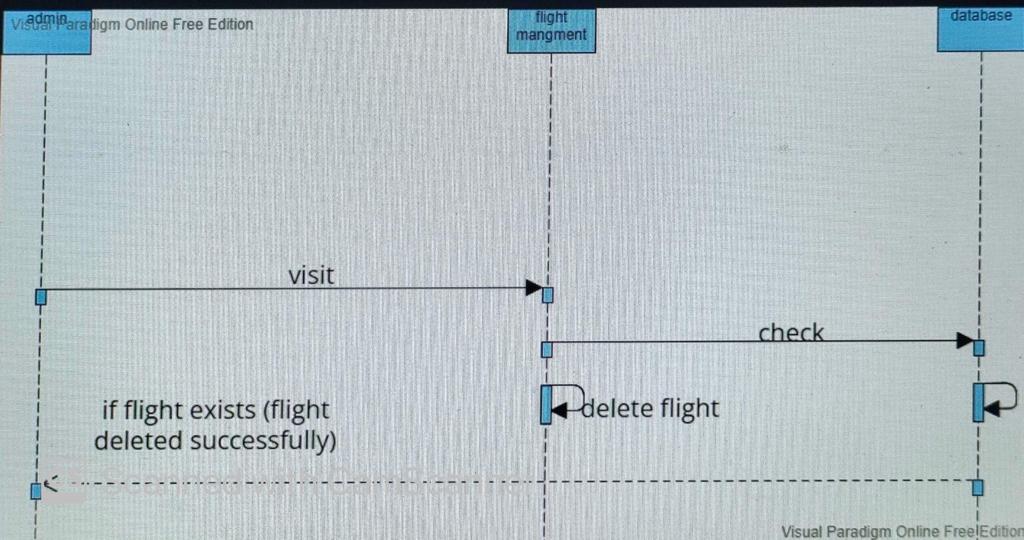
******

***Logo, company name

Description automatically generated*Software Requirements**

**Specifications**

UNDER THE SUPERVISION OF

**Dr. Walaa Medhat**

***Eng. Rameez Barakat***

|  |  |
| --- | --- |
| ***Name*** | ***ID*** |
| Sherif Ali Mahmoud | 202000658 |
| Nouran Hady Shaaban | 202001903 |
| Laura Mostafa Mohamed | 202001736 |
| Sama Ahmed Okasha | 202000452 |
| Omar Yahya Gamal El Deen | 202000259 |

**Table of Contents**

**1. Introduction4**

1.1 Purpose4

1.2 Scope4

1.3 Technologies used4

1.4 Methodology5

1.5 Intended Audience5

1.6 Overview5

2. Overall Description6

2.1 Product perspective6

2.2 User Characteristics6

2.3 Operating Environment6

2.4 Constraints6

2.5 Assumptions and Dependencies7

3. Interfaces8

3.1 System Interface 8

3.2 Software Interface9

4. Functional Requirements10

4.1 User Class 1- The user10

4.2 User Class 2- The Admin12

4.3 Payment Methods13

4.4 Guidelines 13

5. Non-Functional Requirements 13

5.1 Availability 14

5.2 Performance14

5.3 Maintainability14

6. Diagrams15

6.1 Use case diagrams15

6.1.1 Use case scenarios16

6.2 Class diagram29

6.3 Sequence diagram30

**1.Introduction**

**1.1 Purpose of the Air Line Reservation project:**

The project of the airline reservation system is used for many purposes, which are flight booking system, as the system allows the passengers to search and book a flight between two cities in a specific time, customer booking counters, pricing updates, hotels availability, offers available, tickets and much more. The project is built for the administration, thus only the administration has the access to all the information.

The purpose is to streamline the project using computerized equipment and a unique computer software that meets the requirements, so that the data can be saved for a longer period and easily accessed by management.

**1.2 Scope of the project:**

The project's primary goal will be to make airline reservations easier and automated with shorter reservation times while enhancing passenger interaction and engagement. Some features that will differentiate our project from any other one which are the database will be used to make the UX better. We will design a UI to make it user-friendly. This project is expected to be ready by the end of this course, between December 20, 2022, and December 30, 2022.

**1.3 Technologies Used:**

1. Coding Language: Java.

2- Frontend Implementation: GUI (Android Studio).

3- Backend: MySQL.

**1.4 Methodology:**

The development methodology adopted for the proposed system is Agile model.

**1.5 Intended Audience:**

**Developers**: To have ease of implementation.

**Software Tester**: To be able to develop the correct test cases for the system

and test it to improve the system's quality.

**Project Manager/SCRUM Master:** To be able to develop a good plan to work

on the project and construct the team.

**Project Supervisor:** To be able to mentor the teams’ work and assist with any

advices and make sure that the goals are met within the time allotted.

**Customer:** to follow up with project team if there is a requirement change.

**1.6 Overview of the document:**

In the upcoming chapter of the SRS documents the detailed information explaining the airline reservation system will be explained, where the exact requirements and user specification will be explained in very detailed manner. Chapter 3 of the SRS document will explain the implementation details of the interface. The functional and nonfunctional requirements, diagrams, and other pertinent requirements for the "Airline Reservation System" Application are then presented in the remaining chapters.

**2.Overall Description**

**2.1 Product Perspective:**

* The "Travelito" application is an identity and self-contained System with a satisfactory user interface.
* The system has two kinds of users The first type is the user, and the second type is the administrator. Each type has distinctive activities they can do with the system.
* The system makes use of a centralized database that contains all of the information. It is compatible with any Android phone running version 4.2 or higher.

**2.2 User Characteristics:**

There are two kinds of users for the airline reservation system. One is the customer and the other is the administrator.

* The customer should be comfortable with English language and have prior experience using general android applications or on the computer.
* The Administrators need to be trained to use the application.

**2.3 Operating Environment:**

The system will run on Android operating system version 4.2 or higher. Therefore, any devices that support this version of the android OS or higher can easily run the application anytime indoor or outdoor.

**2.4 Constraints:**

A working internet connection is a constraint for the system as all the data, passenger information and flight information, needed for the application to function is stored in a database which needs to be always accessible by the system. Thus, it is a must for there to be a stable internet connection for the application to operate.

**2.4.1 Software Constraints**

* Programming language used is Java in addition to the SDK.
* System uses MySQL for database.

**2.5 Assumptions and Dependencies:**

Users will download the application from their respective mobile app store.

* It is assumed that all users of the system have access to the internet.
* All users are assumed to understand English and be familiar with online booking applications.
* The system is dependent on an up-to-date database which stores all updated user, booking and flight information.

**3. Interfaces**

**3.1 System Interface:**

When a first-time user opens the mobile application, he or she should see the log-in page; see Figures 01&02. If a user is not yet registered in the system, he or she should be able to access the Sign-Up page. As a result, he or she will create an account to use the system's features.

Graphical user interface, application

Description automatically generatedA screenshot of a video game

Description automatically generated with medium confidence

Figure 01 Travelito App and Login Page

Figure 02 Sign Up Page

When the user opens the app on the mobile, he will see a log in page to open his account. If he doesn’t have one, he can enter to the sign-up page and create one or enter by his google or Facebook account. There is an option also if he forgot his password to create a new one.

**3.1 Software Interface:**

The Airline reservation app will be developed as a native mobile application for the Android operating system using the Java JDK and Android SDK tools. Android studio, as well as packages and libraries, are used. The following are the app development process's primary stages, followed by each stage's details.

The System interacts with the database on the server side. To run <MySQL> data queries. Within the System class, the username and password are set for MySQL﻿ login in the connection object.

Upon receiving login credentials from the User class, the database is connected by using the driver class that was constructed earlier. After successfully connecting to the Database, we retrieve the record that contains information about flights according to the customer’s input and compare it against our reference data. If a match is found, a message is sent to notify the user and the retrieved information is displayed to the user in a list format. Each flight is displayed on the screen with relevant information such as date, time, price, duration etc. The user may also add or delete flights from this list according to his preferences. Once the user is done with his booking, he can choose to complete the booking process by confirming the order or by paying via PayPal or with credit card. Once the payment is complete and the order is confirmed the system will notify the user of the flight details and the payment status.

**4. Functional Requirements**

**4.1 User Class 1- The User**

* + 1. **Functional Requirements**

**Title:** Confirmation messages.

**Desc:** Both a visual confirmation and a booking confirmation email should be sent by the system to the user's contact.

* + 1. **Functional Requirements**

**Title:** Help manger.

**Desc:** The user sends a message to the help center in case they face any problem while using the app and they respond to their request in 24 hours.

* + 1. **Functional Requirements**

**Title:** Sign Up.

**Desc:** User should be able to make a new account on the application.

* + 1. **Functional Requirements**

**Title:** Sign in.

**Desc:** User should be able to sign into his account using username and password

* + 1. **Functional Requirements**

**Title:** Search For flights.

**Desc:** User should be able to search for flights in the search bar by adding the destination and flight origin and he also should specify the data of departure and date of return.

* + 1. **Functional Requirements**

**Title:** Choose flight type.

**Desc:** User should have the optional choices that he can specify while searching for flights like adding the type of trip for example if it’s a round trip or a one-way trip and, he can add the number of passengers and their type like whether they are adults or children and, he can specify the cabin type that he prefers.

* + 1. **Functional Requirements**

**Title:** Show flight matching with the user’s requirements.

**Desc:** Once User clicks on search after specifying all the flight details system should show the user the flights that are matching with the user’s requirements and if not available display for the user flights that are close to his requirements and if no flight available at all display a message to the user that specifies that there are no flights available.

* + 1. **Functional Requirements**

**Title:** Cancel a reservation.

**Desc:** User should be able to cancel a reservation and register the information regarding her/his ticket including (confirmation no, name, date of journey, fare deducted).

* + 1. **Functional Requirements**

**Title:** Show given response to cancellation.

**Desc:** User should be able to see given response to reservation cancellation request.

* + 1. **Functional Requirements**

**Title:** Specify departure date.

**Desc:** User should be able to specify only departure date for one-way trips.

* + 1. **Functional Requirements**

**Title:** Specify departure and arrival dates.

**Desc:** User should be able to specify both departure and arrival dates for round trips

* + 1. **Functional Requirements**

**Title:** Select currency.

**Desc:** User should be able to select the currency while searching for a flight.

* + 1. **Functional Requirements**

**Title:** Pay using credit card.

**Desc:** User should be able to pay and will be asked to enter various credit card details including (credit card type, credit card holder name, credit card no and expiration date).

* 1. **User Class 1- The Admin**

**4.2.1 Functional Requirements**

**Title:** Add new flight.

**Desc:** Admin should be able to add new flights to the system.

**4.2.2 Functional Requirements**

**Title:** Modify existing flights.

**Desc:** Admin should be able to modify the details of existing flights.

* + 1. **Functional Requirements**

**Title:** See reservation cancellation.

**Desc:** Admin should be able to see reservation cancellation requests.

* + 1. **Functional Requirements**

**Title:** Accept reservation cancellation.

**Desc:** Admin should be able to accept reservation cancellation requests.

* + 1. **Functional Requirements**

**Title:** See details of existing bookings.

**Desc:** Admin should be able to see details of existing bookings.

* + 1. **Functional Requirements**

**Title:** Delete User.

**Desc:** Admin should be able to delete user.

* + 1. **Functional Requirements**

**Title:** Delete flight.

**Desc:** Admin should be able to delete a flight.

**4.3 Payment Methods**

* The system should accept payments made using a variety of methods, including PayPal, wallets, cards, and vouchers

**5. Non-Functional Requirements**

We examined and measured some quality attributes, which are listed below, based on the testing carried out on our system using the Android testing tool and Java.

**5.1 Availability**

* The system should be able to operate 24 hours a day, seven days a week.

**5.2 Performance**

* The app should be highly efficient that its loading time doesn’t exceed a limit (1 sec response time limit).

**5.3 Maintainability**

* The system can be updated later on and be added to he application after.

**6. Diagrams**

**Diagram

Description automatically generated6.1 Use case diagram**

**6.1.1 Use case scenarios**

Use case 01

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Sign up*** |
| ***Actors*** | ***User*** |
| ***Main success scenario:*** | * ***User clicks "Sign up" button once the system opens*** * ***System will display several fields and will prompt the user to fill them with the information such as full name, email, phone number and address etc...*** * ***User should enter the required information and press "Create account" button*** * ***System should validate the entered information*** * ***System should send a confirmation email to the user mobile number or email*** * ***User should follow confirmation steps sent to him*** * ***System should save the data of the user in the database once the user does the confirmation steps sent to him*** |
| ***Exceptions*** | * ***User already has an account so application pops up error message*** |

Use case 02

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Sign in*** |
| ***Actors*** | ***User*** |
| ***Main success scenario:*** | * ***System will prompt the user to enter his sign in information such as username and password*** * ***User enters his sign in information and presses "Sign in"*** * ***System checks the validity of the sign in information in the database*** * ***System confirms the validity of the information and redirects the user to use the offered features in the system*** |
| ***Exceptions*** | * ***User entered wrong credentials*** * ***User doesn’t have an account on the system*** |

Use case 03

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Sign out*** |
| ***Actors*** | ***User*** |
| ***Main success scenario:*** | * ***User clicks on sign out button in the System*** * ***System redirects the user to the sign in page*** |

Use case 04

|  |  |
| --- | --- |
|  | |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Search of a ticket for a flight*** |
| ***Actors*** | ***User*** |
| ***Main success scenario:*** | * ***After User has logged in successfully to the system*** * ***User chooses to book a flight option*** * ***System will prompt the user to enter flight details such as destination and flight type or passengers number and cabin type and origin of the flight and date of departure and data of arrival preferred*** * ***User enter the flight details correctly in the System*** * ***System searches for flights that match the user flight requirements in the data base*** * ***System displays the choices that match the user preferences in his flight*** * ***User navigates through the flights that the System showed to him and chooses preferred flight*** * ***User chooses more preferences through the preferences bar where he can choose how many transits he prefers and seating options and meals preferred and more options*** * ***System will search through the database for the updates preferences and what flights will match with the user preferences*** * ***System will display the updates flight results for the user to choose from them*** * ***User will choose his preferred flight*** |
| ***Alternate*** | ***user enters the flight details either by the required place, time of the flight, the day of the flight, and the system chooses the result based on this requires*** |
| ***Exceptions*** | ***there is no such flight available (either in that time or day)*** |
| ***Pre-condition*** | * ***User has logged in successfully*** * ***User must enter the details of the flight correctly*** |
| ***Post condition*** | ***System displays the results based on the users details if the flight*** |

Use case 05

|  |  |
| --- | --- |
|  | |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***viewing the ticket*** |
| ***Actors*** | ***User*** |
| ***Main success scenario:*** | * ***After User has logged in successfully to System*** * ***User clicks on the button called "My bookings"*** * ***System retrieve flight details from database*** * ***System displays the flight details on the screen containing origin and destination and departure date and arrival date and cabin type and number of passengers*** * ***User will click on the desired booked flight to see furthermore details*** * ***System will display further details for the chosen flight which are the departure airport and arrival airport and meals served in the flight and an estimation of the flight duration and whether the arrival country requires any vaccinations*** |
| ***Pre-condition*** | * ***User has logged in successfully*** * ***User must have searched for the ticket of the flight correctly*** |
| ***Post condition*** | ***System displays the ticket*** |

Use case 06

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***booking the ticket*** |
| ***Actors*** | ***User*** |
| ***Main success scenario:*** | * ***User searched for the desired flight and chose the preferred flight*** * ***user clicks on the "Book" button*** * ***System will prompt the user to confirm if the flight chosen is the exact desired flight and will display all flight details*** * ***user will press "Confirm" button*** * ***System will redirect the user to another page which will prompt the user for details of the passengers such as full name, number, passport number etc...*** * ***User will enter his information correctly and clicks "Done"*** * ***System confirms the information validity*** * ***System sends a confirmation mail that the ticket is booked and add the flight to "My bookings" section in the user profile*** * ***System will update the database by decreasing the quantity of tickets available in the flight that the user chose and will place the flight in the booked flights section in the database*** |
| ***Exceptions*** | ***User didn't fill all the required fields or fills with false information*** |
| ***Actions*** | ***System displays an error message to enter of the required fields*** |
| ***Pre-condition*** | * ***User has logged in successfully*** * ***User must have searched for the ticket of the flight correctly*** * ***User must have viewed the ticket*** |
| ***Post condition*** | ***System send an email and a message that they have booked the required ticket*** |

Use case 07

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Cancel Reservation*** |
| ***Actors*** | ***User, System, database*** |
| ***Main success scenario:*** | * ***User clicks on "My bookings" button*** * ***System loads the user flights from the database and displays them to the user*** * ***User should enter the required information and press "Create account" button*** * ***User goes through the flights and clicks on the flight he need to cancel*** * ***System will load the exact details of the flight the user chose*** * ***User clicks on "Cancel flight reservation” button on the top right corner in the system*** * ***System will display refund guidelines for the user to explain how he'll receive his money back in case of online payment*** * ***User should click "agree button" on the bottom left of the guidelines pop up message*** * ***System displays a confirmation message that says "Are you sure you want to cancel your flight reservation"*** * ***User should click "Yes" button*** * ***System should process the cancellation and send the user a cancellation confirmation email through the mobile or email containing the flight canceled and the refund details*** * ***System should update the database by increasing the number of tickets available in this flight and removing the flight the user canceled from user bookings*** |

Use case 08

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Select currency*** |
| ***Actors*** | ***User, system, database*** |
| ***Main success scenario:*** | * ***User signs into the system using sign in information*** * ***User clicks on "My profile" button*** * ***User clicks on "Change currency" button*** * ***System will load from the database the available currencies*** * ***User will choose the preferred currency from a list of currencies that the system will display to the user*** * ***User clicks on "save" button after choosing preferred currency*** * ***System will update the currency that the flight prices are displayed with.*** |

Use case 09

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Add new flight*** |
| ***Actors*** | ***Admin, system, database*** |
| ***Main success scenario:*** | * ***Admin chooses to add new flight option*** * ***Admin enters all required flight information correctly*** * ***Admin clicks on "Add flight" button once finished filling all the fields*** * ***System adds new flight to the database and displays to the admin message "Flight added successfully"*** |
| ***Exceptions*** | * ***Admin did not enter all required information or entered false information.*** |

Use case 10

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Modify existing flights*** |
| ***Actors*** | ***Admin, system, database*** |
| ***Main success scenario:*** | * ***Admin clicks on "Manage flights" button*** * ***System loads the flight from the database and displays them to the admin*** * ***admin chooses flight to modify information of it*** * ***System displays flight's current information*** * ***Admin modifies information that he needs to edit*** * ***Admin clicks "Save" button*** * ***System updates flight information in the database*** |

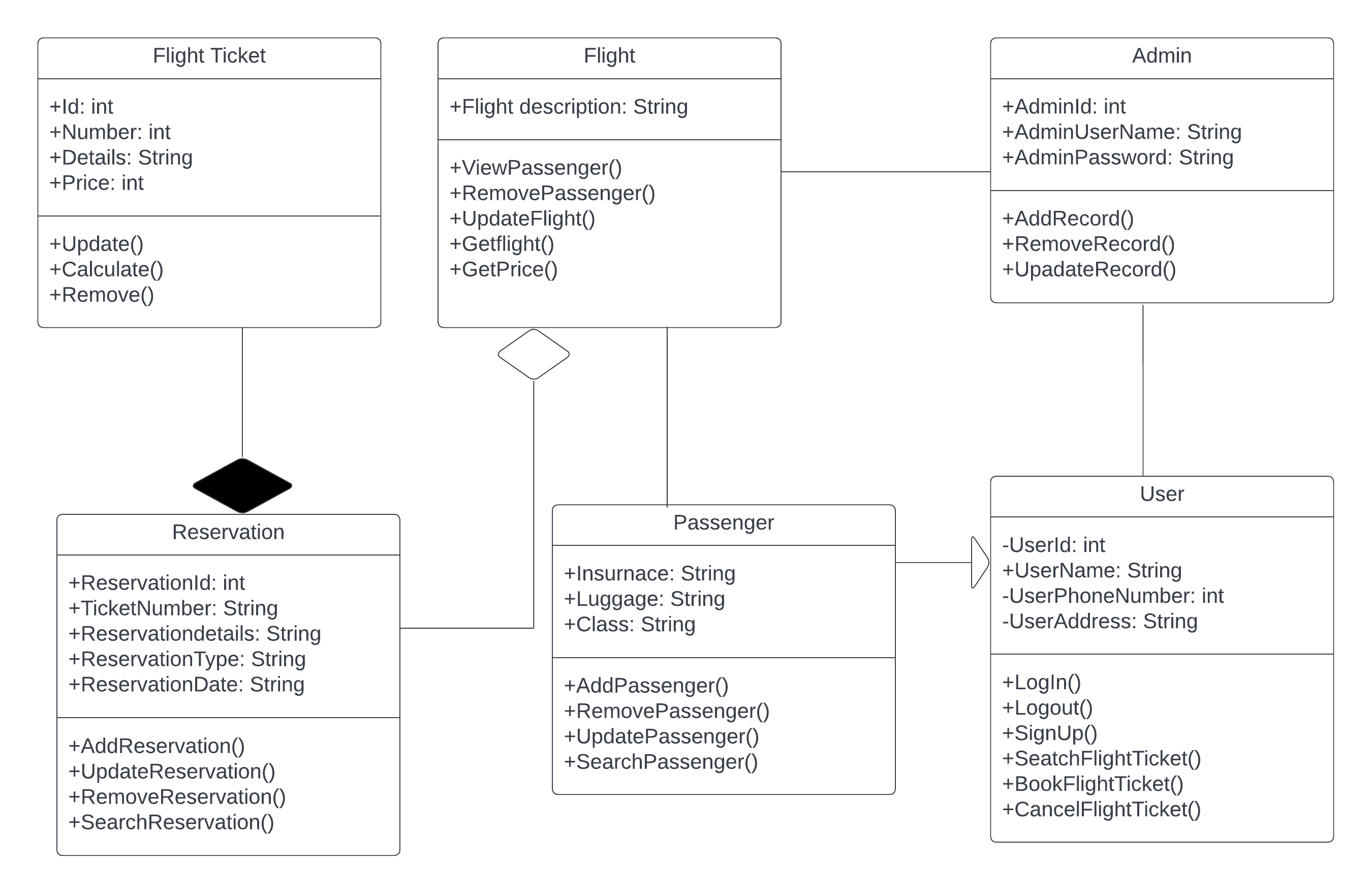
Use case 11

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Delete user*** |
| ***Actors*** | ***Admin, system, database*** |
| ***Main success scenario:*** | * ***Admin clicks on "Manage users" button*** * ***System loads the users from the database and displays them to the user*** * ***Admin chooses from the users list loaded the one he wants to delete*** * ***System loads the chosen user information*** * ***Admin clicks on "Delete user" button on the top right of the system*** * ***System will remove the user from the database*** * ***System will display message "User deleted successfully"*** |

Use case 12

|  |  |
| --- | --- |
| ***Shape  Description automatically generated with low confidence*** | |
| ***Use-case Name*** | ***Delete flight*** |
| ***Actors*** | ***Admin, system, database*** |
| ***Main success scenario:*** | * ***Admin clicks on "Manage flights" button*** * ***System loads the all the flights from the database and displays them to the Admin*** * ***Admin will choose the exact flight he wants to delete*** * ***Admin clicks on "Delete flight" button on the top right corner in the system*** * ***System displays a confirmation message that says "Are you sure you want to delete this flight"*** * ***Admin should click "Yes" button*** * ***System will prompt from the Admin a reason for deletion of the flight that will be provided as the reason of the forced cancellation of the flights for the users who had already booked this flight*** * ***Admin will enter the exact reason*** * ***System should process the cancellation and display a message "Flight deleted successfully"*** * ***System should send to all the users that booked this flight before it got deleted an email explaining that the flight was deleted due to the reason that the Admin provided*** |

**6.2 Class Diagram**



**6.3 Sequence Diagram**

**A picture containing text, map, indoor

Description automatically generated**

