with low confidence

**Flight Booking System**

By

|  |  |
| --- | --- |
| Aya Elneanaei Fouda | 202002609 |
| Salma Ahmed Ali | 202000757 |
| Eman Awney Abdalla | 202001001 |
| Khadija Khaled Mohamed Hatem | 19200008 |
| Randa Aldaba Mahmoud | 202003003 |

CSCI305: Database Systems

Supervised By:

Dr. Mona Arafa

Eng. Amira Tarek

Table of Contents

[I. Introduction 3](#_Toc136316497)

[II. Project Objective 3](#_Toc136316498)

[III. ERD and Business Rules 4](#_Toc136316499)

[Descriptions 5](#_Toc136316500)

[***A.*** ***Entity Description*** 5](#_Toc136316501)

[***B.*** ***Attribute Description*** 6](#_Toc136316502)

[IV. Normalization 10](#_Toc136316503)

[V. Mapping 12](#_Toc136316504)

[VI. MY SQL Diagram 13](#_Toc136316505)

[VII. SQL Implementation and Codes: 14](#_Toc136316506)

[A. Tables Creations 14](#_Toc136316507)

[*B.* *Sample Select Query* 18](#_Toc136316508)

[*C.* *Sample Insert Query* 18](#_Toc136316509)

[*D.* *Sample Update Query* 19](#_Toc136316510)

[*E.* *Sample Delete Query* 19](#_Toc136316511)

[*F.* *Sample Create View* 20](#_Toc136316512)

[VIII. GUI Screenshots 20](#_Toc136316513)

[Tools Used 20](#_Toc136316514)

[Samples 20](#_Toc136316515)

[IX. Conclusion 25](#_Toc136316516)

# Introduction

The flight booking database system has revolutionized the way air travel arrangements are managed by providing a comprehensive and efficient solution. Prior to its development, the process of booking flights and managing travel arrangements was often time-consuming, cumbersome, and prone to errors. In the past, travelers had limited options for securing their desired flights, relying on manual methods such as visiting airline ticket counters or travel agencies, making phone calls, or sending faxes. These methods often resulted in delays, limited availability of flight information, and a lack of flexibility in making changes to bookings. The traditional paper-based systems used by airlines, airports, and travel agencies were also inefficient in managing flight schedules, seat availability, and passenger information.

The flight booking database system addresses these challenges by serving as a centralized repository of critical flight-related information and passenger details. It provides real-time access to accurate and up-to-date flight schedules, airline options, departure and arrival times, and available seats. With a few clicks, they can compare options, select the most suitable flight, and proceed with the booking process. One of the significant advantages of the flight booking database system is its ability to handle a large volume of bookings and changes efficiently. With the automated system, airlines, airports, and travel agencies can easily manage flight schedules, seat availability, and passenger information. This ensures that bookings are processed accurately and promptly, reducing the risk of errors, and providing a seamless booking experience for travelers.

# Project Objective

The main objective of the flight booking database system is to provide a user-friendly and efficient platform for travelers to book and manage their flights and other related services. This includes making it easy for users to search for available flights and services and make reservations.

* Allow user to book flights, pay for them and get issued tickets.
* The system would store information about passengers, including their names, contact information, and flight itineraries.
* Allow users to search for all available flights with their different schedules.

# ERD and Business Rules

A picture containing diagram, text, line

Description automatically generated

## **Descriptions**

Flight represents an individual airplane flight from pickup to destination. It is referenced by a flight ID and has an expected pickup and arrival time.

User represents any user of the database management system, whether it be an admin or a passenger. A user has a name, phone number, email and password.

An admin registers flights to be booked by passengers and has an admin ID. The admin also manages the whole application.

When a passenger makes a booking, they create an account on the system and books with their details (name, nationality, age, and their passport number). They choose a pickup and destination time and place on their preferred airline, after which they’re issued a ticket.

A ticket includes the gate of the airplane to which the­­­ passenger should head to catch their flight, as well as their seat on the airplane.

### ***Entity Description***

|  |  |
| --- | --- |
| Staff | User who uses the software. |
| Admin | Software Admin |
| Passenger | Personal booking information of passengers |
| Flight | The flight trip who passengers’ books to |
| Booking | Where trip is booked, with its destination and schedule |
| Ticket | The issued ticked for a booking |
| Fare | Fare of booking |

### ***Attribute Description***

|  |  |
| --- | --- |
| Staff | |
| Email | Email of the user and unique identifier for each user. This field is required by the DBMS Primary key  Type: Text |
| Name | Name of the user. Composite attribute, first name, middle name and last name  Type: Text |
| Password | Password of each staff.  Type: Number |
| Phone number | User’s phone number. Can be multivalued  Type: text |

|  |  |
| --- | --- |
| Admin: Subtype of user | |
| AdminID | ID of the admin  Type: Text |
| Email | Contact email of the admin |
| Phone\_number | Content telephone number |
| Password | To store password |
| name | name of the user. Composite attribute, first name, middle name and last name |
| Gender | Gender of admin |

|  |  |
| --- | --- |
| Passenger: Subtype of user | |
| PassportID | Passport number used for booking  Type: text |
| Nationality | Passenger’s nationality  Type: Text |
| Birthdate | Birthdate of passenger  Type: date |
| Age | Age of passenger .  Type: Number |
| Name | name of the user. Composite attribute, first name, middle name and last name |
| Email | Email of the passenger and unique identifier for each user. This field is required by the DBMS Primary key |
| Phone\_num | Passenger phone number  Type: Number |

|  |  |
| --- | --- |
| Booking | |
| bookingID | Passport number of the passenger and unique identifier for each passenger. This field is required by the DBMS  Primary key  Type: Text |
| Airline | Airline name of the flight.  Type: Text |
| Time | Time of the trip booked |
| TicketNum | Ticket number issued to the passenger  Type: Number |
| Date | Data of passenger contain data\_T and data\_R  Type: text |
| Class | Class of the seat |
| FlightID | ID of the flight booked |
| Desttination | Destination of passenger |
| From | Place from which to travel |

|  |  |
| --- | --- |
| Ticket | |
| TicketNum | Ticket number issued for each passenger  Primary key  Type: Text |
| passportID | Passport ID of the passenger.  Type: Text |
| Gate | Gate number  Type: Text |
| Date | Date of passenger.  Type: Date |
| Seat | Seat number on the flight |

|  |  |
| --- | --- |
| Flight | |
| FlightID | Flight id of each trip  Primary key  Type: Text |
| Seats | Number of seats available on the train  Type: Text |
| Destination | Destination of the flight.  Type: text |
| Gate | Gates of flight  Type: text |
| Time | Time of departure  Type: Time |
| Date | Date of arrival  Type: Date |
| Airline | Airline of the flight  Type: text |
| pay | Pay option of the payment  Type: text |
| From | The place from which to travel |

|  |  |
| --- | --- |
| books for(ternary Entity) | |
| email | Primary key  Type: Text |
| Booking\_id | Primary key  Type: Bool |
| PassportID | Primary key  Type: number |
| Amount | Amount number |
| Transation\_id | Transation id of eash payment |

|  |  |
| --- | --- |
| Fare | |
| Id | Primary key |
| from | Country from |
| To | Country to |

# **Normalization**

All the database tables specified above are in 3rd normal form accept.

1-flight

Identify the functional dependencies

Flight\_ID -> {airline,Destination, seats}

2-Booking

Identify the functional dependencies

Booking\_ID-> From, dest, Fare

3-Ticket

Identify the functional dependencies

{Ticket\_num } -> {seats, date}

4-iso\_code

Identify the functional dependencies

iso\_code->country

5- staff

Identify the functional dependencies

email-> { Fname,Mname,Lname, Phone}

6-passenger

Identify the functional dependencies

passportID -> Fname,Mname,Lname, phone\_num,Email,nationality

7-fare

Identify the functional dependencies

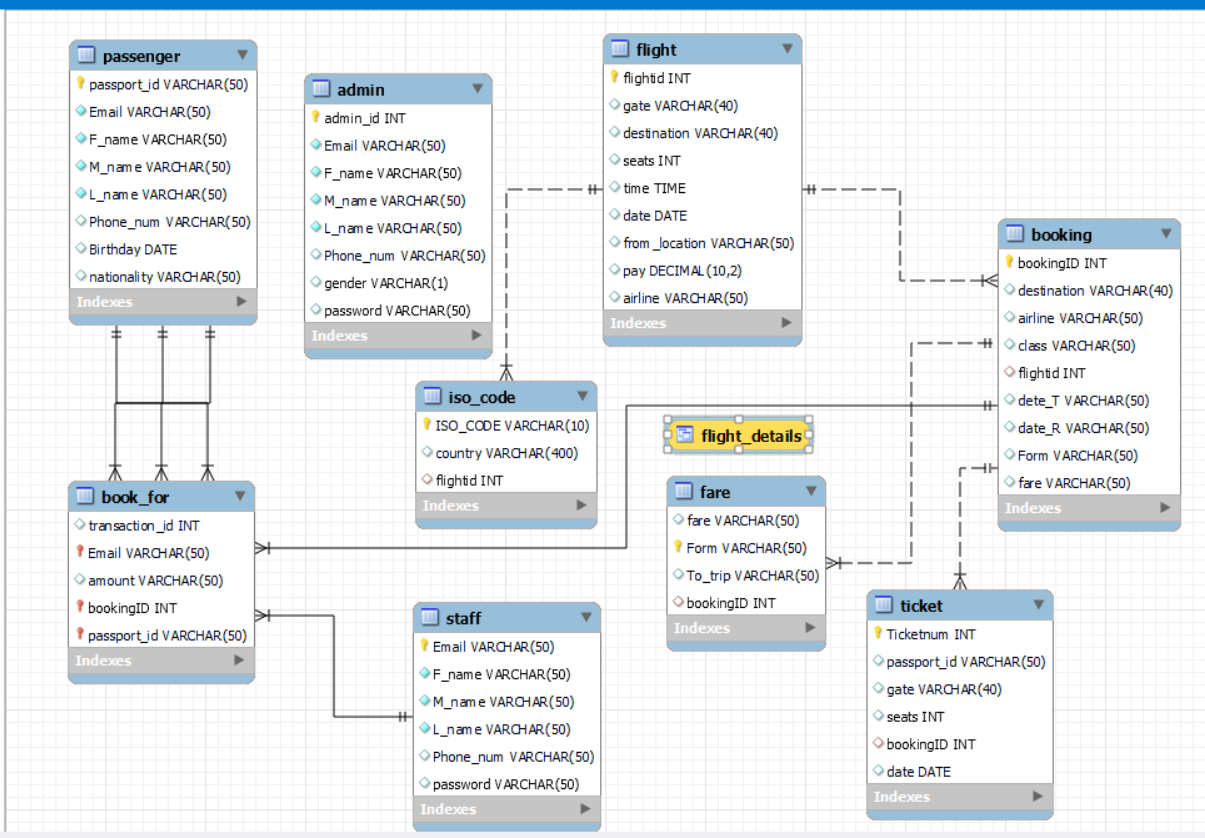
Fare-> to,from

# **Mapping**

A screenshot of a computer screen

Description automatically generated with low confidence

# **MY SQL Diagram**



# **SQL Implementation and Codes:**

## **Tables Creations**

***Create Database & use it***

create database flight\_booking\_final;

use flight\_booking\_final;

***Create flight table***

CREATE TABLE flight (

flightid INT PRIMARY KEY AUTO\_INCREMENT,

gate VARCHAR(40),

destination VARCHAR(40),

seats INT,

time TIME,

date DATE,

from\_location VARCHAR(50),

pay DECIMAL(10,2),

airline VARCHAR(50)

);

***Create booking table***

create table booking(

bookingID INT PRIMARY KEY AUTO\_INCREMENT,

destination VARCHAR(40),

airline VARCHAR(50),

class varchar(50),

flightid INT ,

dete\_T varchar(50),

date\_R varchar(50),

Form varchar(50),

fare varchar(50)

);

ALTER TABLE booking ADD FOREIGN KEY (Flightid) REFERENCES flight(flightid);

***Create fare table***

create table fare(

fare varchar(50),

Form varchar(50) PRIMARY KEY,

To\_trip varchar(50),

bookingID INT

);

ALTER TABLE fare ADD FOREIGN KEY (bookingID) REFERENCES booking(bookingID);

***Create table Ticket***

create table Ticket(

Ticketnum INT PRIMARY KEY AUTO\_INCREMENT,

passport\_id varchar(50),

gate VARCHAR(40),

seats INT,

bookingID INT,

date DATE);

ALTER TABLE Ticket ADD FOREIGN KEY (bookingID) REFERENCES booking(bookingID);

***Create table staff***

create table staff(

Email varchar(50) primary key not null,

F\_name varchar(50) not null,

M\_name varchar (50) not null,

L\_name varchar (50) not null,

Phone\_num varchar(50) ,

password varchar(50)

);

***Create table passenger***

create table passenger(

passport\_id varchar(50) primary key not null,

Email varchar(50) not null,

F\_name varchar(50) not null,

M\_name varchar (50) not null,

L\_name varchar (50) not null,

Phone\_num varchar(50) ,

Birthday date ,

nationality varchar(50));

***Create table book\_for***

create table book\_for(

transaction\_id int ,

Email varchar(50) ,

amount varchar(50),

bookingID INT,

passport\_id varchar(50),

primary key (Email,bookingID,passport\_id)

);

***Create table admin***

create table admin(

admin\_id int auto\_increment primary key,

Email varchar(50) not null,

F\_name varchar(50) not null,

M\_name varchar (50) not null,

L\_name varchar (50) not null,

Phone\_num varchar(50) ,

gender varchar(1),

password varchar(50)

);

## ***Sample Select Query***

***A picture containing text, screenshot

Description automatically generated***

***A picture containing screenshot

Description automatically generated***

## ***Sample Insert Query***

***A picture containing text, multimedia software, software, screenshot

Description automatically generated***

## ***Sample Update Query***

***A screen shot of a computer

Description automatically generated with medium confidence***

## ***Sample Delete Query***

***A screenshot of a computer

Description automatically generated with medium confidence***

## ***Sample Create View***

***A screenshot of a computer

Description automatically generated with medium confidence***

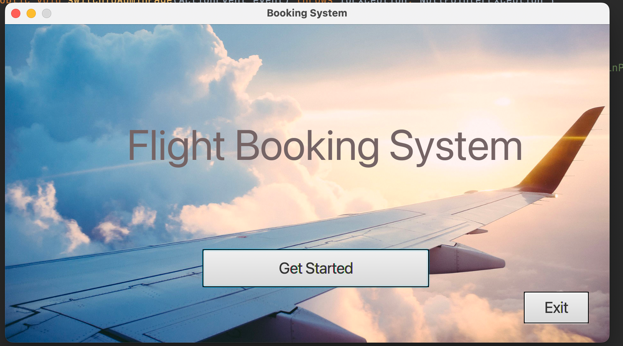
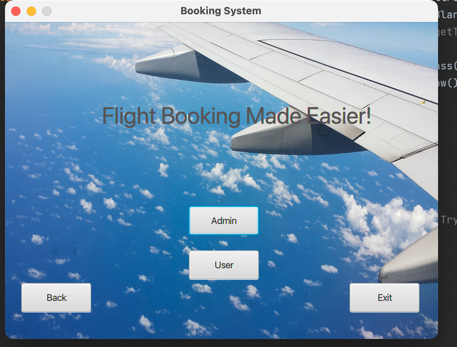
# **GUI Screenshots**

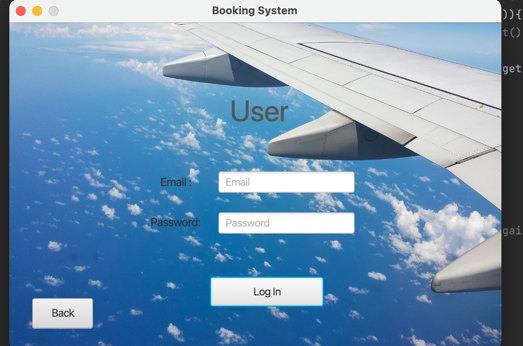
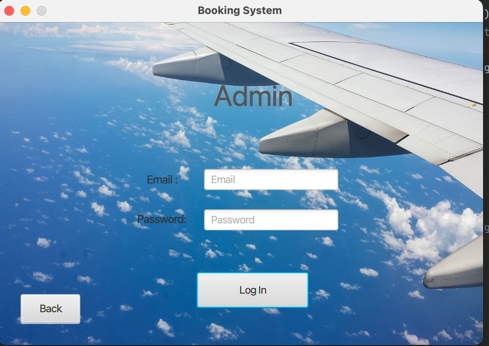
## **Tools Used**

The software is implemented using Java, and the GUI using Javafx, and MySQL and MySQL workbench for the database.

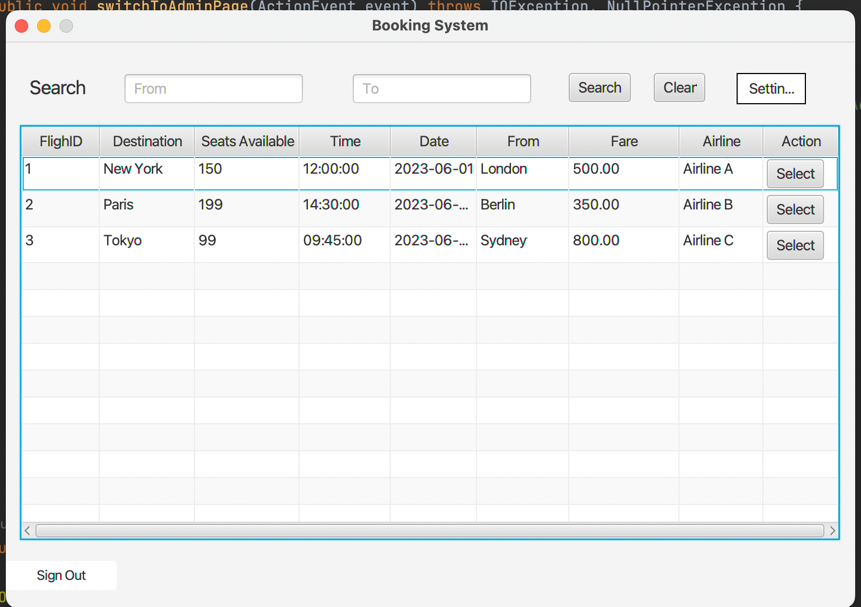
## **Samples**

Homepage, User/Admin side and the Log-in pages

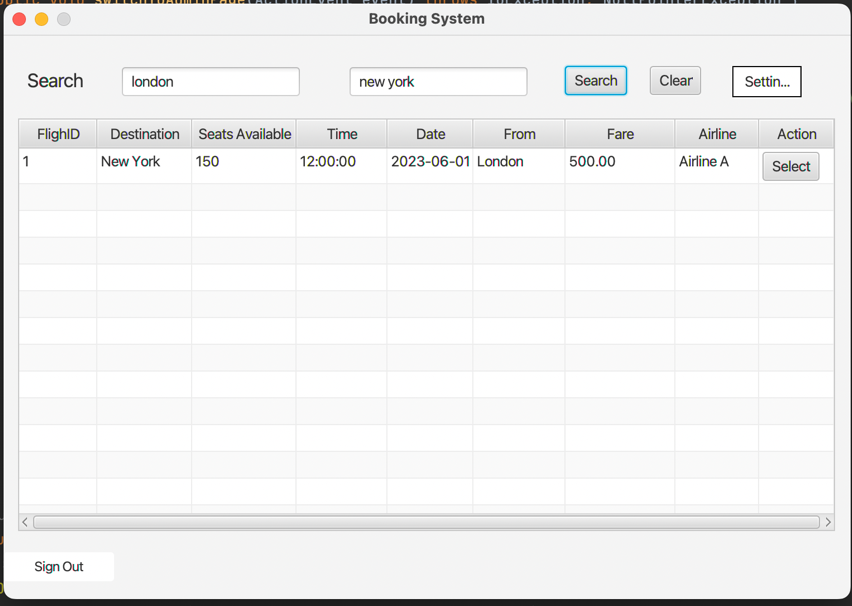
 

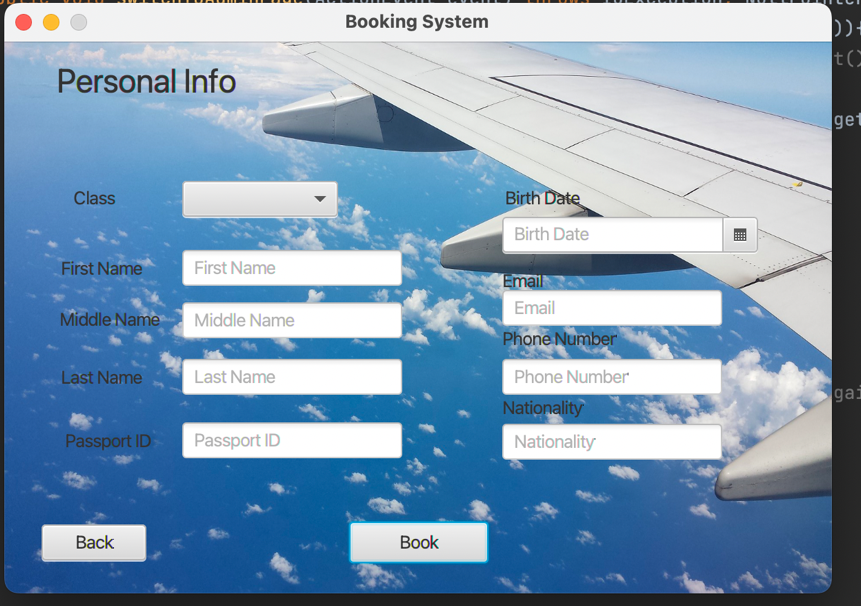
User page that displays flight to select from, and search function.



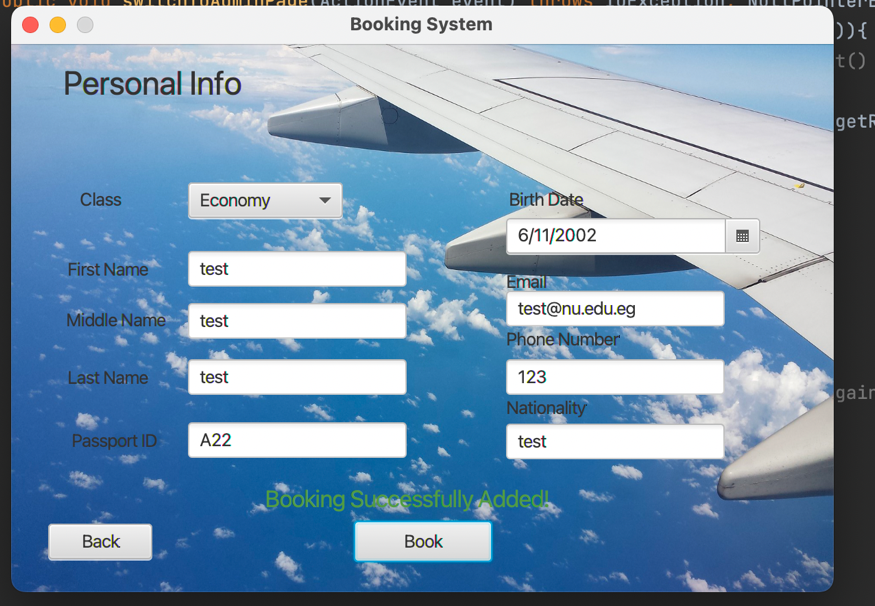
Search Function used



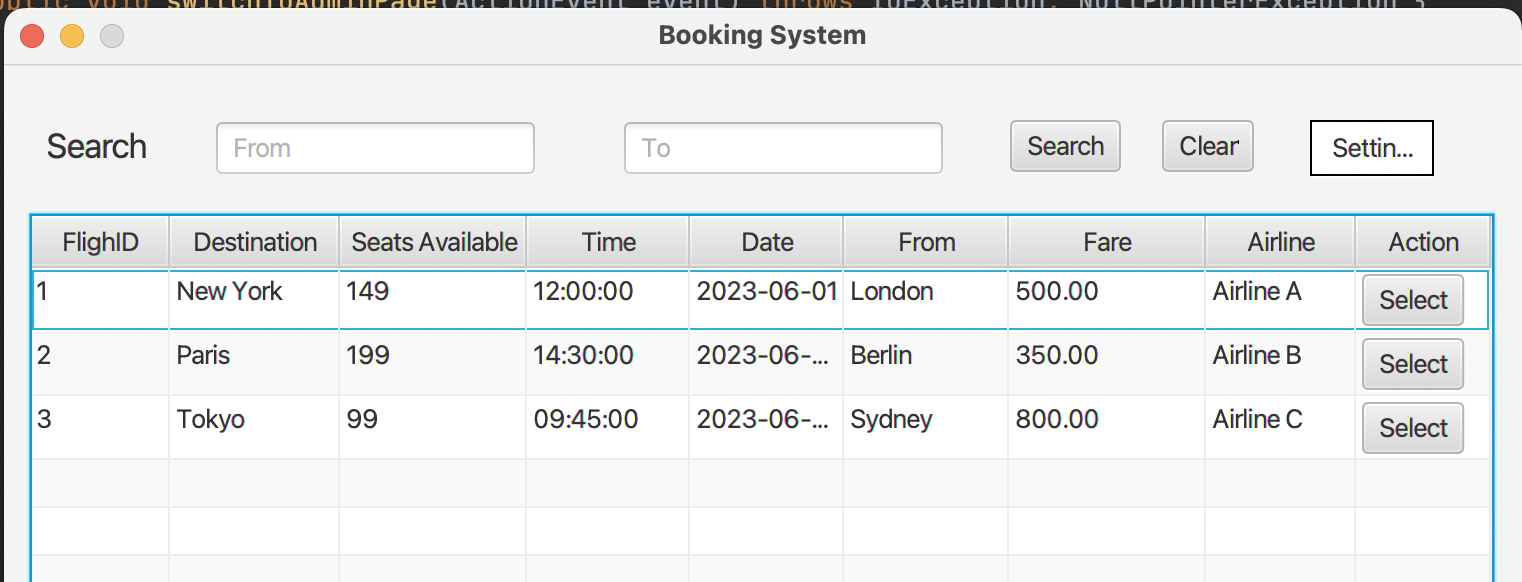
When a flight is selected, user is prompted to enter the passenger information to complete the booking.



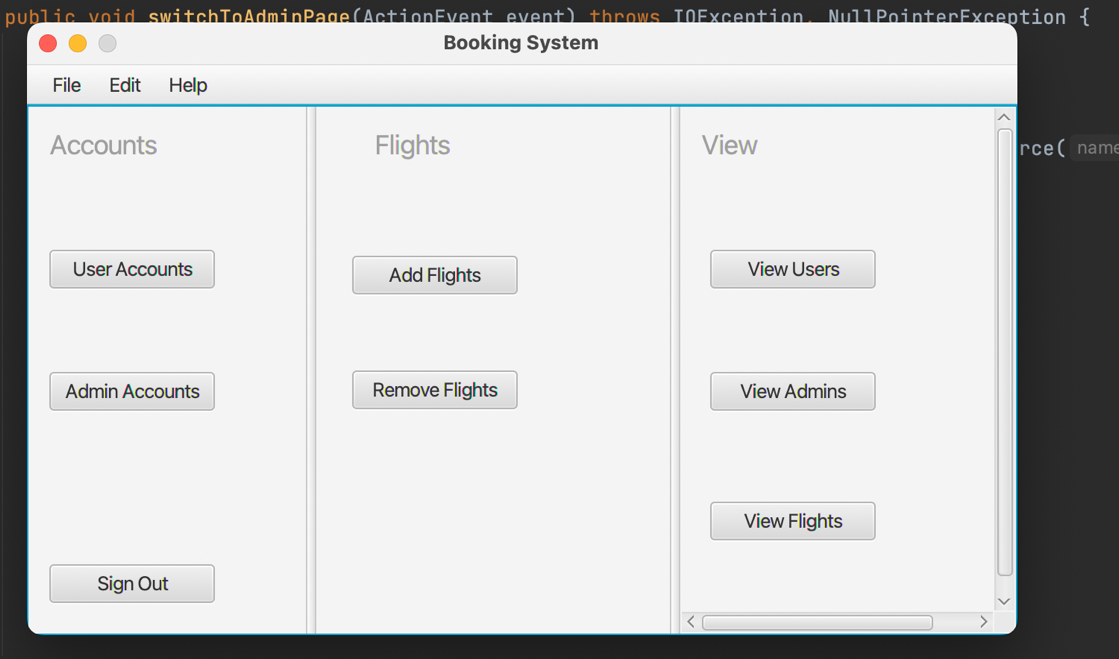
Passenger information entered and booking is done successfully!



Booking is done successfully, and seats available is now reduced by 1.



Admin side and functions.



Viewing Admins in the system.

A screenshot of a computer

Description automatically generated with medium confidence

# **Conclusion**

In conclusion, our project “Flight Booking Database System” aims to facilitate booking of airplane flights. In doing so, it meets the needs of travelling agencies, allowing them to book flights in an effective manner. The system’s database design has been created and discussed in this document, including the Entity Relationship Diagram (ERD) and the database mapping. The system is implemented using modern tools and technologies to ensure a great user experience and high performance and security.