AIRLINES
BOOKING SYSTEM

Version 0.1

Prepared by:

- **Menna ashraf hanafy**
- **Hoda Mohamed Abdellatif**
- **Zeyad ashraf Mahmoud**

Faculty of Engineering

1. Preface

1.1 Document Purpose

The purpose of this document is to provide a detailed and complete specification of Airlines Booking System (ABS).

The document will provide an overview of the system in the first section; then each part will be explained in detail in the second and third sections.

2. Introduction

2.1 Purpose

-The main objective of Airlines Booking System is to manage the details of Airlines Ticket, Flights, Customers, Booking, create user-friendly efficient and secure platform that allows the user to book flights, manage their reservations, and reduce the manual work where the user can create account and have the accessibility to select as many flights as he wanted at the preferred time. -The system provides a simple and intuitive interface that enables users to search for flights based on destination. -The system should allow the user to view flight schedules, seat availability and choose their preferred seats. -Additionally, the system should be able to manage user information, including personal details and travel history

- Make Account
- Log In.
- manage user information
- Search for a flight

- Booking a flight
- choose preferred seats
- Show ticket’s information
- Show the booked flights schedule

2.2 Scope

Airlines Booking System (ABS) is a software to manage the details of Airlines Ticket, Flights, Customers, Booking, create user-friendly efficient and secure platform that allows the user to book flights, manage their reservations, and reduce the manual work

.

2.3 Similar Systems

Our system is similar to some existing systems which allow user to search for specific flight based on their needs, make reservations and view the flight's schedule.

3. Glossary

3.1 Acronyms, definitions, and abbreviations.

- **ABS**: Airlines Booking System
- **User**: Passengers, Airlines Administrator

4. System Users

4.1 System stakeholders

- System Engineer
  - Responsible for requirements gathering
  - Responsible for development
  - Responsible for deployment and support

- Administrator
  - add new flights
  - delete flight
  - add empty seats
  - delete seats
  - Do all other functionalities mentioned above.

- End user
  - Sign up
  - Log in
  - Booking flight
  - View his flights schedule.

4.2 User's objectives

- System Engineer:
  - Gain Experience in software engineering and development

- Administrator
    - Ensure students data security

5. User Requirements definitions

5.1 System Functions

1. Make account
2. Log in
3. Manage user information
4. Search for a flight by source
5. Search for a flight by destination
6. Choose a flight based on preferred time and date
7. Choose preferred seats
8. Choose flight's class
9. Book a flight
10. Show ticket's information
11. View the booked flight schedule

5.2 Constraints

- Airlines Policies
    - All data should be ready in printable form.
- Cultural Constraints
    - All Code must follow Team standards
- Technologies Limitations
    - No support for Internet Explorer
- Hardware limitations
    - Single Server Deployment issues

6. System Functional requirements

6.1 Make account

The user can make a new account by adding the following information:

- Main information (name, birthdate, nationality, address)
- Contact information (phone number)
- Email, password

6.2 login

user can login to the system by entering his email and password the system checks whether the user’s account is in the database or not and whether the password is correct or not

6.3 manage user information

the system should be able to manage user information through database tables which include the user's email and password, including personal details such as user's name, address, birthdate, and travel history through database tables which include source and destination of flights, airport, flight date and arrival date

6.4 Search for a flight by source

A user should be able to search for a certain flight by specifying the required source of flight.

6.5 Search for a flight by destination

A user should be able to search for a certain flight by specifying the required destination of flight

6.6 Choose a flight based on preferred time and date

A user should be able to choose a flight according to his preferred time and date

6.7 Choose preferred seats

A user should be able to choose preferred seats while booking the flight

6.8 Choose flight's class

After choosing the required flight based on preferred time and date then choosing the preferred seat form the available empty seats then the user should be able to choose the flight's class whether it's economic, business or first class.

6.9 Book a flight

After choosing the required flight based on preferred time and date then choosing the preferred seat form the available empty seats then the user should be able to choose the flight's class whether it's economic, business or first class where each type has a different price

The user should confirm the ticket's price then the flight will be booked

6.10 Show ticket's information

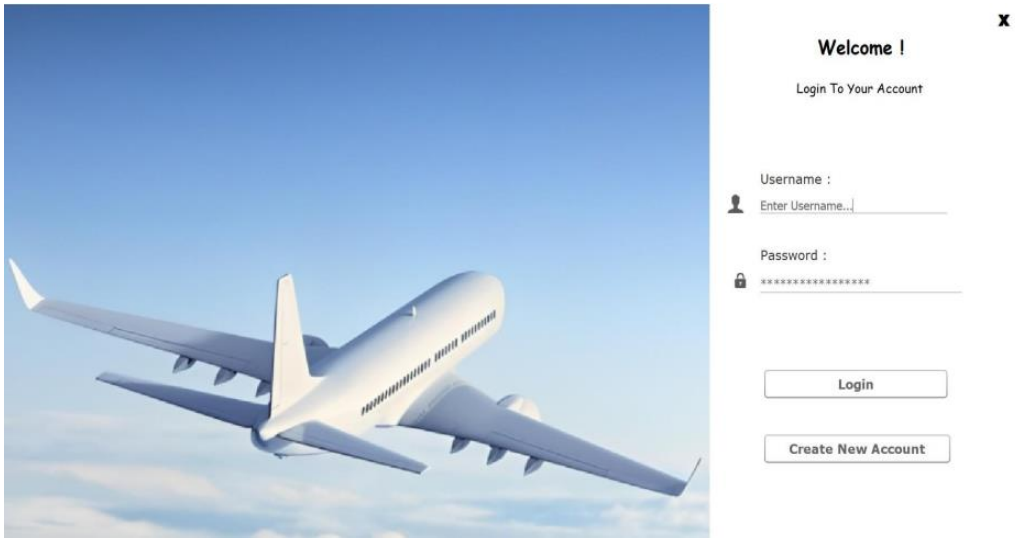
After finishing booking the flight, the application will register the information in the database then it will provide the user with the ticket's information which include the ticket's price, flight number, time of flight, date of flight, the source and the destination of flight and flight's duration

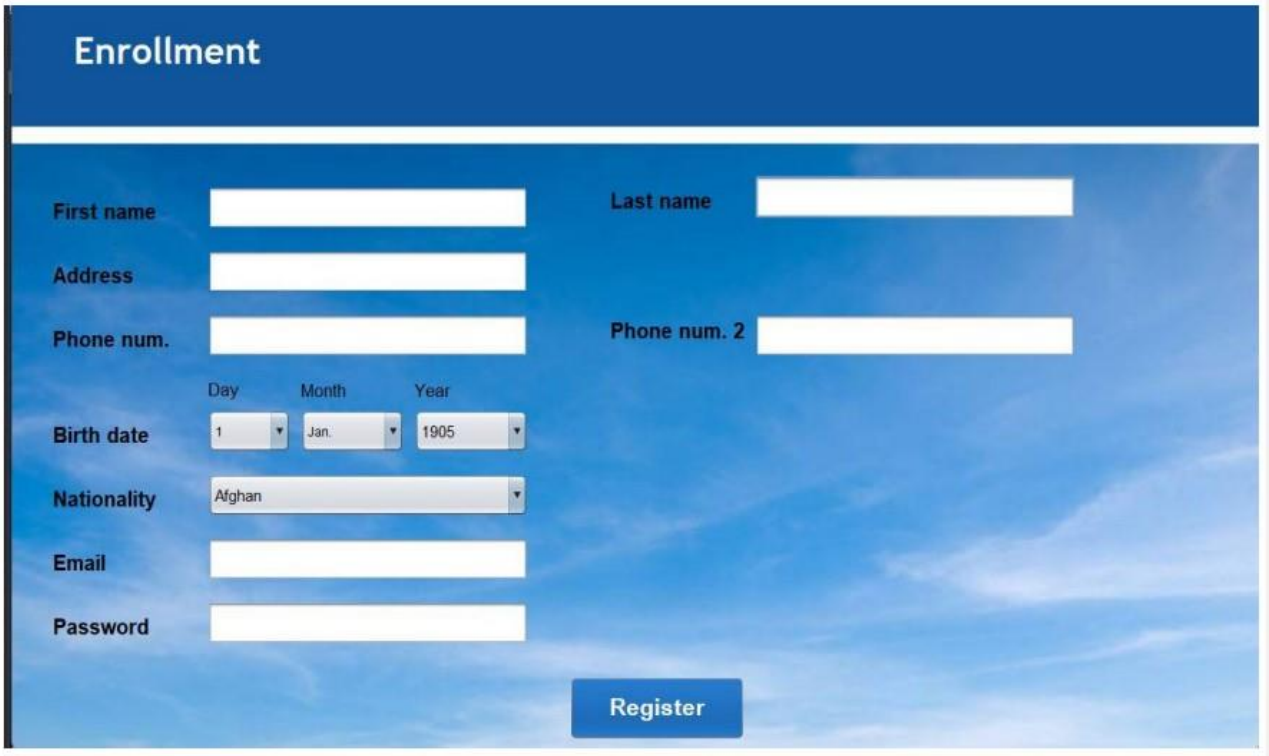
6.11 View the booked flight schedule

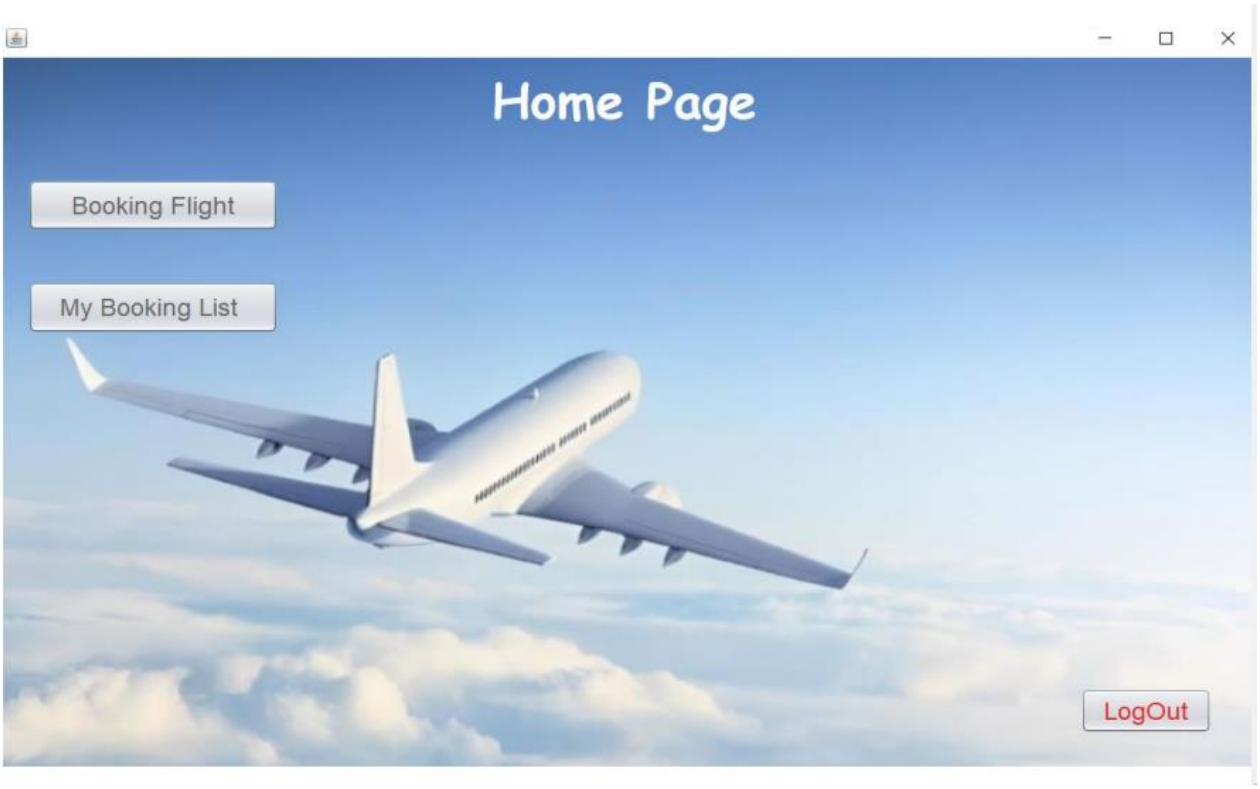
A user can access all the flights that he has booked through the application where each flight's information that he has booked is registered in the database and then the application will provide these information in the form of graphical interface for the user.

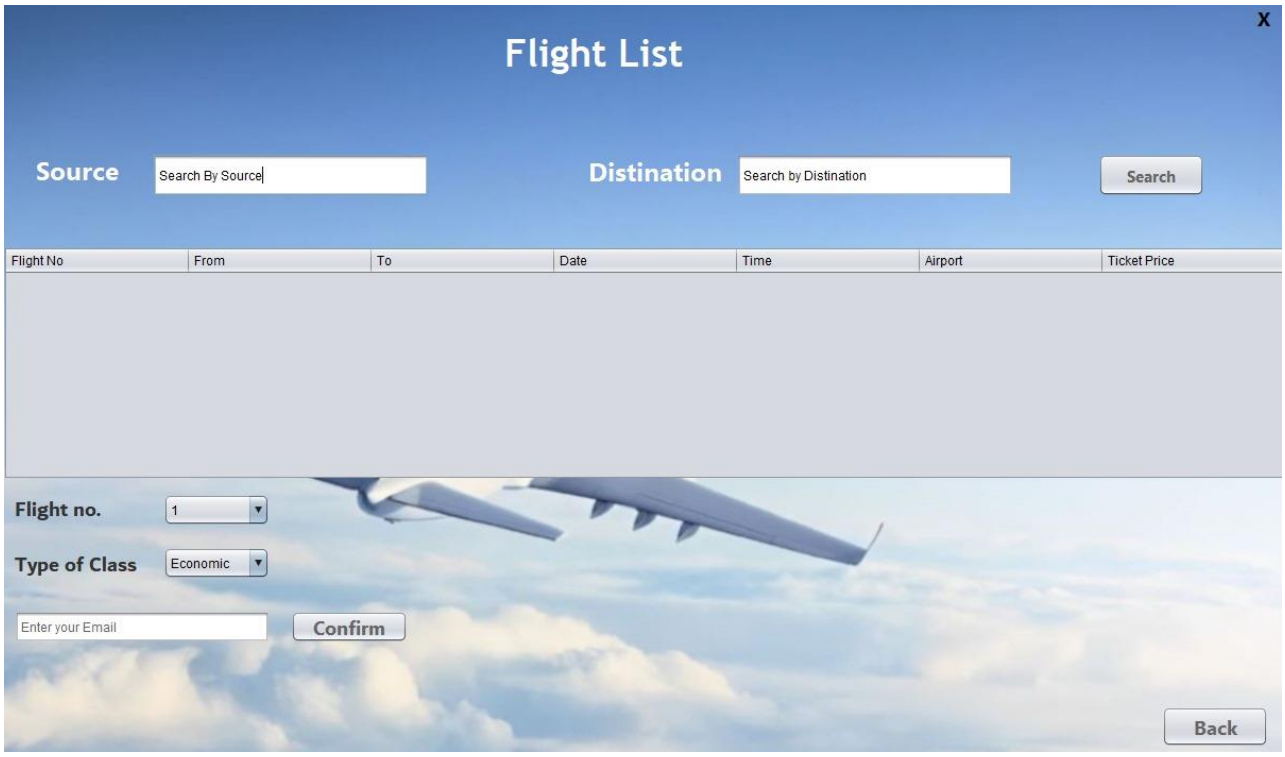
7. Interface requirements

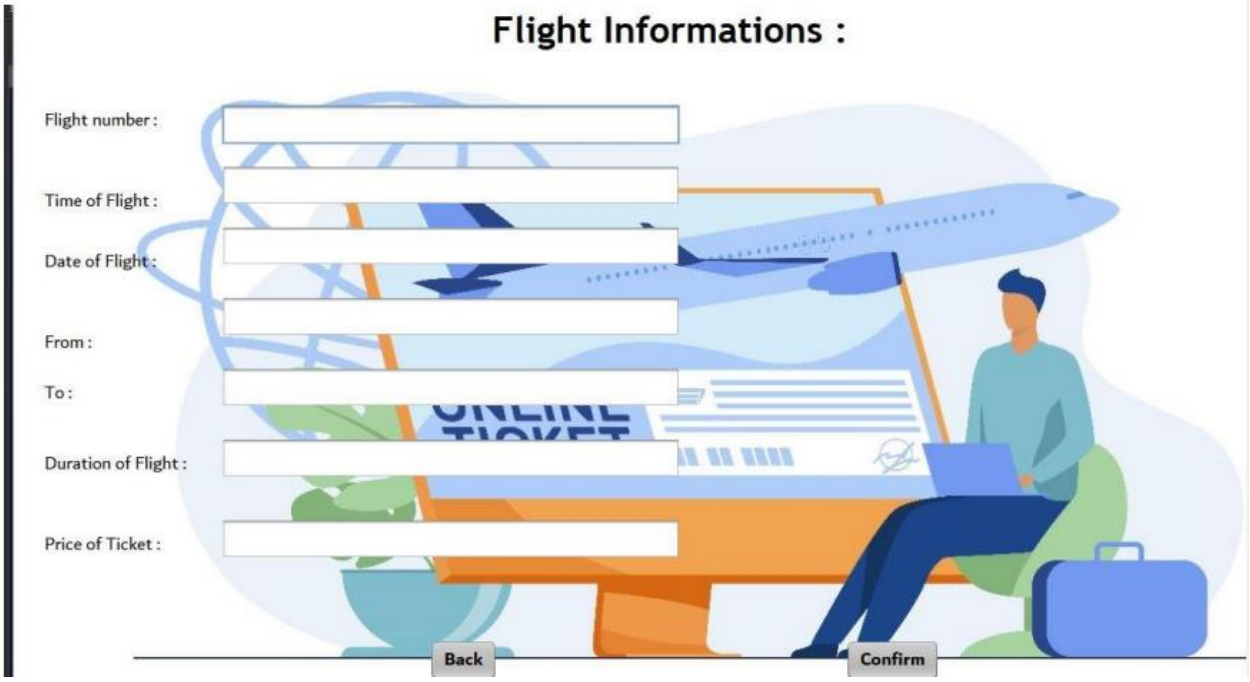
7.1 User interfaces:

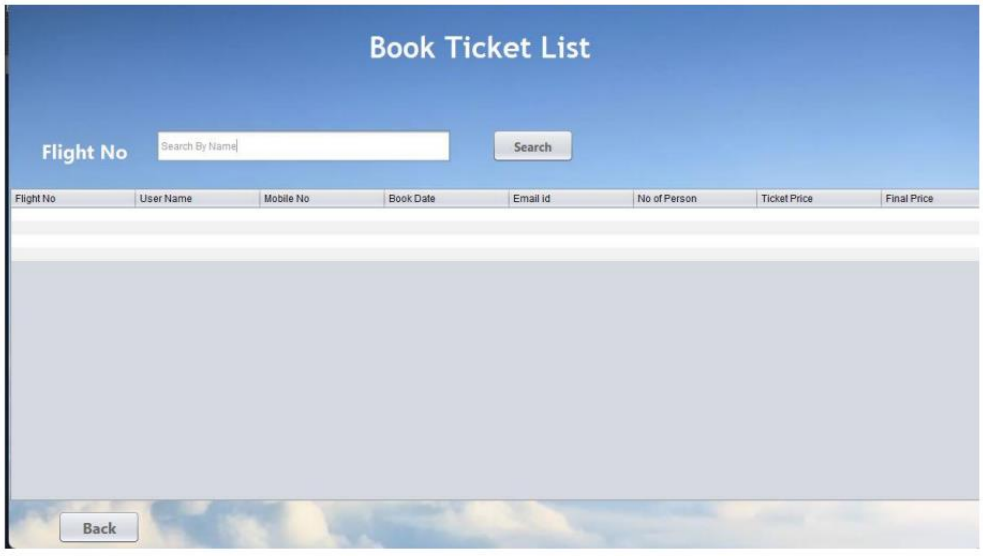

x
Welcome !
Login To Your Account
Username :
Enter Username...
Password :
Login
Create New Account


Enrollment
First name
Last name
Address
Phone num.
Phone num. 2
Day
Month
Year
Birth date
1
Jan.
1905
Nationality
Afghan
Email
Password
Register


Home Page
Booking Flight
My Booking List
LogOut


Flight List
Source
Search By Source
Distination
Search by Distination
Search
Flight No
From
To
Date
Time
Airport
Ticket Price
Flight no.
1
Type of Class
Economic
Enter your Email
Confirm
Back


Flight Informations :
Flight number :
Time of Flight :
Date of Flight :
From :
To :
Duration of Flight :
Price of Ticket :
Back
Confirm


Book Ticket List
Flight No
Search By Name
Search
Flight No
User Name
Mobile No
Book Date
Email id
No of Person
Ticket Price
Final Price
Back

7.2 Software

- Database access will be wrapped through Object Relation Mapping Framework (Entity framework)
- Integration points will be handled in future versions

8. Non-functional requirements

8.1 Availability:

The system should be available during the whole day.

8.2 Security:

No one can access the system database from outside the company.
No user can access the data of another user

8.3 Safety:

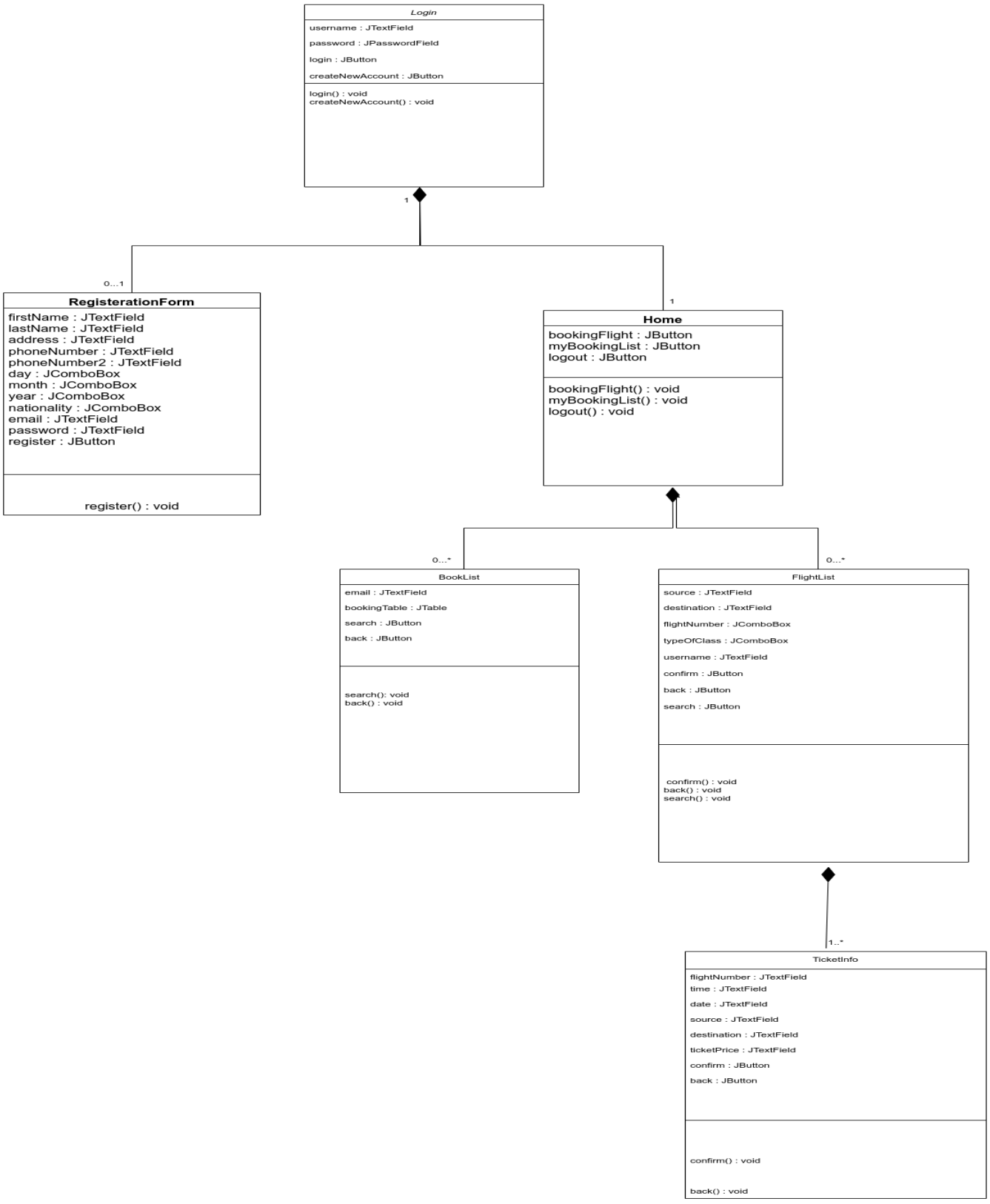
Database should always be connected to the application.

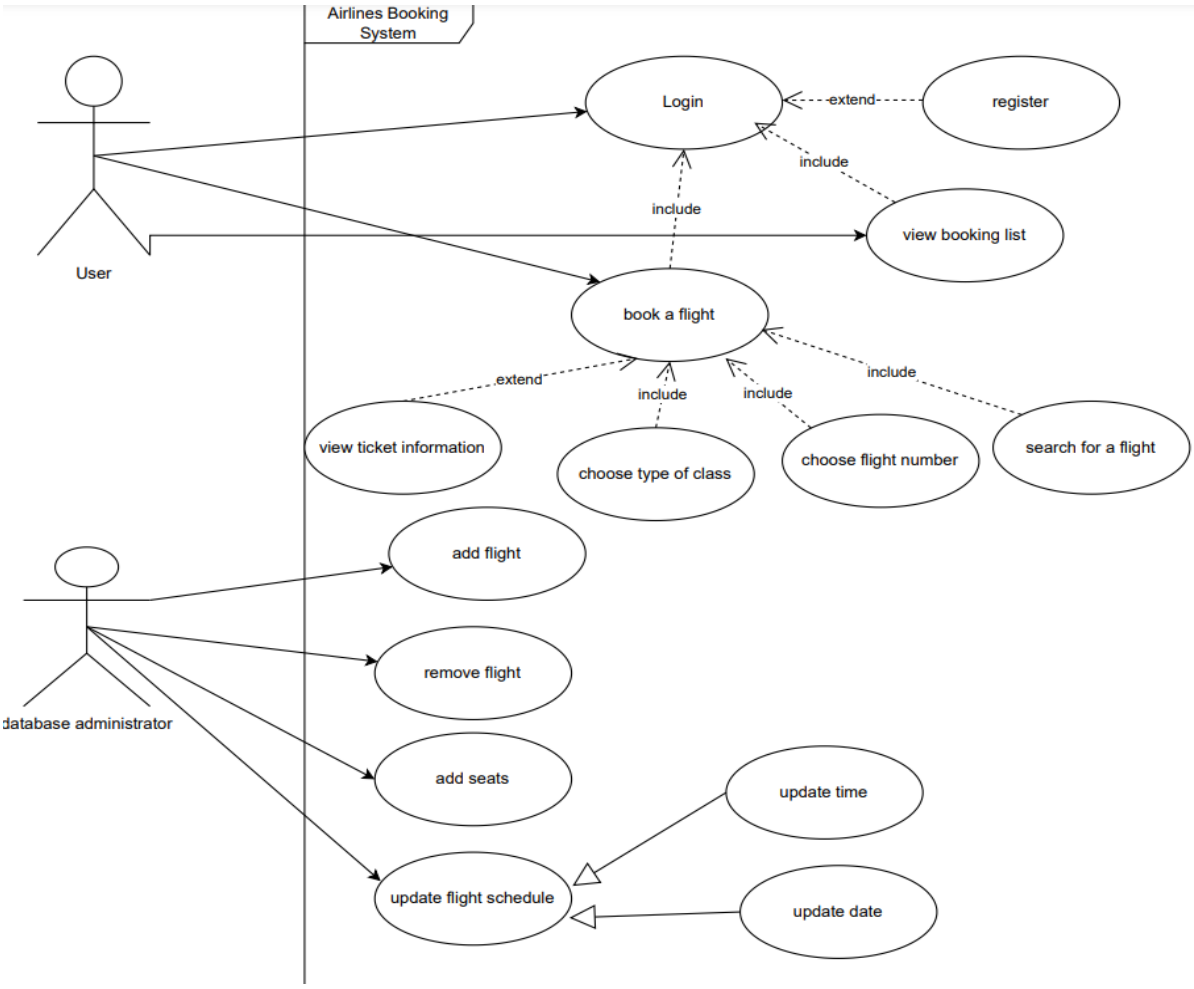
8.4 Reliability:

Any changes made by the user through the application should be handled by the database.

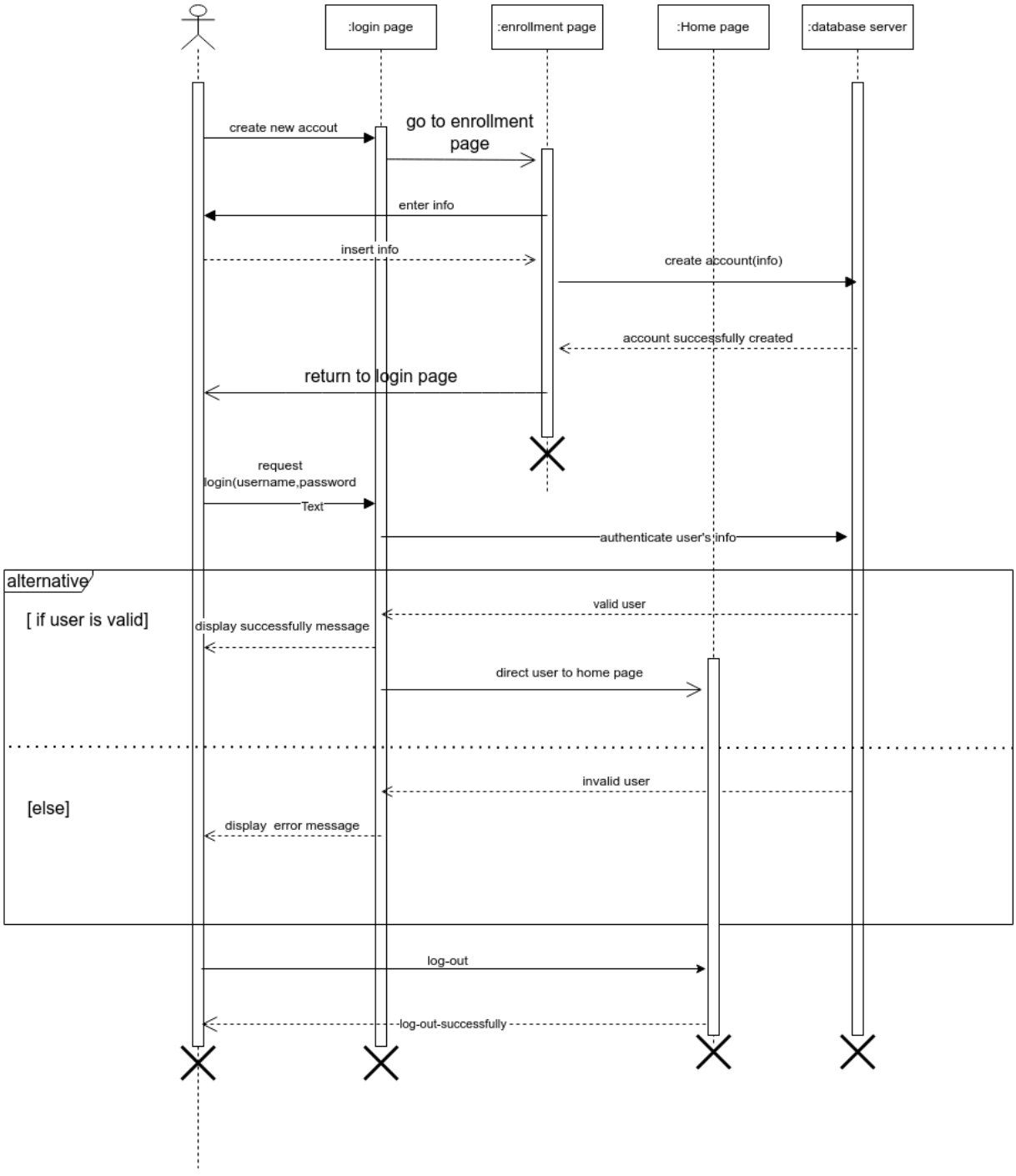
9. System Models and Diagrams

Here you should include the system's context diagram, use case diagram, class diagram and sequence diagrams for any needed functional requirement.

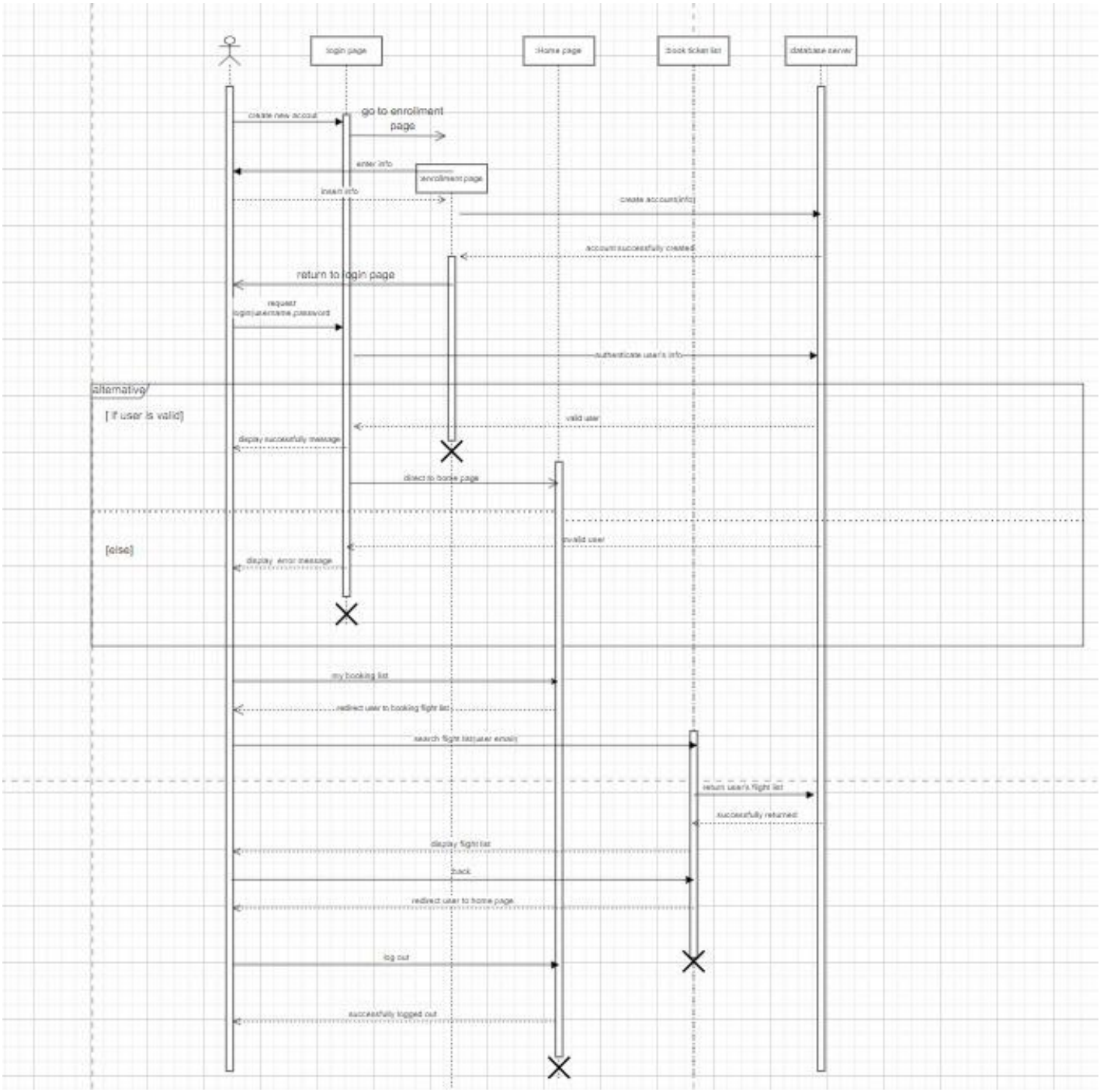

Login
username : JTextField
password : JPasswordField
login : JButton
createNewAccount : JButton
login() : void
createNewAccount() : void
1
0...1
1
RegisterationForm
firstName : JTextField
lastName : JTextField
address : JTextField
phoneNumber : JTextField
phoneNumber2 : JTextField
day : JComboBox
month : JComboBox
year : JComboBox
nationality : JComboBox
email : JTextField
password : JTextField
register : JButton
register() : void
Home
bookingFlight : JButton
myBookingList : JButton
logout : JButton
bookingFlight() : void
myBookingList() : void
logout() : void
0...*
0...*
BookList
email : JTextField
bookingTable : JTable
search : JButton
back : JButton
search(): void
back() : void
FlightList
source : JTextField
destination : JTextField
flightNumber : JComboBox
typeOfClass : JComboBox
username : JTextField
confirm : JButton
back : JButton
search : JButton
confirm() : void
back() : void
search() : void
1..*
TicketInfo
flightNumber : JTextField
time : JTextField
date : JTextField
source : JTextField
destination : JTextField
ticketPrice : JTextField
confirm : JButton
back : JButton
confirm() : void
back() : void


Airlines Booking System
User
Login
extend
register
include
include
view booking list
book a flight
extend
include
include
include
view ticket information
choose type of class
choose flight number
search for a flight
database administrator
add flight
remove flight
add seats
update flight schedule
update time
update date

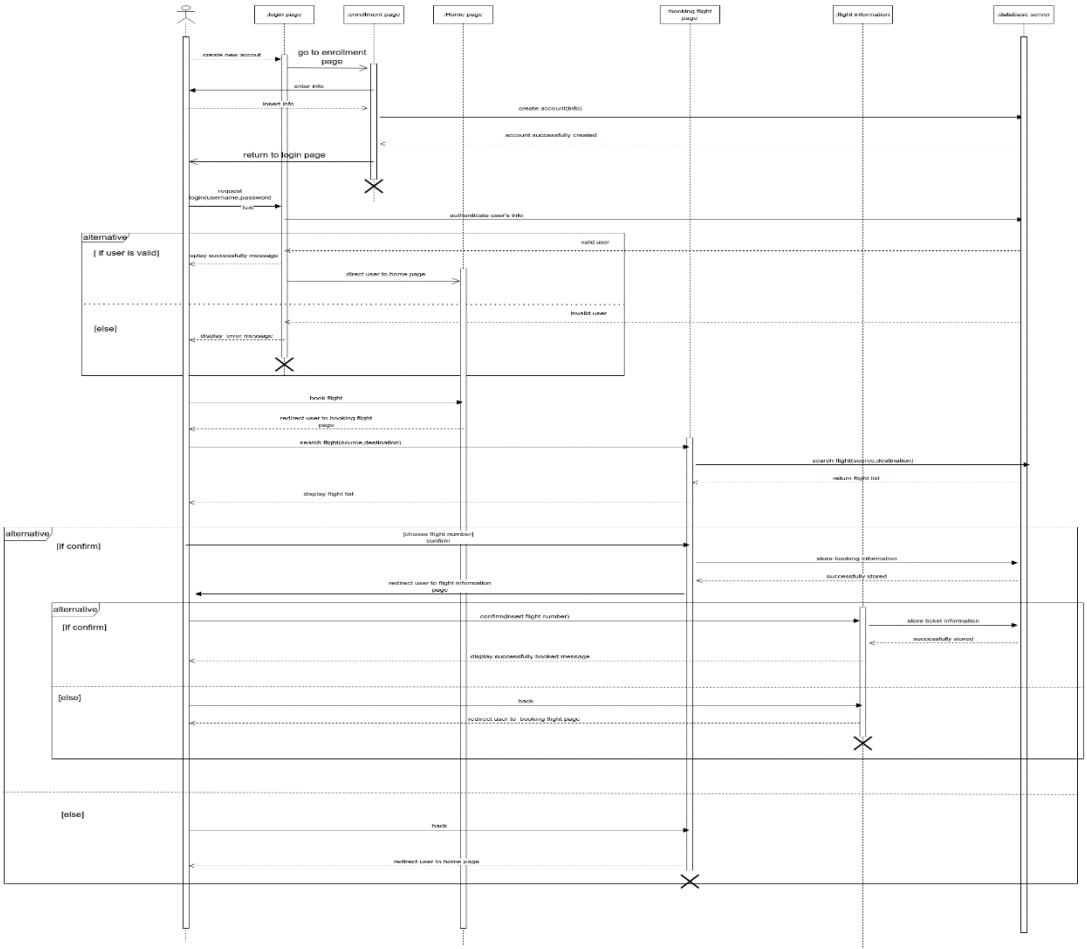
- **Login-log out – enrollment**



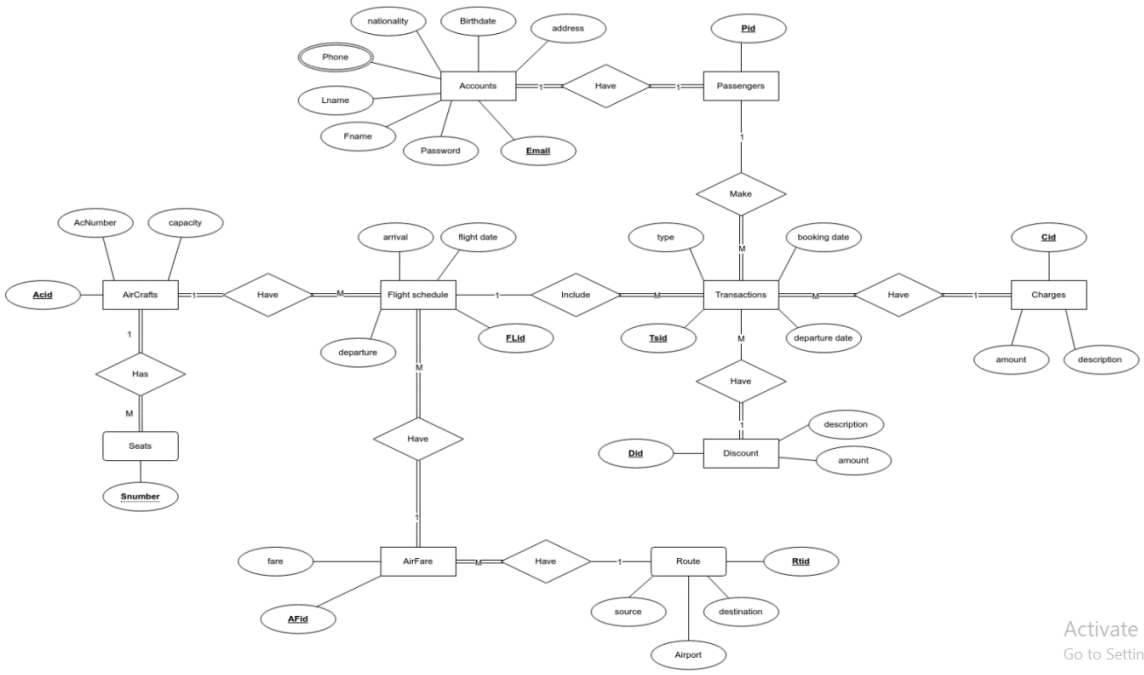
- **Booked list**



- **Book flight and flight information**


:login page
:enrollment page
:Home page
:booking flight page
:flight information
:database server
create new accout
go to enrollment page
enter info
insert info
create account(info)
account successfully created
return to login page
request login(username,password
hot
authenticate user's info
alternative
[if user is valid]
valid user
display successfully message
direct user to home page
invalid user
[else]
display error message
book flight
redirect user to booking flight page
search flight(source,destination)
search flight(source,destination)
return flight list
display flight list
alternative
[if confirm]
[choose flight number] confirm
store booking information
successfully stored
redirect user to flight information page
alternative
[if confirm]
confirm(insert flight number)
store ticket information
successfully stored
display successfully booked message
[else]
back
redirect user to booking flight page
[else]
back
redirect user to home page

- **ER diagram**



10. System Evolution

- The system should be able to work on different operating systems.
- It should work properly on devices with low specifications. Only the server on which it's installed will be powerful.