Saarthak Jain, Samar Saini

2K18/CO/306, 2K18/CO/312

SUBMITTED TO:

Dr. ROHIT BENIWAL



OOSE PROJECT REPORT

AIRLINE RESERVATION SYSTEM

**DELHI TECHNOLOGICAL UNIVERSITY**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| S. No | Experiment Name | Page number |
| 1 | Problem statement of the case study. | 2 |
| 2 | Elicitation techniques for different domains. | 3 |
| 3 | Initial requirement document of the case study. | 5 |
| 4 | Software Requirement Specification document of the case study. | 8 |
| 5 | Use case diagram of the case study. | 16 |
| 6 | Use case description of the case study. | 17 |
| 7 | Class diagram of the given case study. | 21 |
| 8 | Sequence diagram of the given case study. | 22 |
| 9 | Collaboration diagram of the given case study. | 26 |
| 10 | Activity diagram of the given case study. | 33 |
| 11 | Test cases/ Test report of the given case study. | 34 |

**PROBLEM STATEMENT:**

* The purpose of this document is to build an online system to manage flights and passengers to ease the flight management.
* Users of the system should be able to retrieve flight information between two given cities with the given date/time of travel from the database. The system will support two types of user privileges, Customer, and Employee.
* Customers will have access to customer functions, and the employees will have access to both customer and flight management functions. The customer should be able to make a flight reservation for either one-way trip or a round trip on a particular date. He will be asked for the confirmation before making his reservation. Customer will also be able to cancel the existing reservation and can also view the details of already reserved flights.
* On the other hand, Employee can access details of all the customers as well as of the flights, view flight schedule and calculate sales for a given flight. The administrator access will allow an employee to add, delete and update the information. He will be able to add/delete a flight, update fare for flights and update arrival/departure time of flights. Each flight has a limited number of available seats. There are a number of flights which depart from or arrive at different cities on different dates and time.

**PROCESS MODEL:**

For the development process of this system, we have chosen the *Waterfall model*.

We decided to go with this model because:

* Requirements are very well documented, clear and fixed at the start of the software making.
* There are no ambiguous requirements.
* Technology used in making this software is not new and well understood by the developers.
* Since it is not a real project to be used in the market, we can use Waterfall model

**REQUIREMENT ELICITATION:**

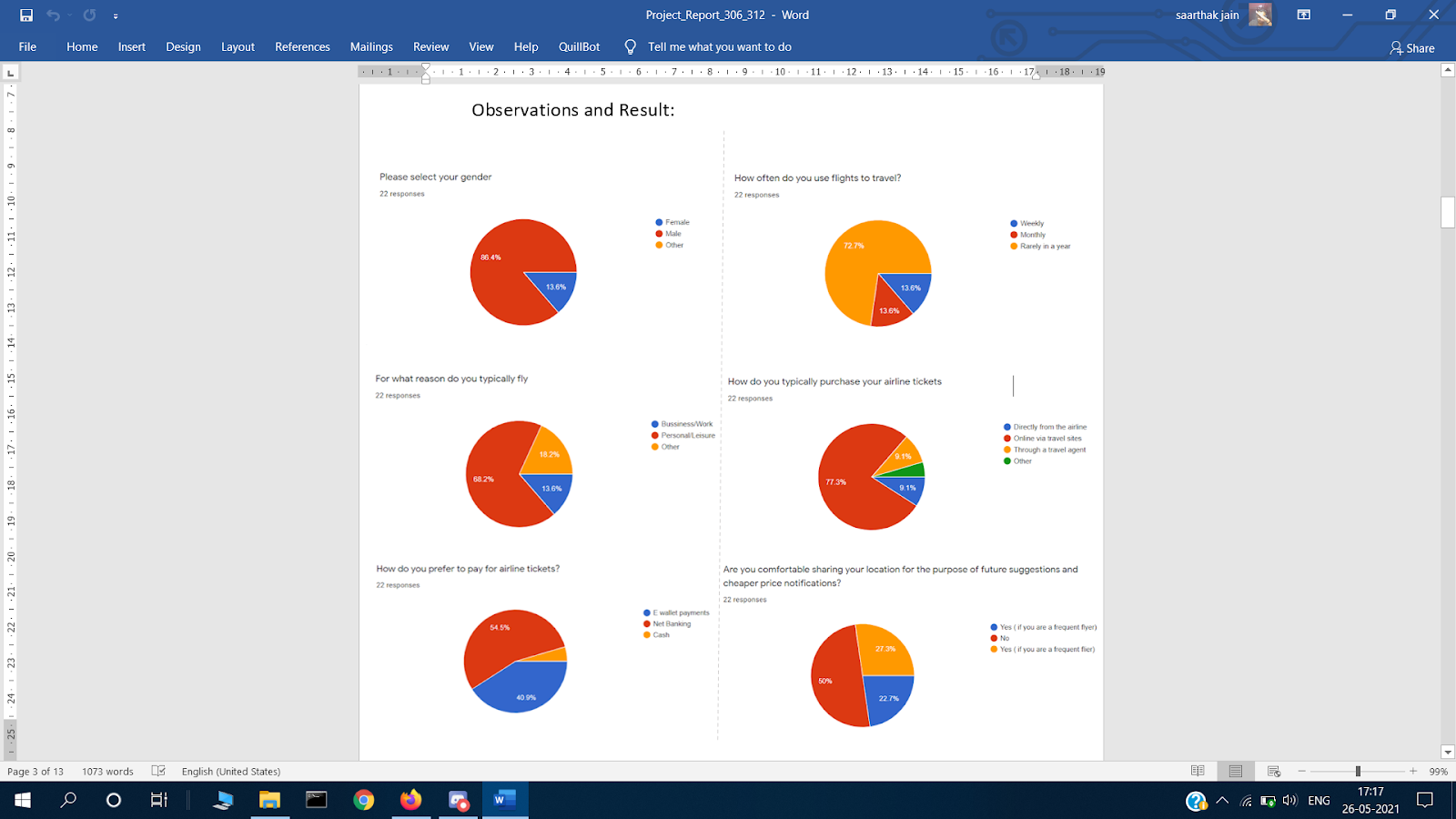
1. **SURVEY:**

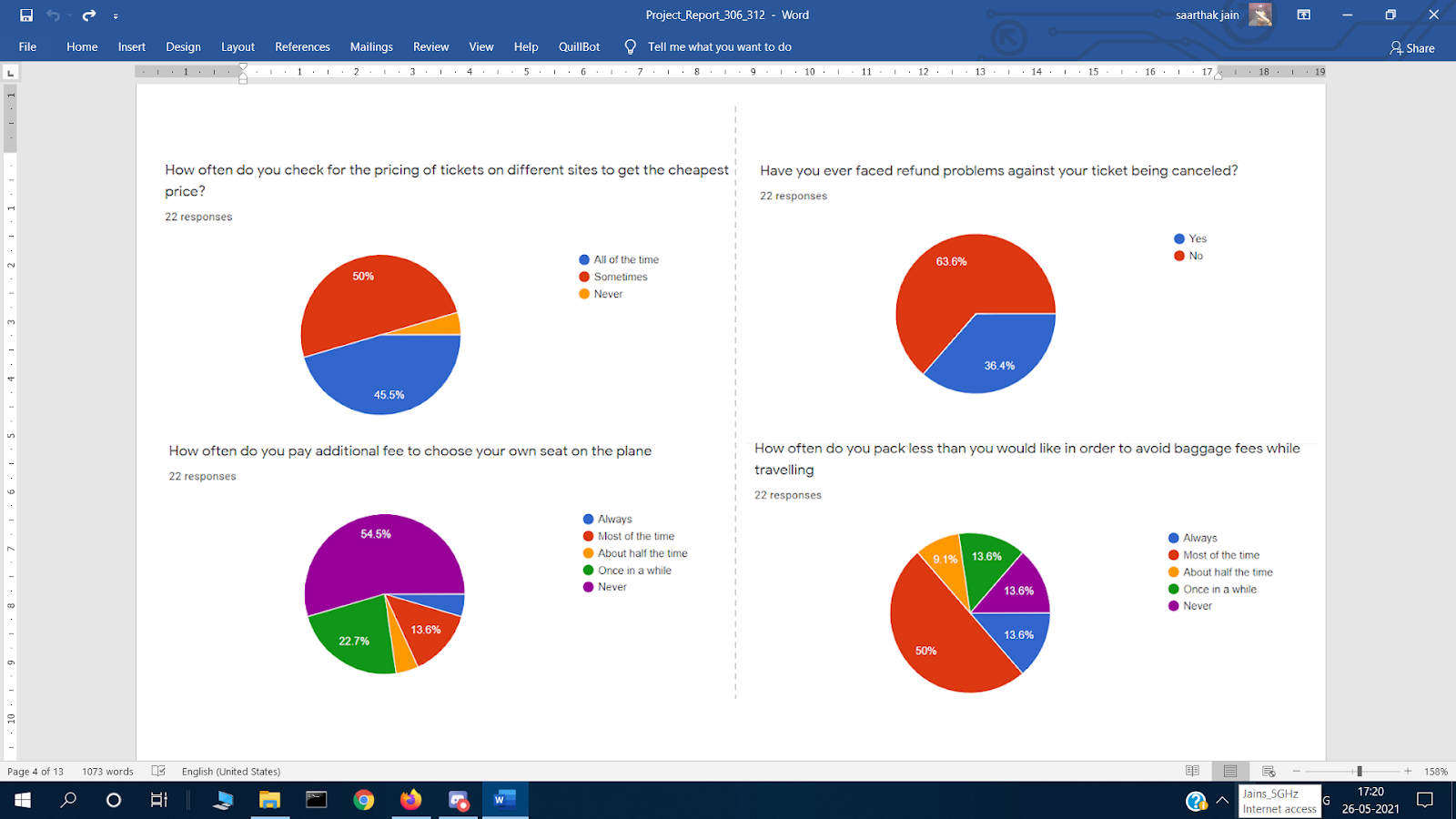
The survey was carried out among the stakeholders and was conducted to gather information regarding their preference in various online airline reservation systems and how often they consider the online booking system.

Survey Link:

[https://docs.google.com/forms/Airline\_Reservation\_System](https://docs.google.com/forms/d/1BeIQxJVGMfWdYdSGRi3mu1bEAuNyGt5PaQG73IptL7A/edit?ts=5f79c902&gxids=7757)

Observations and Result:





Approximately 77% of people prefer online reservations. But due to some trust issues, people use other methods for booking. The major concern for an online airline reservation system is secure booking, smooth cancellations, and genuine price for traveling. People taking part in the survey prefer net banking for payment, mostly people use flight once in a year for personal reasons. People generally don’t pay extra charges for seat preference as well as for extra luggage.

**Initial Requirement Document**

## Introduction:

Airline Reservation Systems (ARS) used to be standalone systems. Each airline had its own system, disconnected from other airlines or ticket agents, and usable only by a designated number of airline employees. Travel agents in the 1970s pushed for access to the airlines' systems. Today, air travel information is linked, stored, and retrieved by a network of Computer Reservations Systems (CRS), accessible by multiple airlines and travel agents.

## Goals:

This proposal is for performing a case study to be undertaken to design and implement an airline reservation system. Reviewing the literature of reservation systems and exploring the advantages and limitations of reservation systems in real-life situations will also be a part of the case study.

## Specifications:

Interviews and questionnaire methods will be a major part of the data collection phase, these data collection methods will help to better understand the existing system in use. Case tools and data flow diagrams are intended to be used during the development process to simulate the process of airline reservation and ticket booking. The case study will also contain ER diagrams and SRS for the system.

## Milestones:

The current system is manual, this system is slow, time-consuming and it is very difficult for each person to book through office agents. Users inquire about the tickets through phones, and it is tough for the user to remember all the details that they received through phones. It is very difficult to calculate how many people registered and how many seats on a particular plane are vacant. This requires quite a lot of time and wastage of money as it requires quite a lot of manpower to do. The general objective is to automate the process of airline ticket reservation, booking, and airline management hence minimize errors resulting from manual system operations.

## Scope of Study:

Minimize repetitive work done by the system administrator and reservation clerks. Maintain consistency among different access modes, e.g. by phone, by web, at the information desk, and across different physical locations. In addition, the outcome of this study will provide a basis for developing the appropriate approach to the problems associated with air traveling operations in relation to Airline Reservation Systems.

## Specific Objectives:

* To study the current system identifying its inefficiencies.
* To design an online airline reservation information system to facilitate online booking and flight schedules.
* To determine both functional and non-functional requirements for the new system.
* To provide a user-friendly interface.
* To implement the developed web-based airline information system.
* To test and validate the developed system by use of a case study.
* To provide flexibility and more security in the payment transaction.
* To provide some additional features as per requirements.
* The system should minimize repetitive work done by the system administrator and reservation clerks.
* The system should maintain customer information in case of emergency, e.g., flight cancellation due to inclement weather.

## Functional Requirements:

**1. User registration:**

This case describes the scenario where the user registers with the application by providing all the necessary details in order to make reservations, motels, special packages.

## 2. User login:

In this user logs into the application with the username and password he has provided during registration.

## 3. Contact the company:

Users can contact the company for any info.

## 4. Booking Instructions:

Users can view the instructions for booking flights.

## 5. Booking packages:

Users can either book a two-way trip or a one-way trip at the time of reservation.

## 6. Login/Logout:

Administration of the application logs into the system and logout after the work is done.

## 7. Add/Delete/modify customer info:

Admin adds, deletes, or modifies customer info in the system database.

## 8. Cancellation of reservation:

Admin handles the cancellation of reservations by the customers.

## 9. Email confirmation:

Admin sends an email confirmation to the customers of the application.

## Non-functional requirements:

**Performance Requirements**

Queries filed by users should be responded quickly by the ARS. When a user looks for a flight departing from one airport to another airport, the ARS should quickly return the results. As the ARS application is not that big, it should show six-seven results at a time on each page to the user, when he/she searches for some data. Time taken by the application to answer the requests of users should be less than two and a half seconds.

**Safety Requirements**

If a large chunk of the database suffers significant harm due to accidental failure, for example a storage device crash, a recovery technique should be present that recovers a former copy of the database that was backed up to backed-up storage and recreates a more present state by reapplying or repeating the operations of committed transactions from the backed-up log, up until the moment of failure.

**Security Requirements**

Security systems, like so many other systems, require storage in the form of database. But, due to the unique needs of the security sector, suppliers must hand pick their database partner.

**Software Quality Attributes**

* **AVAILABILITY of system:** Users should be able to access application on any time of the day for booking or cancelling the tickets.
* **CORRECTNESS of system:** The flights available on application should depart from the correct terminal and also arrive at the proper destination.
* **MAINTAINABILITY of system:** Correct schedules of flights should be maintained by the administrators and flight management crew.
* **USABILITY of system:** Flight itineraries should be designed to meet the requirements of as many consumers as possible.

SYSTEM REQUIREMENTS SPECIFICATIONS

FOR THE

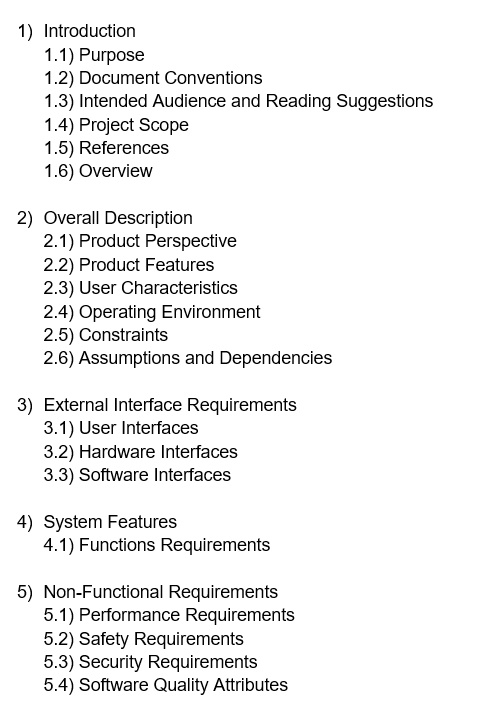
<AIRLINE RESERVATION SYSTEM>

By:

Samar Saini (2K18/CO/312)

Saarthak Jain (2K18/CO/306)

TABLE OF CONTENTS



1. **INTRODUCTION**

**1.1 PURPOSE**

The goal of this article is to create an online platform for handling flights as well as travellers in order to make flight management easier.

**1.2 DOCUMENT CONVENTIONS**

The mentioned conventions are used in this document:

* ARS: Airline Reservation System
* ERD: Entity Relationship Diagram
* DBMS: Database Management System

**1.3 INTENDED AUDIENCE AND READING SUGGESTIONS**

This whole project is a model for a flight management system that is confined to the university premises. This is expected to be beneficial for both the flight managing crew as well as the passengers.

**1.4 PROJECT SCOPE**

The goal of said online ARS is to make flight management much easier and also to build a comfortable and user-friendly interface for customers attempting to purchase air tickets. Including its flight management and booking functionalities, the platform is constructed on a relational database. We will have a database system that would accommodate hundreds of major locations worldwide along with thousands of flights from numerous airlines. And thus, we want to ensure a delightful customer experience as well as an efficient and fast system for the management.

**1.5 REFERENCES**

**Project Report:** Contains the different views of the system using Sequence Diagrams, Activity Diagrams etc. and the details of the requirement elicitation for the development.

* 1. **OVERVIEW**

Section 2 of this document describes overview of system in terms of general characteristics of the system, information about possible users of the system, possible constraints on the system, operating environment of the system etc. The Section 3 describes in detail, the different interfaces and their requirements. The Section 4 describes the system features and their relationships in brief. Finally, the Section 5 discusses about the non-functional requirements like safety, performance, etc requirements.

## OVERALL DESCRIPTION

**2.1 PRODUCT PERSPECTIVE**

The following records will be maintained within a distributed airline DBMS:

* **Flight** **Details**   
  This comprises the source and destination flight terminals, the schedule, total number of seats reserved or vacant for each class, and so on.
* **Customer/Passenger Details**   
  This comprises the information like customer id, name, address, phone number and so on which may be used to retain the customer's records in case of an emergency or for any other purpose.
* **Reservation Details**   
  This contains the passenger information, a PNR numbers, flight numbers, the date of reservation, the date of travel, billing information, and so forth.
* **Employee Details**

This contains the details of the flight management crew like name, ids, phone numbers, etc.

**2.2 PRODUCT FEATURES**

The ARS supports following functions:

* Functions by which customers can book or cancel an online ticket for a particular date, place, class and check the ticket status.
* Functions by which employees get the status of all customers and flights.
* Functions by which employee can add/delete a flight.
* Functions by which employee can update a flight’s information/status like updating the departure/arrival time of a flight, etc.

The access to these different functions by different users is restricted.

**2.3 USER CHARACTERISTICS**

Users of the system will be permitted to get flight info connecting two specified locations on the specified date and time of travel. Customer and Employee user privileges will be supported by the platform. Customers will be able to use customer services, while staff will be able to access both customer and flight management capabilities

.

The Customer should be able to do the following functions:

* Make a new reservation based on availability  
  • Flexible Date/time and other requirements  
  • Payment and Confirmation
* Cancel an existing reservation
* View his itinerary

The Employee should have following management functionalities:

* Customer Functions:  
  • Get all the passengers and their details, who have seats reserved on a given flight.   
  • Get all flights and their details for a given airport and time.
* Administrative Functions:  
  • Add and Delete a flight  
  • Update fare for flights.  
  • Update departure/arrival.

As a result, each trip will have a fixed number of available seats in various classes, and there are a number of flights that depart or arrive at various places on various days and times.

**2.4 OPERATING ENVIRONMENT**

Operating environment for the airline management system is as listed below.

* Distributed database
* Operating system: Windows.
* Database: SQL+
* Platform: To be decided

**2.5 CONSTRAINTS**

User is required to remember the login ID and password for his account. If lost, user will not be able to access his account again. Customers will be allowed to check the ticket status and book ticket only through internet. Payment for the tickets can only be made through net banking.

**2.6 ASSUMPTION AND DEPENDENCIES**

As of now, there are no assumptions. This article will be updated in future editions.

* + 1. **EXTERNAL INTERFACE REQUIREMENTS**

**3.1 USER INTERFACES**

The ARS project has two sorts of users. The first is the customer, while the second is the employee/administrator. Both Customer and Employee user interfaces would be graphical. The following software is being used for functionalities:

* Front-end software is not yet decided
* Back-end software we are using is SQL+

**3.2 HARDWARE INTERFACES**

The machines should be running Windows and also have a browser that supports CGI, HTML, and JavaScript. For interactivity, the computer must communicate with a conventional output device, a keyboard, and a mouse.

**3.3 SOFTWARE INTERFACES**

The programme should be compatible with the Windows operating system.  
SQL+ is required because the programme requires a database to hold all of the client information, airline information, and reservation information.

The decision for the coding language to implement the coding of the software is yet to be made. It will be either C++/Python/PHP.

Accordingly, we will decide the IDE to work on.

* + 1. **SYSTEM FEATURES**

**4.1 FUNCTIONAL REQUIREMENTS**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Function Name:** Login

**Description**: This function describes how a user logs into the system.

**Basic flow**: User wishes to log in to the system and has to enter valid credentials necessary to log in to the system

**Alternate flow**: Credentials entered by the User do not follow the format prescribed for the credentials.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Verify Password

**Description:** Credentials entered are verified against the database

**Basic flow:** Entered credentials match that of the ones present in the database

User is logged in to the system in their respective account type.

**Alternate flow:** If the entered credentials are not present in the database, an error message is generated.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Check Availability

**Description:** Customers can check for tickets based on their requirements

**Basic flow:** Customers can search for tickets based on the following parameters:

* Date and Time
* Location
* Class

**Alternate flow:** None

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Book Ticket and Payment

**Description:** Provides the main payment functionality that allows Customers to book and pay for the tickets.

**Basic flow:** Customer selects the available tickets and initiates the payment functionality after which verification takes place.

**Alternate flow:** If the payment is unable to be completed due to any reason, a transactional error message is generated.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Cancellation

**Description:** Customers can initiate the Cancel functionality if they wish.

**Basic flow:** The Customer is given the option to confirm cancellation after which they are refunded and the system records are updated.

**Alternate flow:** None

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** View Itinerary

**Description:** Customer can view their flight status, details, etc.

**Basic flow:** Customers can now view their flight itinerary which includes the departure and arrival airports, dates and times of the flights, flight numbers, etc.

**Alternate flow:** None

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** View Customer/Flight Information

**Description:** Allows Employees to access customer and flight information from the database.

**Basic flow:** Employees can search for information using customer id, flight id, etc depending on their needs.

**Alternate flow:** None

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Add Flight

**Description:** Allows Employees to add new flights and their respective information

**Basic flow:** The employee can add a new flight along with their respective information like departure and arrival times, fares, seats, etc.

**Alternate flow:** None

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Update Flight Information

**Description:** Allows employees to update information of existing flights.

**Basic flow:** Employees can search for the flight and perform the following:

* Update Arrival/Departure
* Update Fares
* Delete Flight

**Alternate flow:** If a flight does not exist in the database, an error message is generated.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. **NON-FUNCTIONAL REQUIREMENTS**

**5.1 PERFORMANCE REQUIREMENTS**

Queries filed by users should be responded quickly by the ARS. When a user looks for a flight departing from one airport to another airport, the ARS should quickly return the results. As the ARS application is not that big, it should show six-seven results at a time on each page to the user, when he/she searches for some data. Time taken by the application to answer the requests of users should be less than two and a half seconds.

**5.2 SAFETY REQUIREMENTS**

If a large chunk of the database suffers significant harm due to accidental failure, for example a storage device crash, a recovery technique should be present that recovers a former copy of the database that was backed up to backed-up storage and recreates a more present state by reapplying or repeating the operations of committed transactions from the backed-up log, up until the moment of failure.

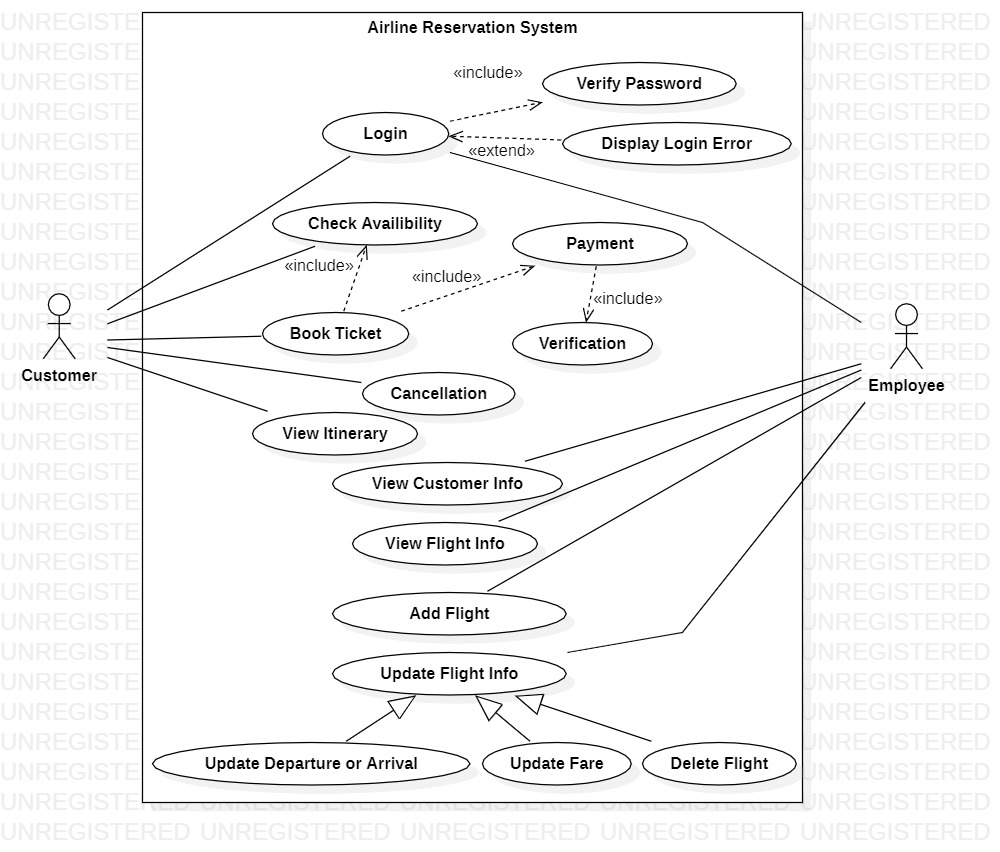
**5.3 SECURITY REQUIREMENTS**

Security systems, like so many other systems, require storage in the form of database. But, due to the unique needs of the security sector, suppliers must hand pick their database partner.

**5.4 SOFTWARE QUALITY ATTRIBUTES**

* **AVAILABILITY of system:** Users should be able to access application on any time of the day for booking or cancelling the tickets.
  + **CORRECTNESS of system:** The flights available on application should depart from the correct terminal and also arrive at the proper destination.
  + **MAINTAINABILITY of system:** Correct schedules of flights should be maintained by the administrators and flight-in chargers.
* **USABILITY of system:** Flight itineraries should be designed to meet the requirements of as many consumers as possible.

### **USE CASE DIAGRAM**



### **USE CASE DESCRIPTION**

**Function Name:** Login

|  |  |
| --- | --- |
| Brief | This use case describes how a user logs into the system |
| Actors | Customer/Employee |
| Precondition | Actors should have already had an existing account of their respective type |
| Basic flow | Actor wishes to log in to the system and has to enter valid credentials necessary to log in to the system |
| Alternate flow | Credentials entered by the Actor do not follow the format prescribed for the credentials. Actor is returned to the beginning of the Use case. |
| Post condition | Verification of credentials |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Verify Password

|  |  |
| --- | --- |
| Brief | Credentials entered are verified against the database |
| Actors | None |
| Precondition | Valid credentials should be present in the database |
| Basic flow | Entered credentials match that of the ones present in the database  User is logged in to the system in their respective account type |
| Alternate flow | If the entered credentials are not present in the database, an error message is generated. |
| Post condition | Customer/Employee is allowed access to his account |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Check Availability

|  |  |
| --- | --- |
| Brief | Customers can check for tickets based on their requirements |
| Actors | Customer |
| Precondition | The Customer must be logged in. |
| Basic flow | Customers can search for tickets based on the following parameters:   * Date and Time * Location * Class |
| Alternate flow | None. |
| Post condition | Customers can now book the tickets if they want. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Book Ticket and Payment

|  |  |
| --- | --- |
| Brief | Provides the main payment functionality that allows Customers to book and pay for the tickets. |
| Actors | Customer |
| Precondition | The Customer must be logged in and must choose the appropriate available tickets based on their need. |
| Basic flow | Customer selects the available tickets and initiates the payment functionality after which verification takes place. |
| Alternate flow | If the payment is unable to be completed due to any reason, a transactional error message is generated. |
| Post condition | Customer receives the booking confirmation and can log out of the system. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Cancellation

|  |  |
| --- | --- |
| Brief | Customers can initiate the Cancel functionality if they wish |
| Actors | Customer |
| Precondition | Customer has already booked and paid for tickets. |
| Basic flow | The Customer is given the option to confirm cancellation after which they are refunded and the system records are updated. |
| Alternate flow | None |
| Post condition | User receives the confirmation and can log out of the system. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** View Itinerary

|  |  |
| --- | --- |
| Brief | Customer can view their flight status, details, etc. |
| Actors | Customer |
| Precondition | Customer must be logged in and the payment for the tickets is completed. |
| Basic flow | Customers can now view their flight itinerary which includes the departure and arrival airports, dates and times of the flights, flight numbers, etc. |
| Alternate flow | None |
| Post condition | None |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** View Customer/Flight Information

|  |  |
| --- | --- |
| Brief | Allows Employees to access customer and flight information from the database. |
| Actors | Employee |
| Precondition | Employee must be logged in. |
| Basic flow | Employees can search for information using customer id, flight id, etc depending on their needs. |
| Alternate flow | None |
| Post condition | If the entity exists, the employee can now access it. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Add Flight

|  |  |
| --- | --- |
| Brief | Allows Employees to add new flights and their respective information. |
| Actors | Employee |
| Precondition | Employee must be logged in. |
| Basic flow | The employee can add a new flight along with their respective information like departure and arrival times, fares, seats, etc. |
| Alternate flow | None. |
| Post condition | Database is updated according to the changes. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Function Name:** Update Flight Information

|  |  |
| --- | --- |
| Brief | Allows employees to update information of existing flights. |
| Actors | Employee |
| Precondition | Employee must be logged in. |
| Basic flow | Employees can search for the flight and perform the following:   * Update Arrival/Departure * Update Fares * Delete Flight |
| Alternate flow | If a flight does not exist in the database, an error message is generated. |
| Post condition | Database is updated according to the changes. |

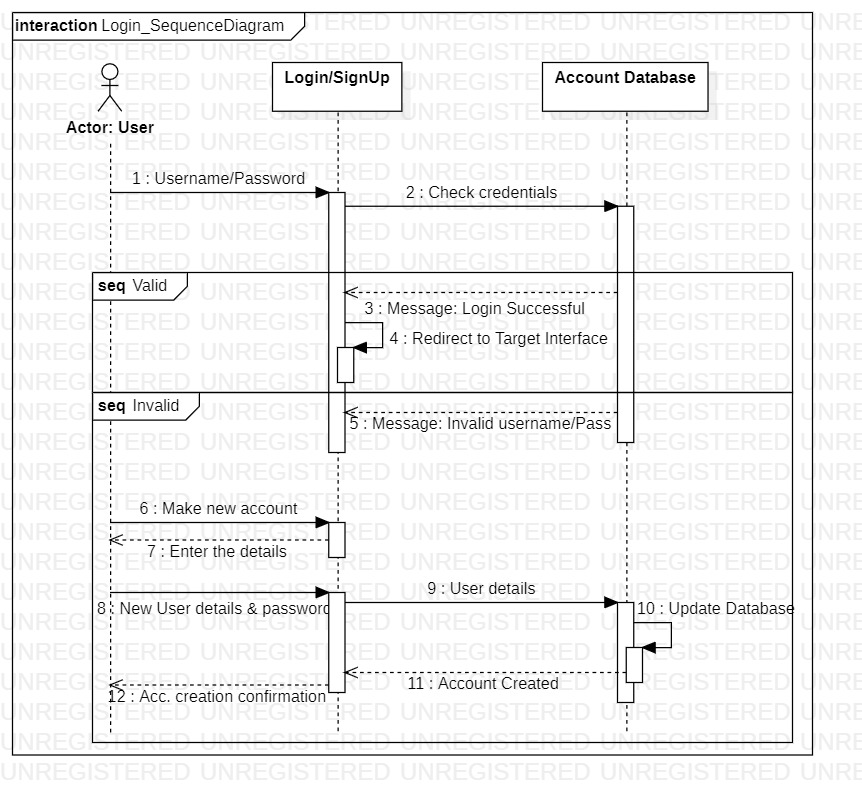
### **CLASS DIAGRAM**



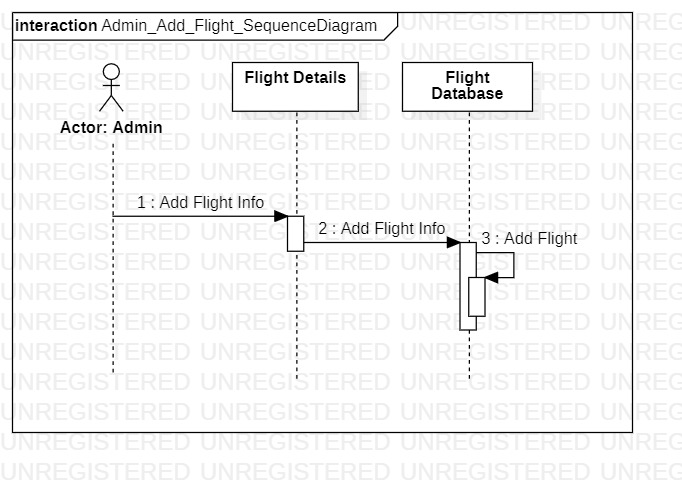
### **SEQUENCE DIAGRAMS**

All the sequence diagrams of our project are given below:

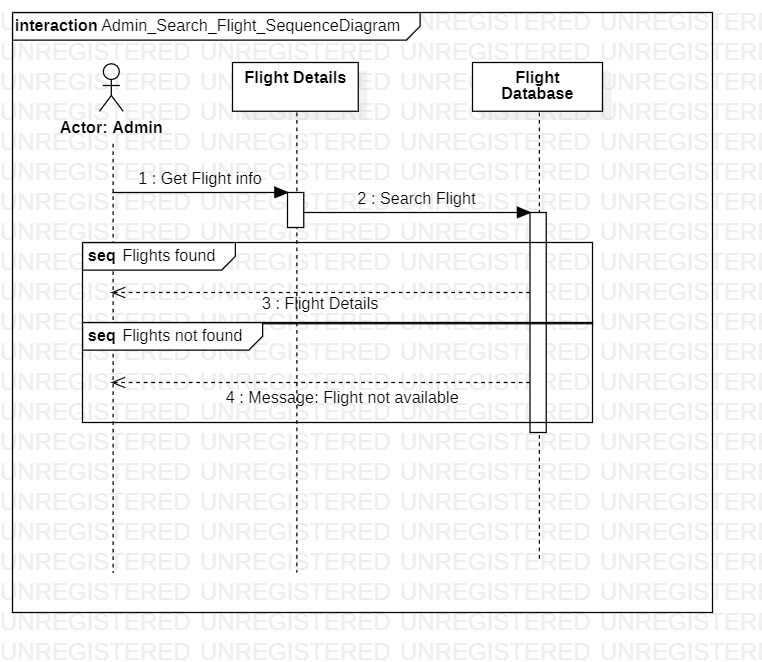
* + 1. Login (Login and Sign Up)



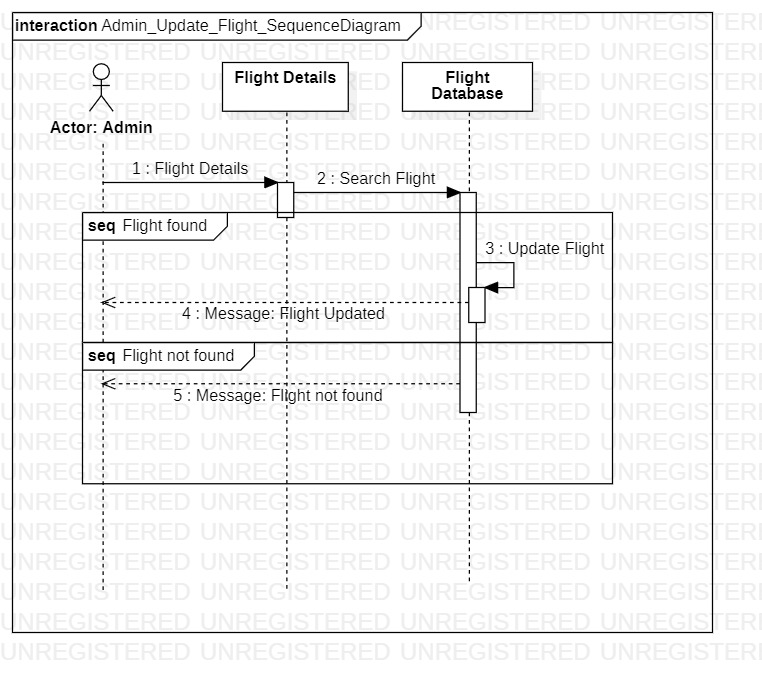
* + 1. Admin’s Add Flight Function



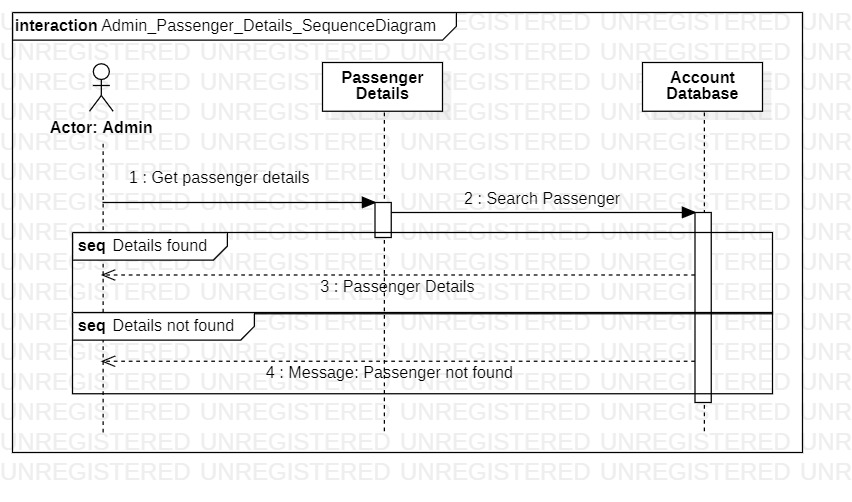
* + 1. Admin’s Search Flight Function



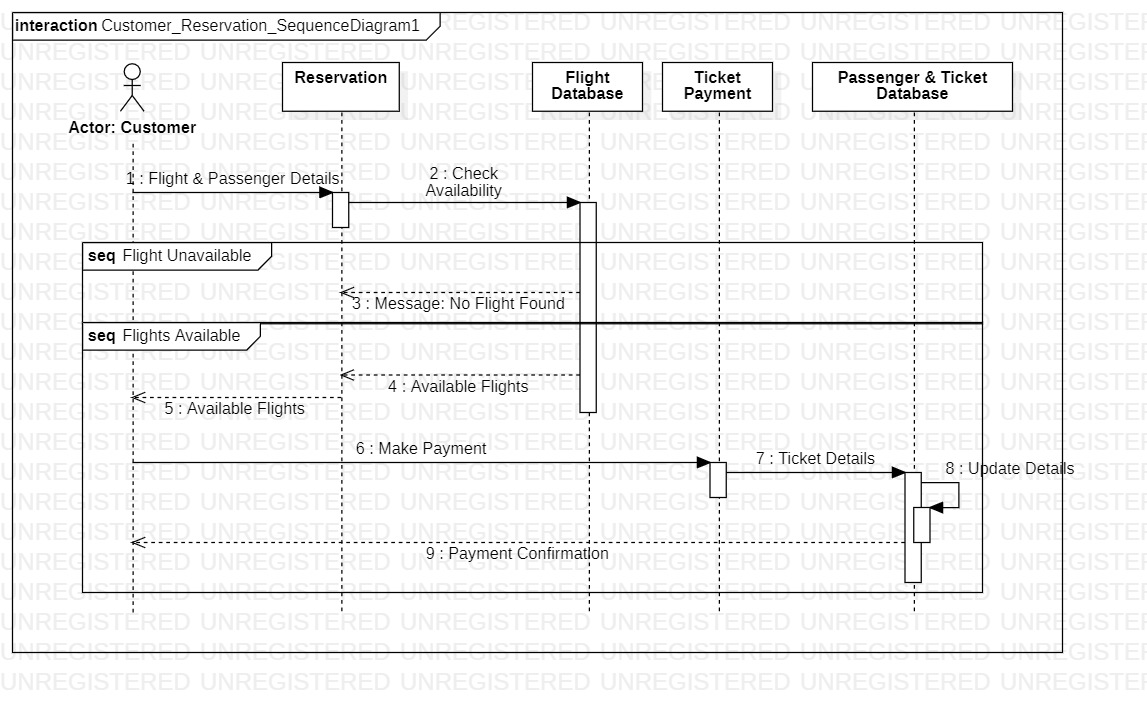
* + 1. Admin’s Update Flight Function



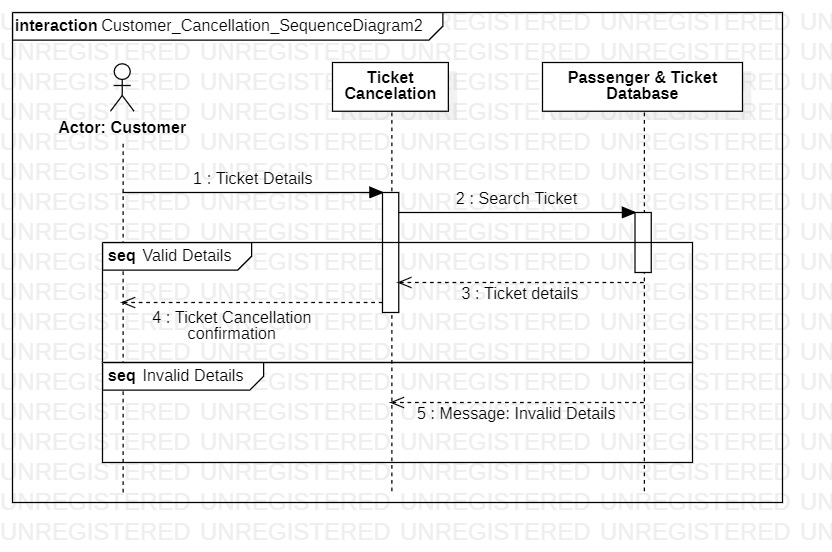
* + 1. Admin’s Get Passenger Details Function



* + 1. Customer’s Reservation Function



* + 1. Customer’s Ticket Cancellation Function

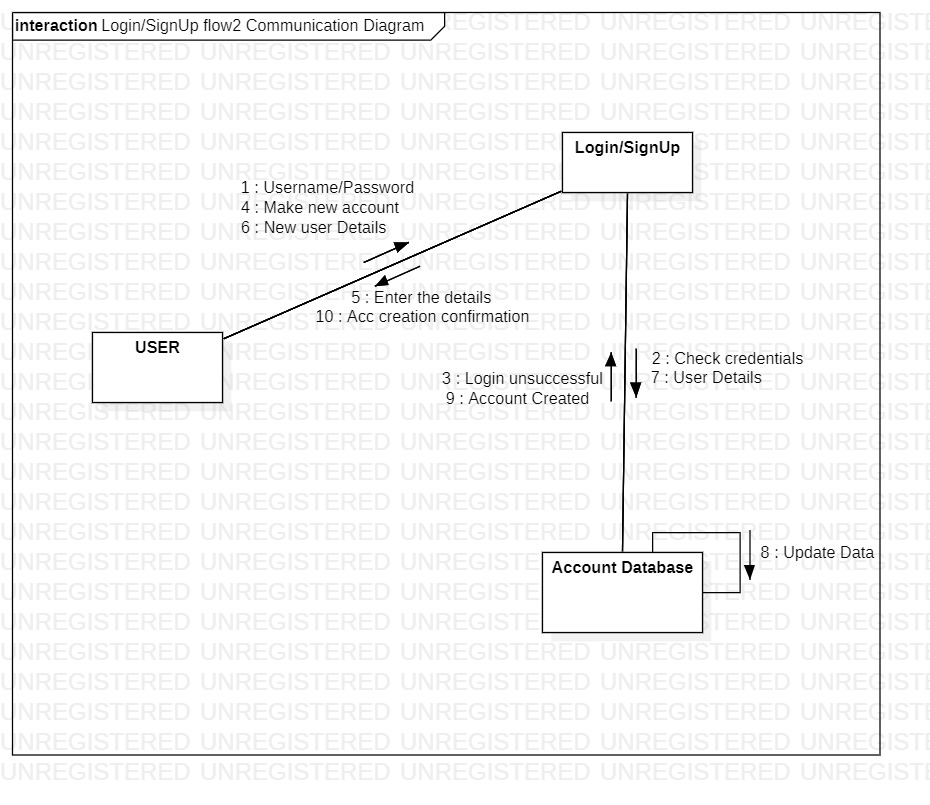


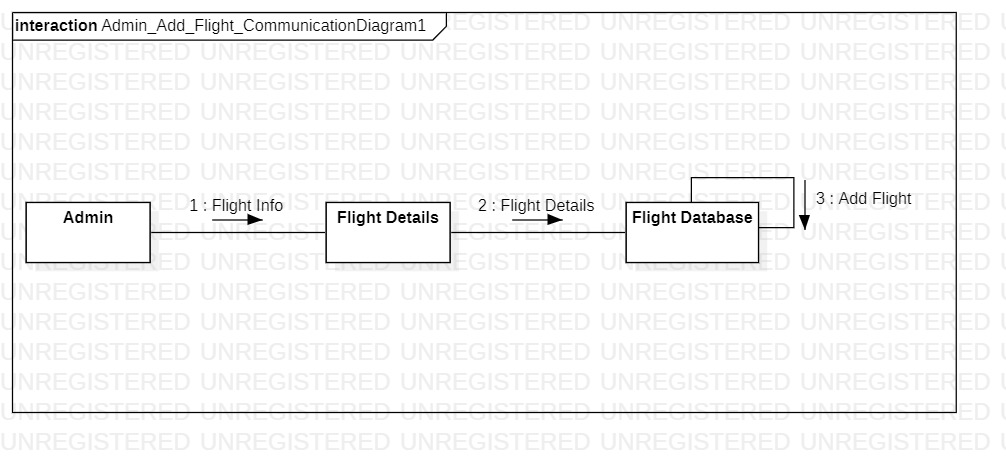
### **COLLABORATION DIAGRAMS**

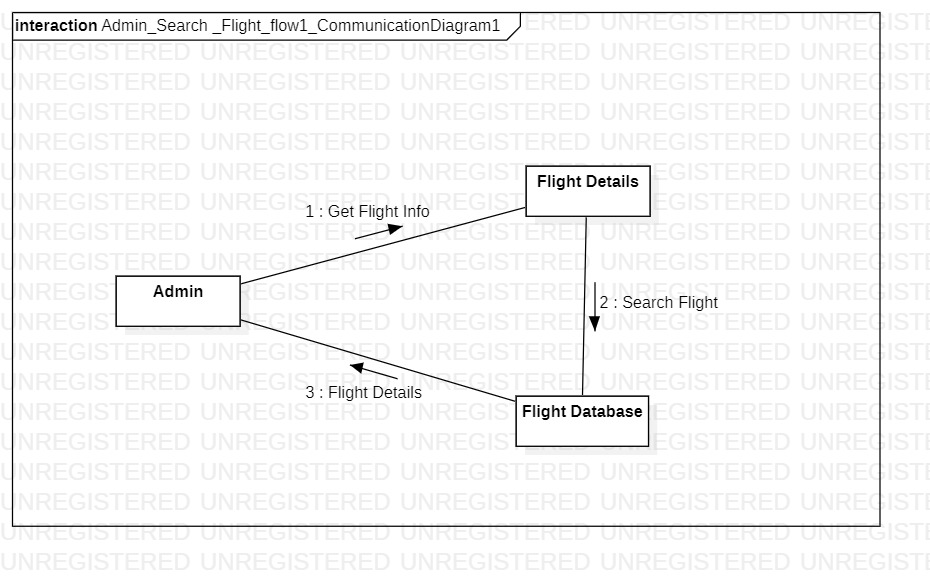
### 1. Login/SignUp flow 1:

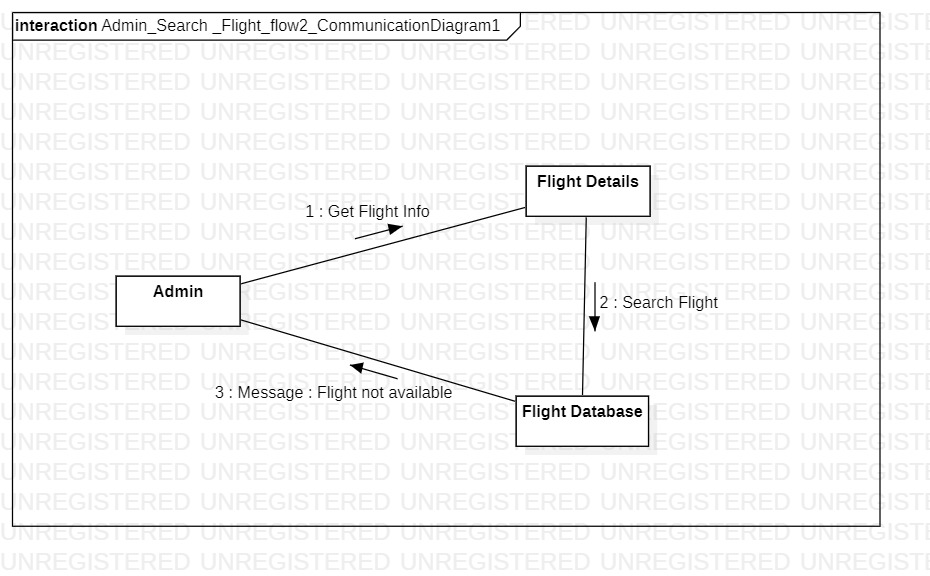
### 

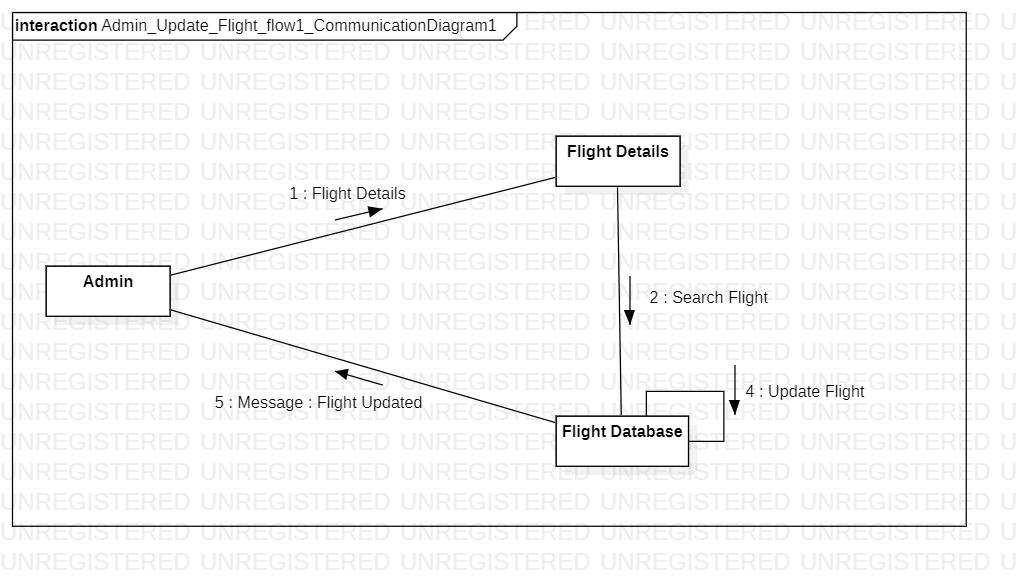
1. Login/SignUp flow 2:

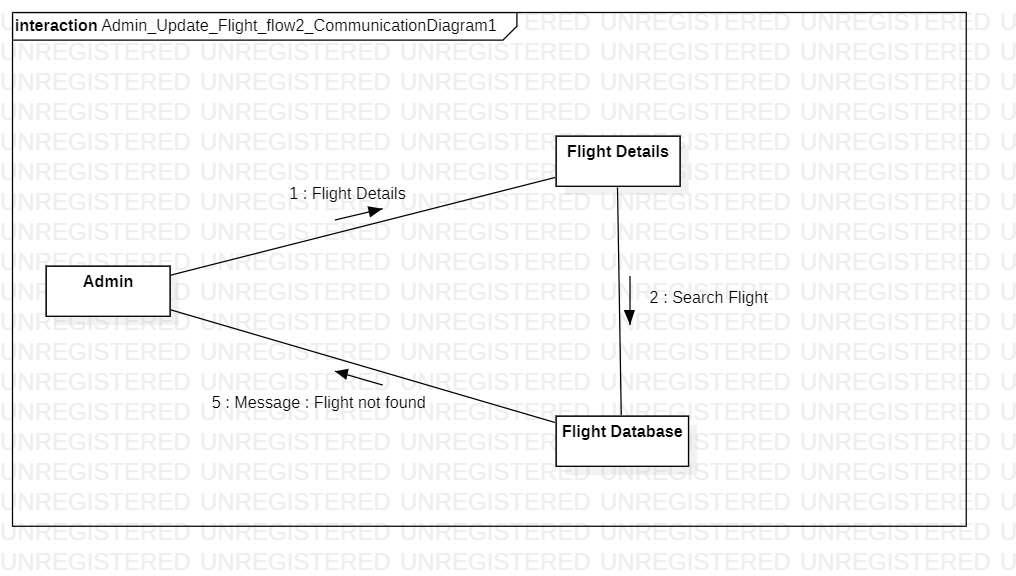


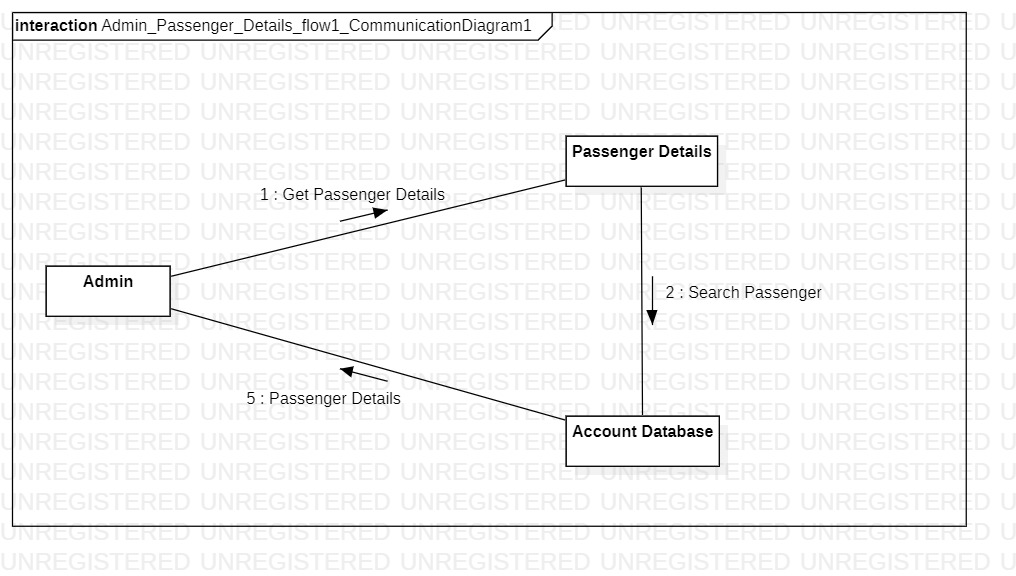
3. Add flight: 

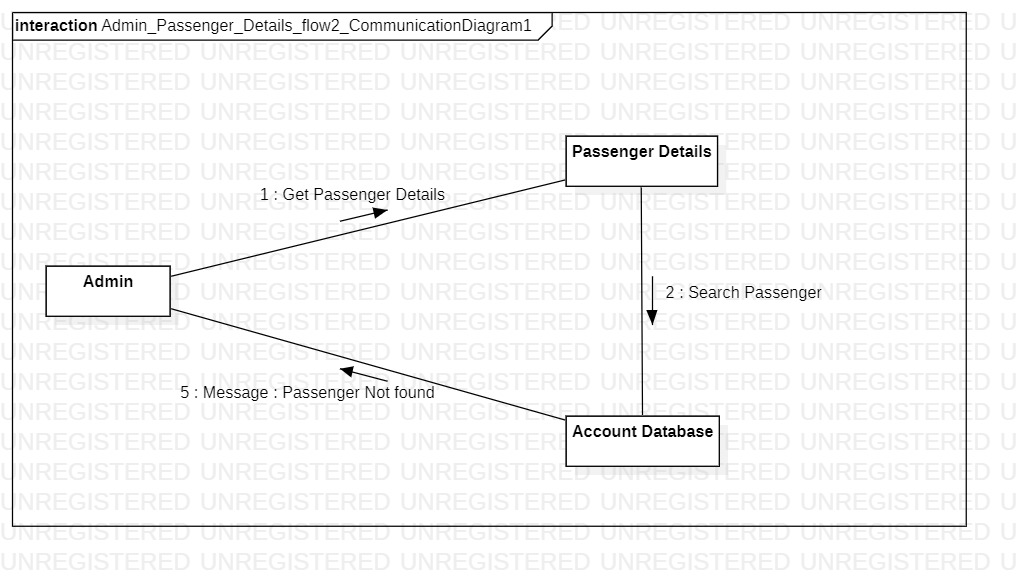
4. Search Flight flow 1: 

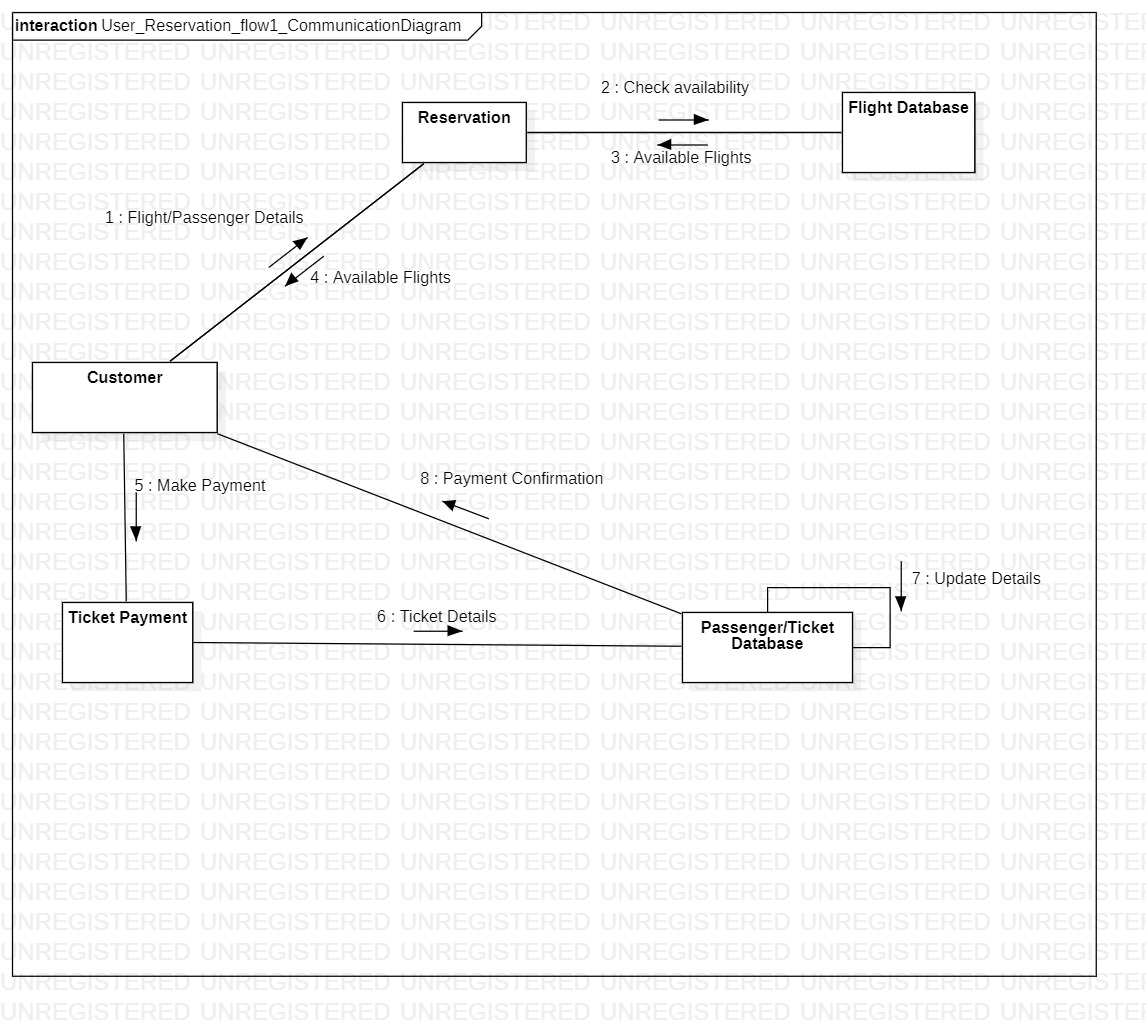
5. Search Flight flow 2: 

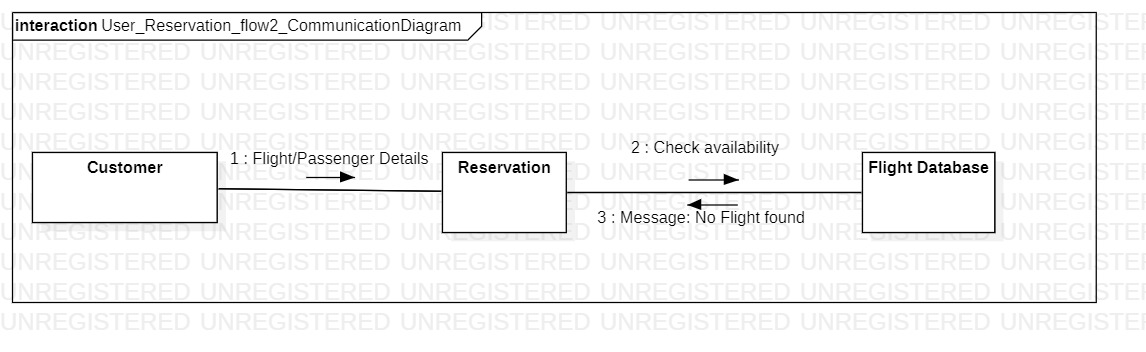
6. Update Flight flow1: 

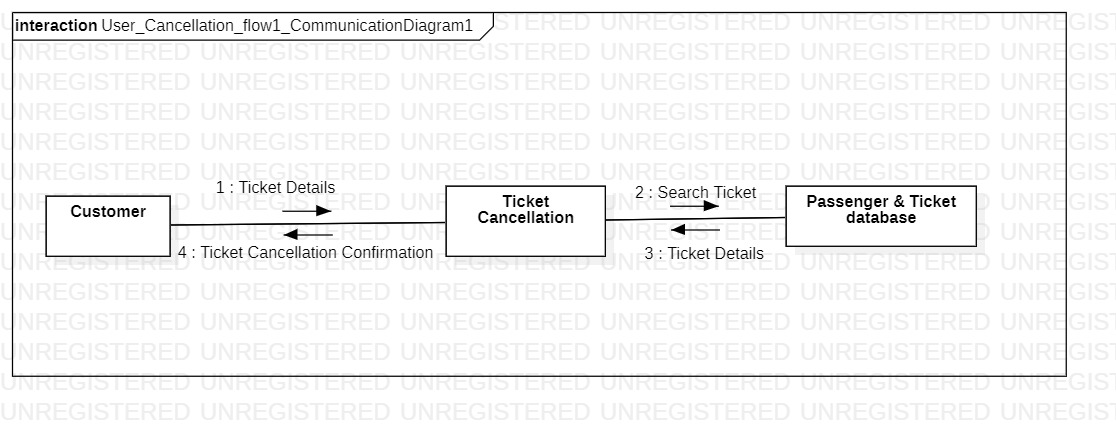
7. Update Flight flow 2: 

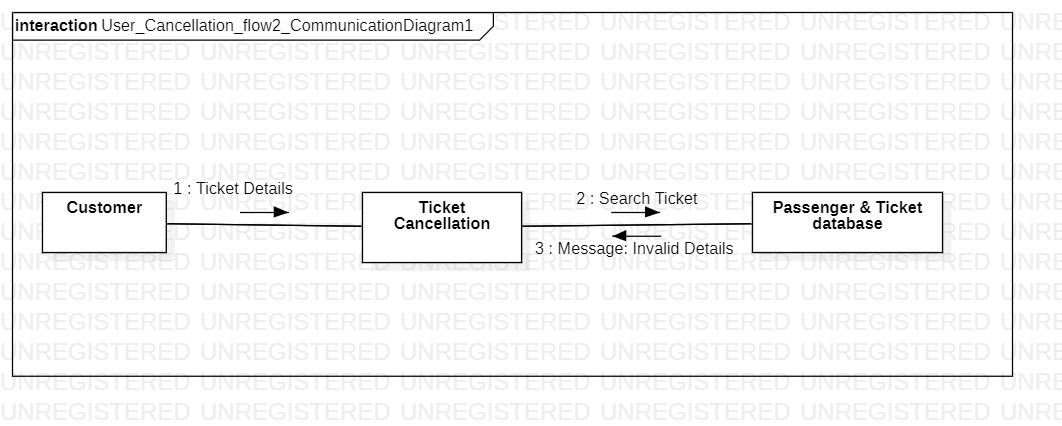
8. Get Passenger's details flow 1: 

9. Get Passenger's details flow 2: 

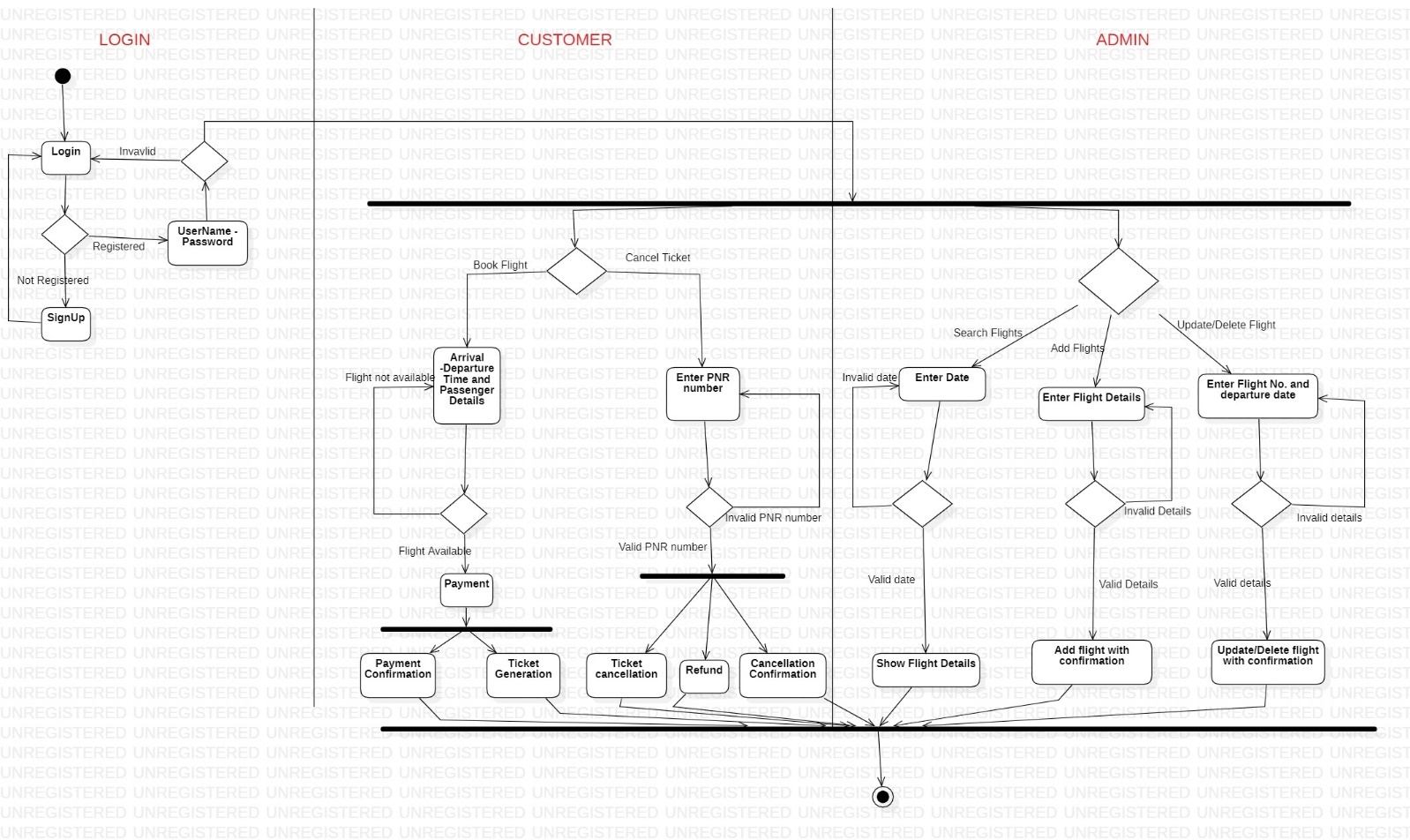
10. Book Ticket flow 1: 

11. Book Ticket flow 2: 

12. Ticket Cancellation flow 1: 

13. Ticket Cancellation flow 2 

**ACTIVITY DIAGRAMS**



**TEST PLAN AND REPORT**

**OBJECTIVES**

The main objectives of the test plan for the Airline Reservation System are as follows:

1. To identify the features of the system that will be tested.
2. To define the pass/fail criteria for each item that will be tested
3. To discuss the testing techniques being used to test the Airline Reservation System.

**REFERENCES**

The following references have been used in the preparation of the Test Plan document for the Airline Reservation System:

**DEFINITIONS**

The following are some of the terms and definitions that are related to the test plan of the Airline Reservation System:

* **Pass/Fail criteria:** Decision rules that are used to determine whether a software item passes or fails a test.
* **Test:** A collection of one or more test cases
* **Test Item:** A software item that is an objective of testing.
* **Test Plan:** A document describing the scope, approach, resources and schedule of the intended testing activities.
* **Test Summary Report:** A document summarizing the testing activities and results.
* **Testing:** The process of analysing a software item to detect the differences between the existing and required conditions.

**ENVIRONMENTAL REQUIREMENTS**

**Hardware**

* Working internet connection

**Software**

* The requirements for a client are:
  + Operating System - Windows/Linux
  + HTML/CSS
* The server should have:
  + Apache
  + PHP
  + MySQL

**TEST ITEMS**

This section of the test plan lists all the items of the Airline Reservation System project that will be tested:

* User Login
* New User Registration
* Search and book flights
* Tickets Cancelling
* View Flight Details
* View Passenger Details
* Add Flight
* Delete Flight

## 

## TEST CASES

The following are the test cases for the Airline Reservation System:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TEST CASE 1 – USER LOGIN**

|  |  |
| --- | --- |
| **Incorrect Input** | Incorrect user credentials (username/password/user type). |
| **Pass Criteria** | Message is generated to indicate that an invalid user credentials have been typed. |
| **Correct Input** | Valid user credentials are entered. |
| **Pass Criteria** | User should be directed to the webpage that he is intended to go to after he logs into the system. |

Photo 1: Invalid Credentials

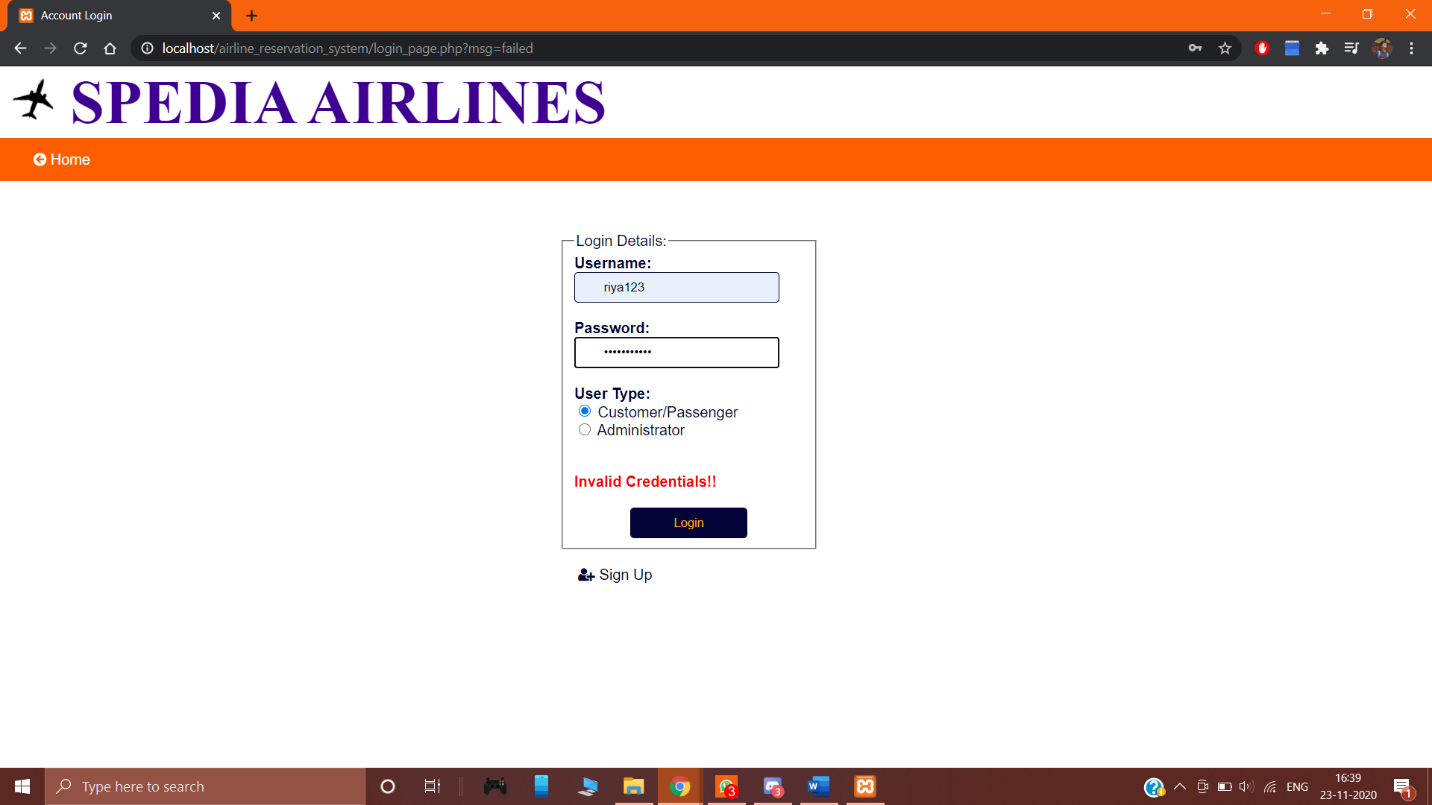
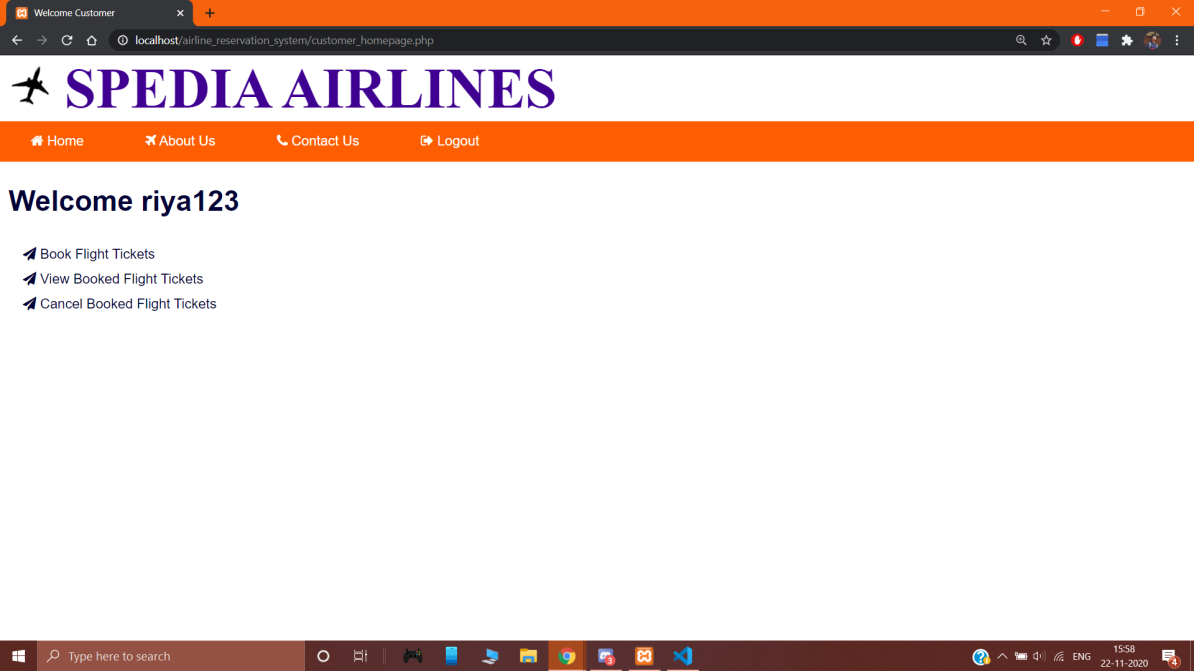


Photo 2: Correct Credentials & User is able to login



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TEST CASE 2 – USER LOGIN**

|  |  |
| --- | --- |
| **Incorrect Input** | Login fields are left empty in the login page. |
| **Pass Criteria** | Appropriate message is generated to indicate that the login fields are left empty. |
| **Correct Input** | Every field is filled in the login page. |
| **Pass Criteria** | Either the user is logged in or an error message is shown depending on the validity of the user credentials. |

Photo 3: Empty field error

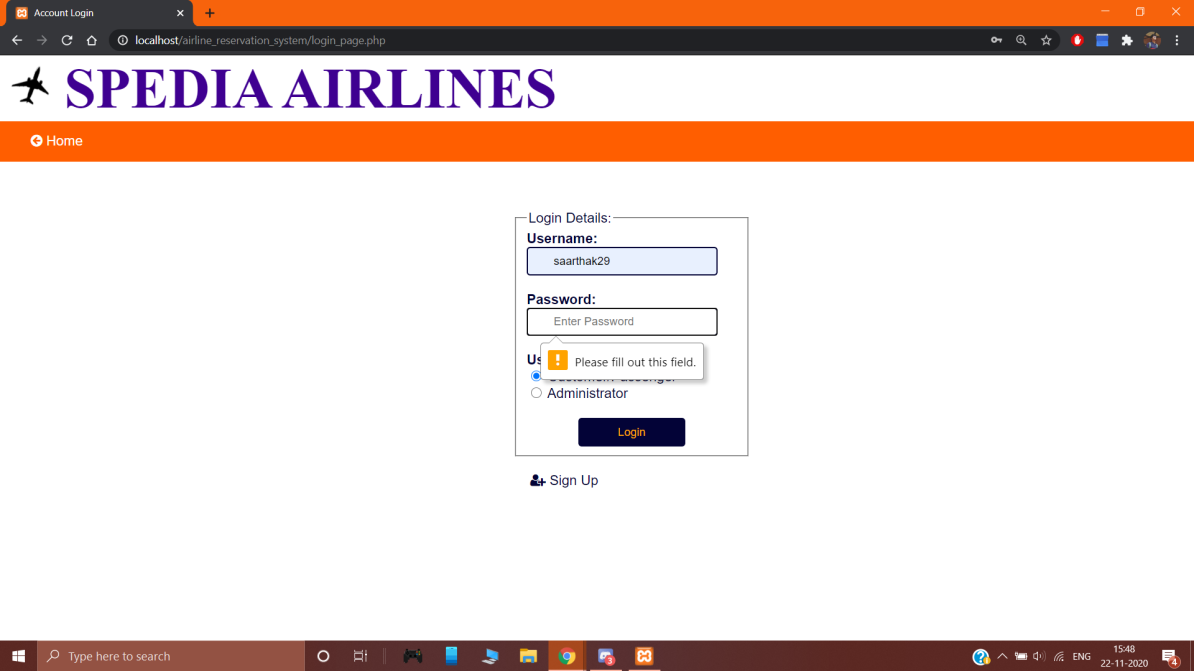
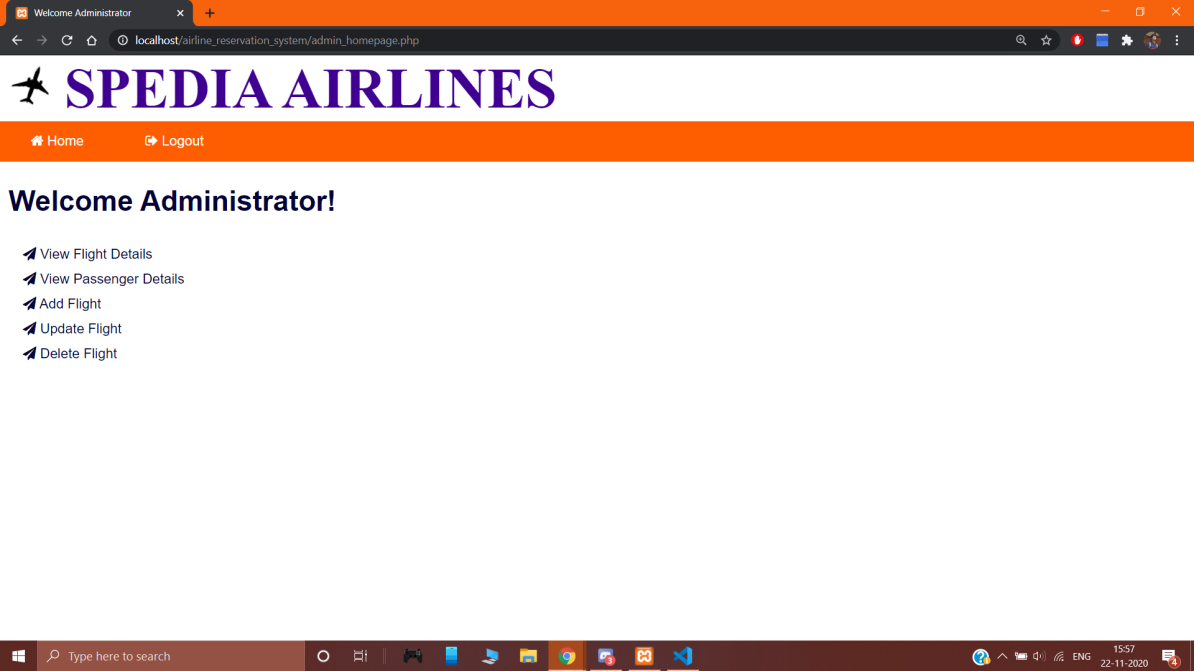


Photo 4: User logged in when every field is filled (Correctly)



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

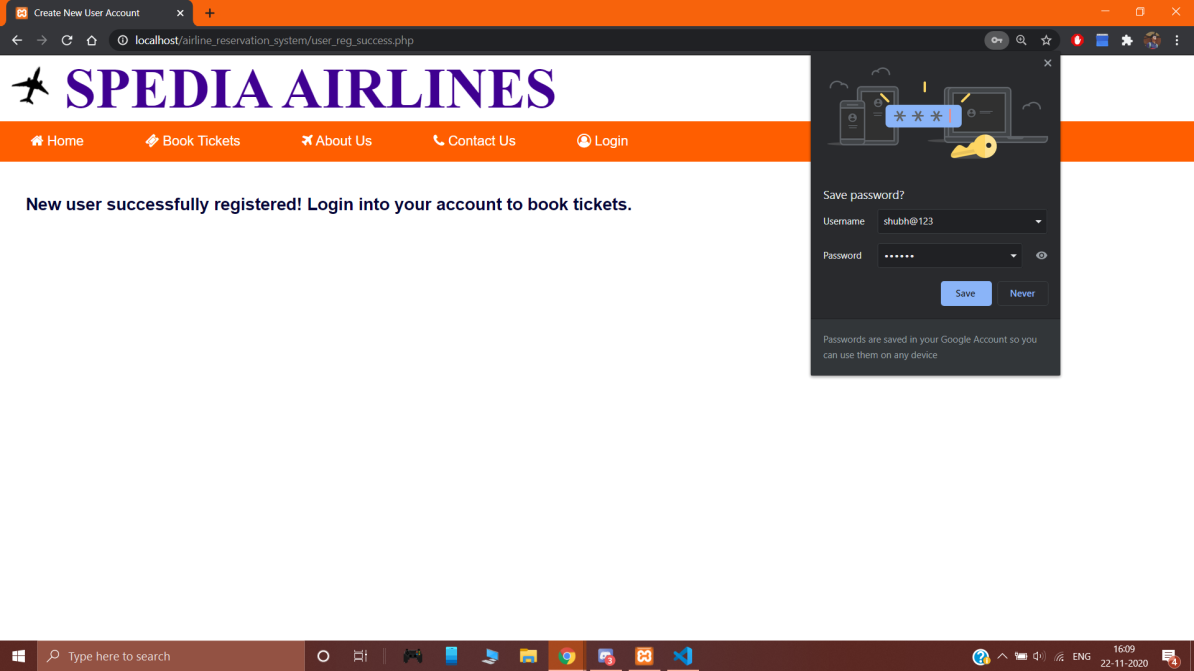
**TEST CASE 3 – NEW USER REGISTRATION**

|  |  |
| --- | --- |
| **Incorrect Input** | The data fields are left out empty in the registration page. |
| **Pass Criteria** | An error message is generated to the user indicating that the fields are to be filled in order to be registered into the system. |
| **Correct Input** | The customer enters the data in all the fields in the registration form. |
| **Pass Criteria** | It accepts all the customer details and then registers the customer and helps him log into the system. |

Photo 5: Empty field error



Photo 6: User signed-up when every field is filled

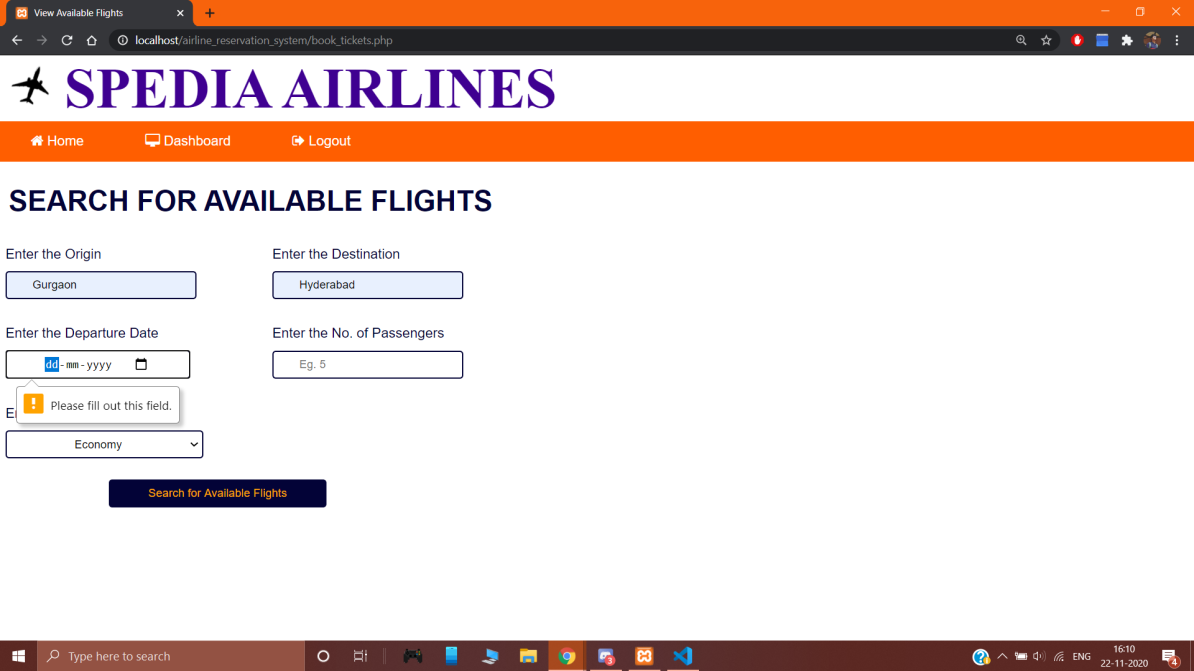


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 4 - Search and Book Flights**

|  |  |
| --- | --- |
| **Incorrect Input** | The data fields are left out empty in the book flights page. |
| **Pass Criteria** | An error message is generated to the user indicating that the fields are to be filled in order to book desired tickets. |
| **Correct Input** | The customer enters the data in all the fields. |
| **Pass Criteria** | Either it would show valid results or an error message depending on the validity of the data. |

Photo 7: Empty field error

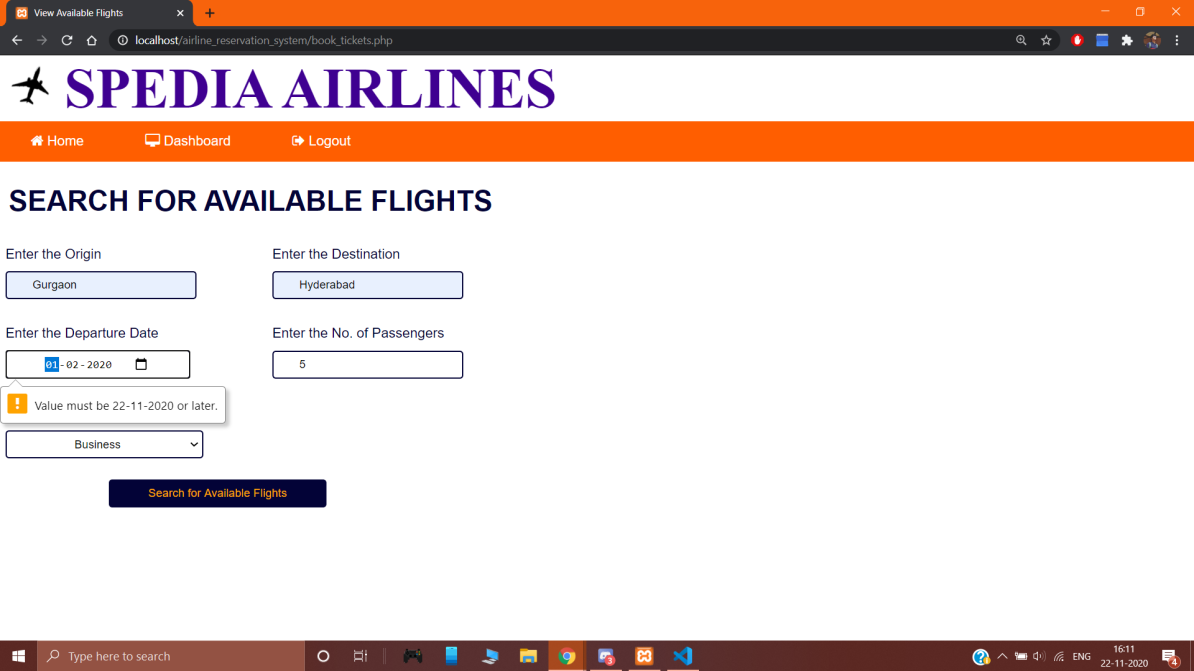


**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 5 - Search and Book Flights**

|  |  |
| --- | --- |
| **Incorrect Input** | Incorrect format of data entered into the data entry fields. |
| **Pass Criteria** | A message is generated indicating the wrong format of data is entered. |
| **Correct Input** | Correct format of input is entered. |
| **Pass Criteria** | The search would return results based on the search parameters after which the booking process is continued. |

Photo 8: Incorrect data format (Date)



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 6 - Search and Book Flights**

|  |  |
| --- | --- |
| **Incorrect Input** | Incorrect search parameters entered. |
| **Pass Criteria** | A message is generated indicating no information is available for the incorrect search parameters. |
| **Correct Input** | Correct search parameters are entered. |
| **Pass Criteria** | The search would return valid results and then after the booking process, the system would generate a PNR number for the customer which is saved in the account. |

Photo 9: Incorrect flight search parameters

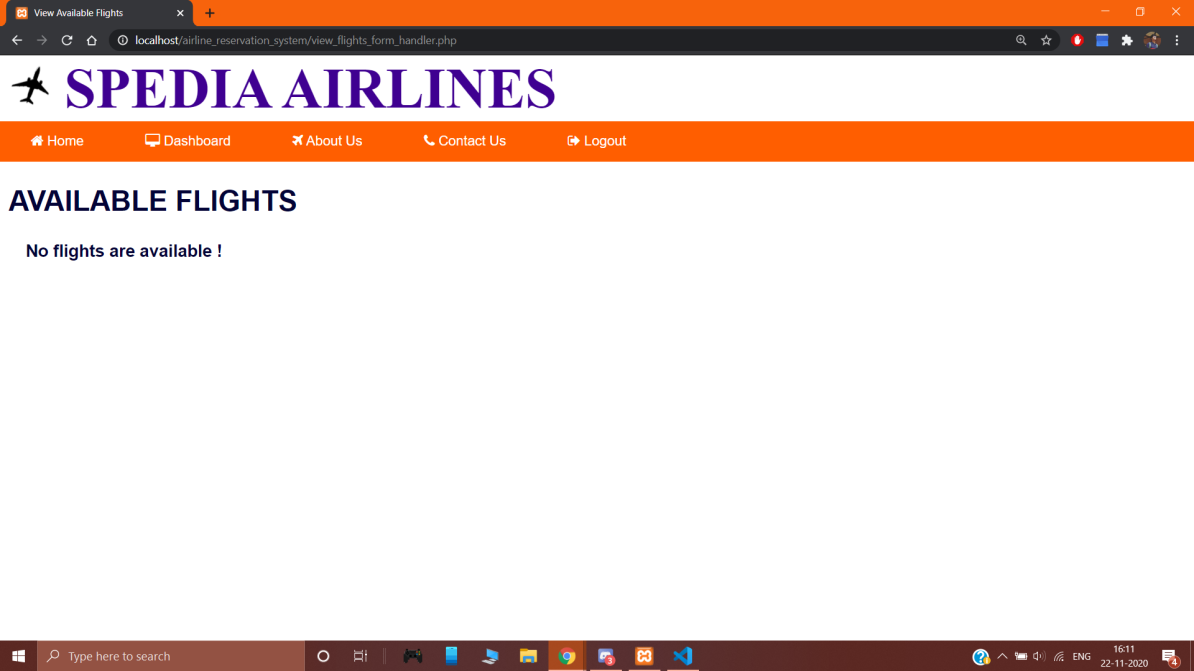
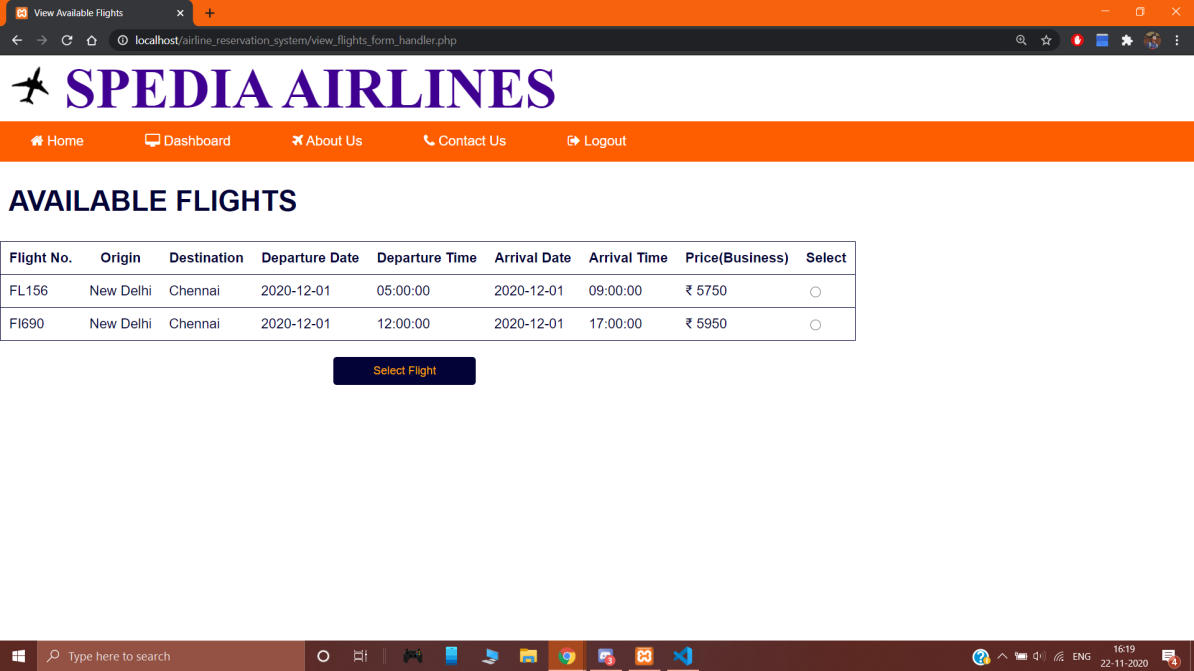


Photo 10: Desired flight results displayed after correct flight search parameters are entered



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 7 - Tickets Cancelling**

|  |  |
| --- | --- |
| **Incorrect Input** | The PNR field is left out empty. |
| **Pass Criteria** | An error message is generated to the user saying that they have to fill out that field in order to cancel flight. |
| **Correct Input** | The customer enters the PNR. |
| **Pass Criteria** | Either it would show valid results or an error message depending on the validity of the PNR. |

Photo 11: Empty field error



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 8 - Tickets Cancelling**

|  |  |
| --- | --- |
| **Incorrect Input** | Customer enters invalid PNR number. |
| **Pass Criteria** | An error message is shown informing the customer about the invalid PNR number. |
| **Correct Input** | Customer enters valid PNR number. |
| **Pass Criteria** | A message is shown that the tickets is cancelled successfully and the ticket amount would be refunded to his account with 15% cancellation charges. |

Photo 12: Invalid PNR number

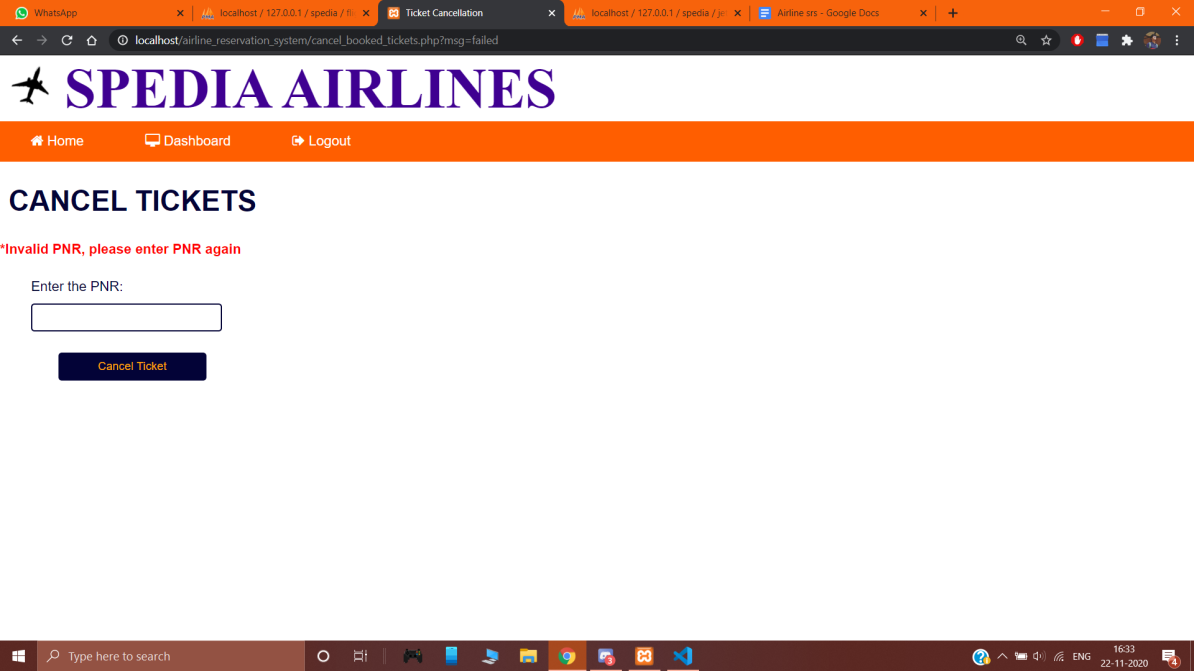
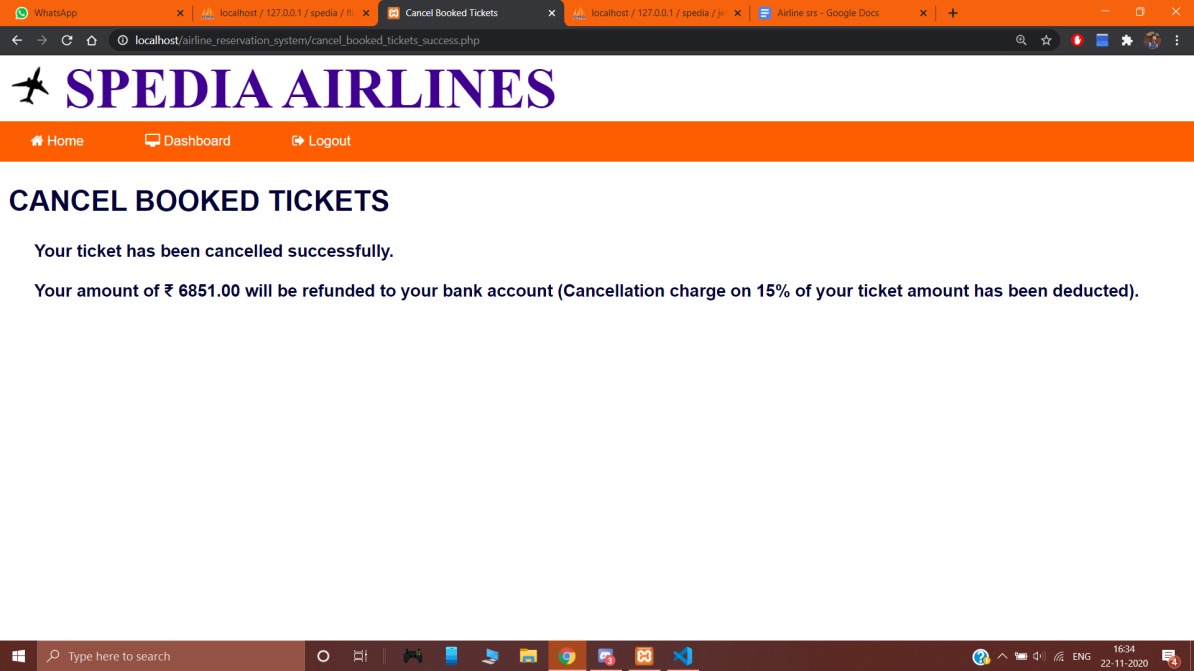


Photo 13: Passenger’s ticket is cancelled after correct PNR number in entered



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 9 - View Flight Details**

|  |  |
| --- | --- |
| **Input** | A date on which there is/was no flight available. |
| **Pass Criteria** | “No flights to show” message is shown to the user. |
| **Input** | A valid date on which there is/was flight(s) available. |
| **Pass Criteria** | Details of flight(s) available on that date are shown to the user. |

Photo 14: View flight details page

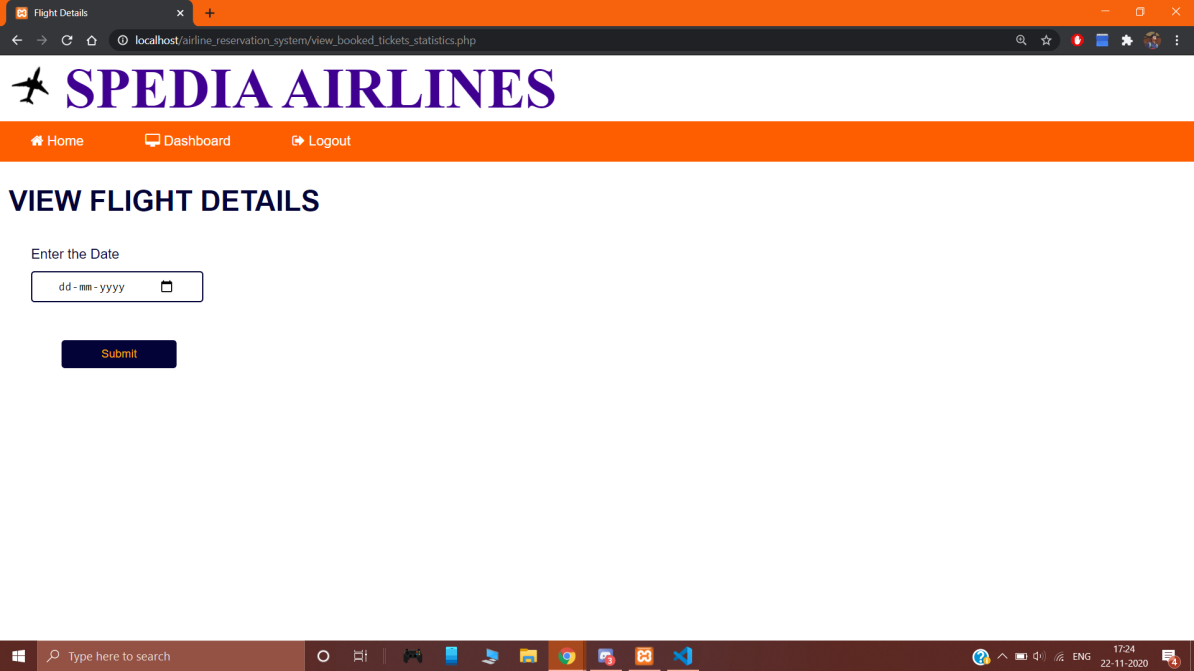


Photo 15: Incorrect flight search parameters

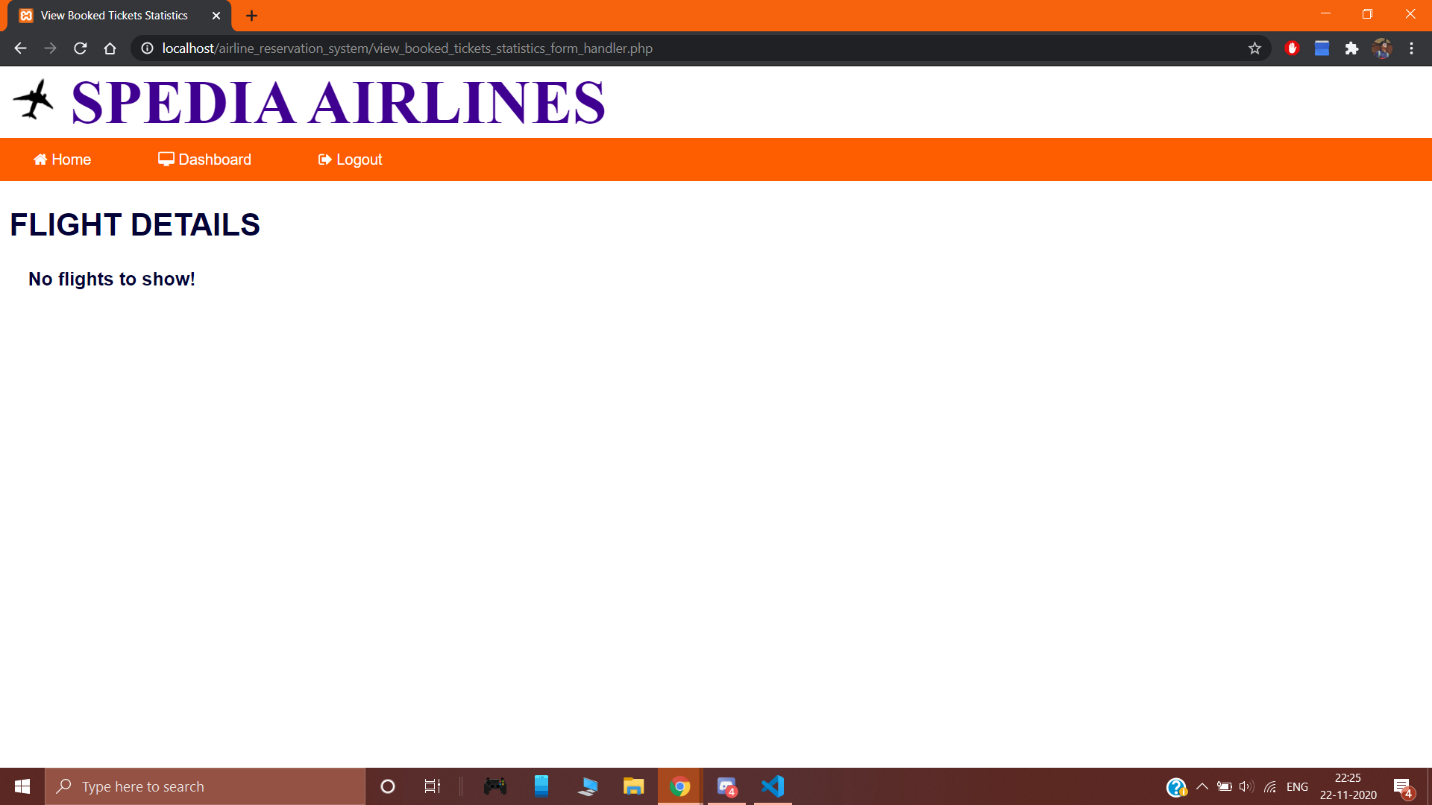
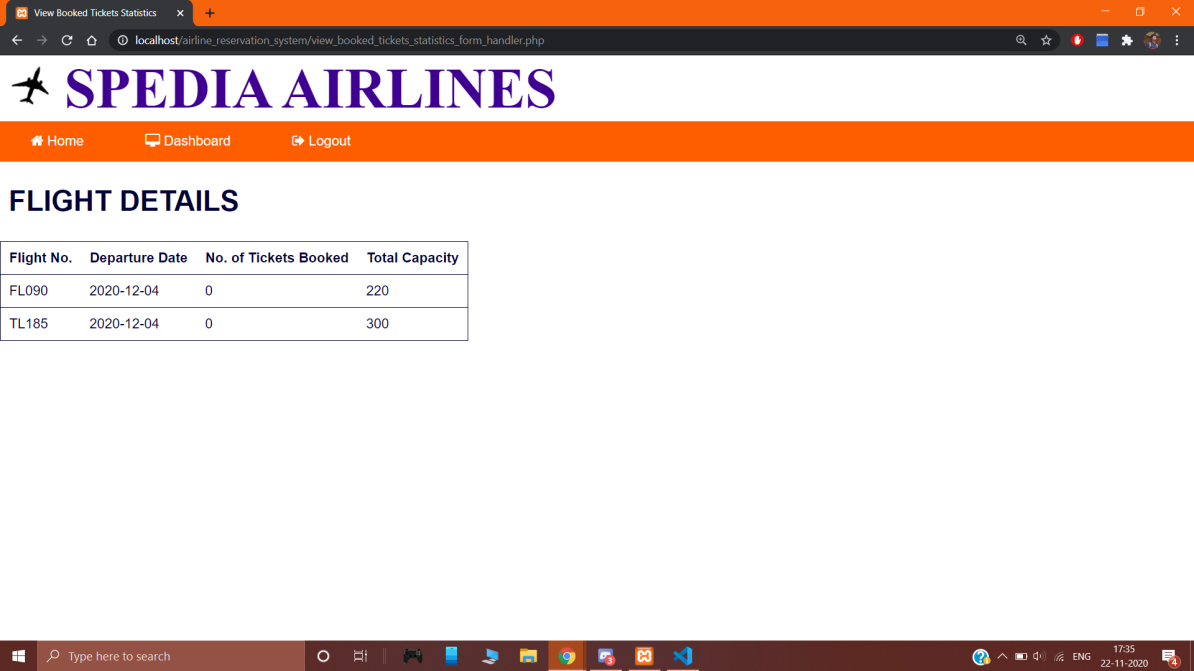


Photo 16: Desired flight details displayed after correct search details



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 10 - Details of Passengers**

|  |  |
| --- | --- |
| **Incorrect Input** | User inputs invalid Flight No. or Date. |
| **Pass Criteria** | ‘No information is available’ message is shown to the user. |
| **Correct Input** | User inputs valid Flight No. and date. |
| **Pass Criteria** | Details of passengers for that particular flight is shown to the user. |

Photo 17: View passenger details page

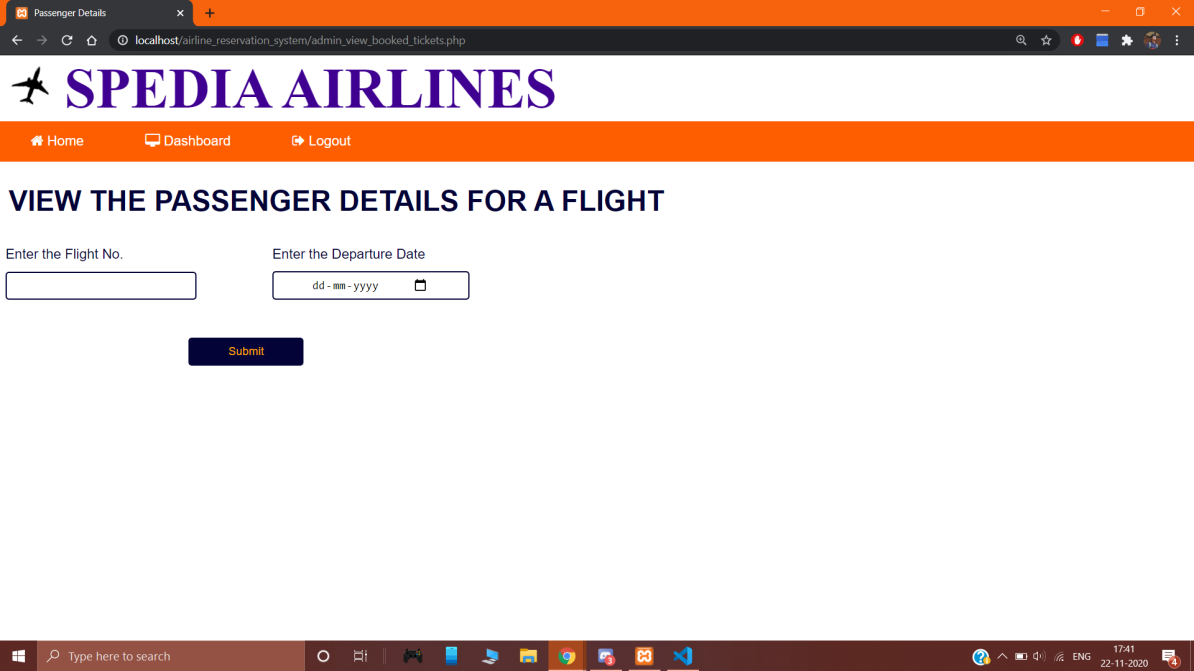


Photo 18: Incorrect search details

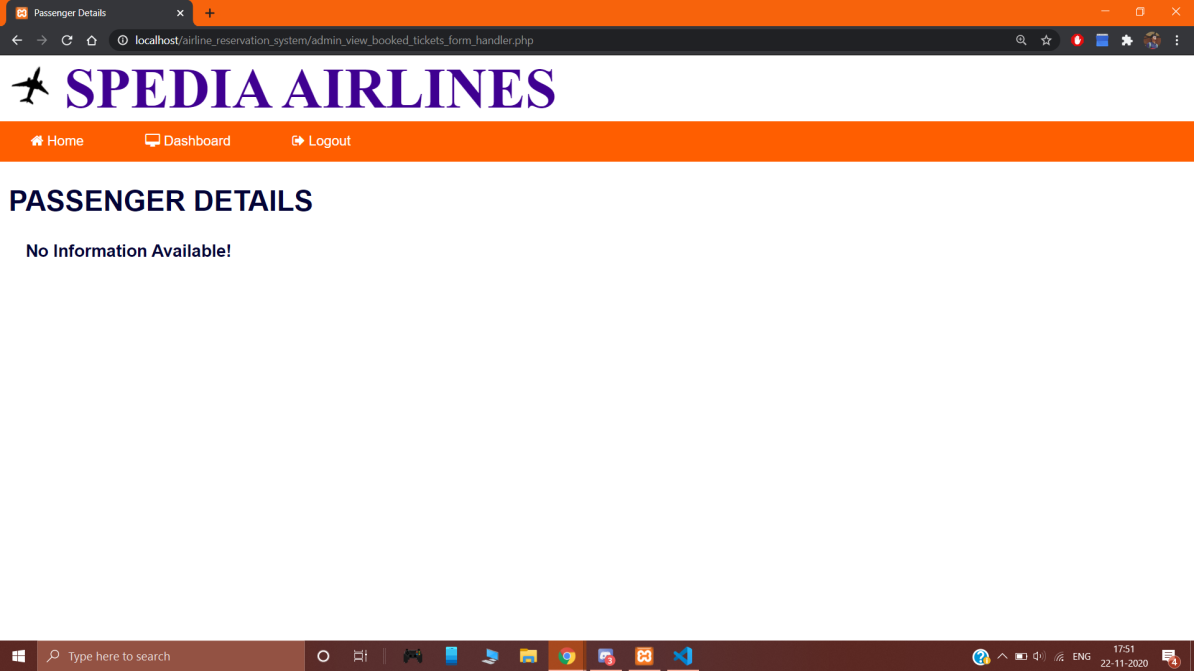
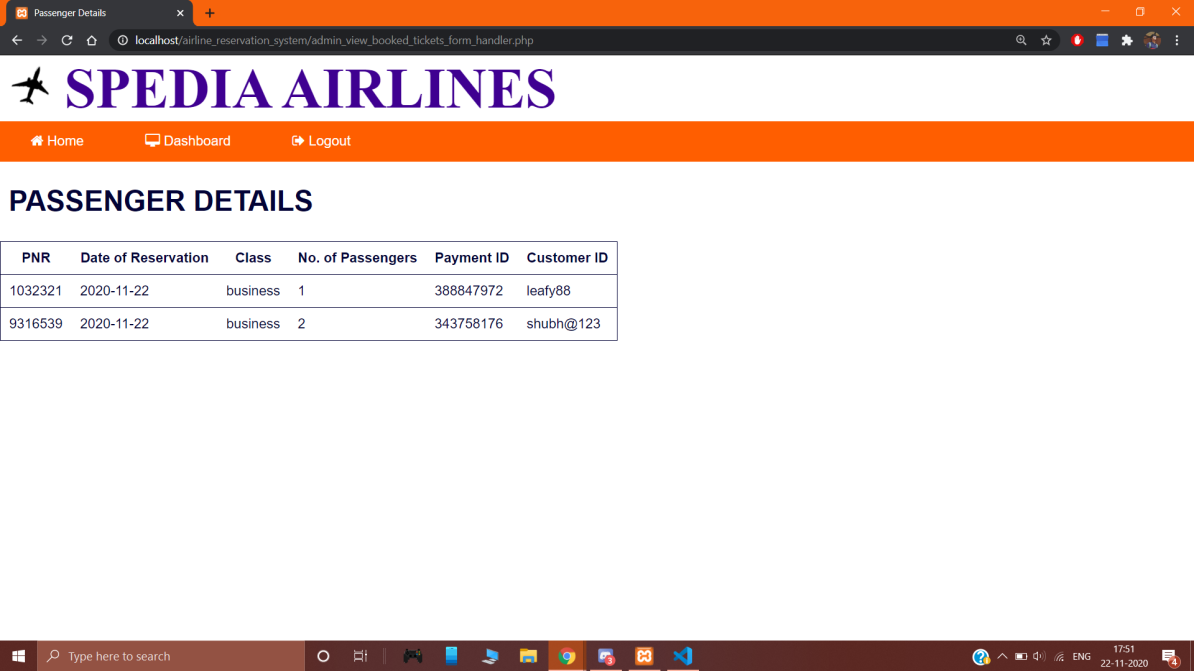


Photo 19: Desired passenger details displayed after correct search details



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 11 - Add Flights**

|  |  |
| --- | --- |
| **Incorrect Input** | The data fields are left out empty in the Add Flights page. |
| **Pass Criteria** | An error message is generated to the user indicating that the fields are to be filled in order to add desired flight. |
| **Correct Input** | The user enters the appropriate data in all the fields. |
| **Pass Criteria** | A message is shown to the user informing about successful flight addition. |

Photo 20: Empty field error

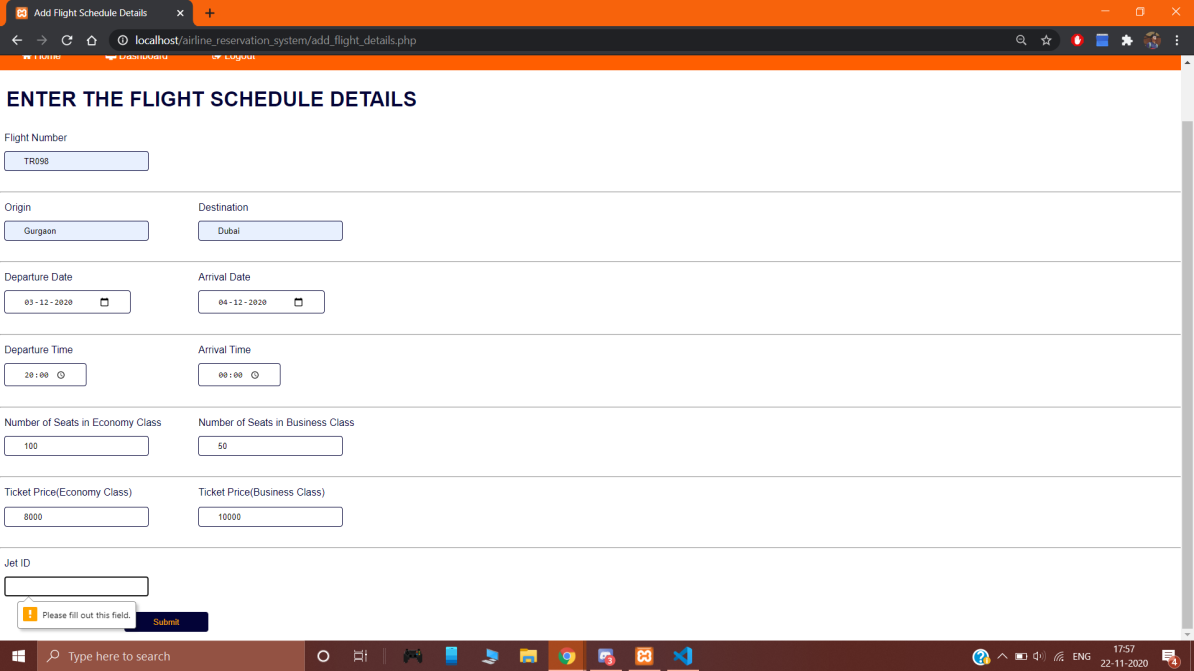
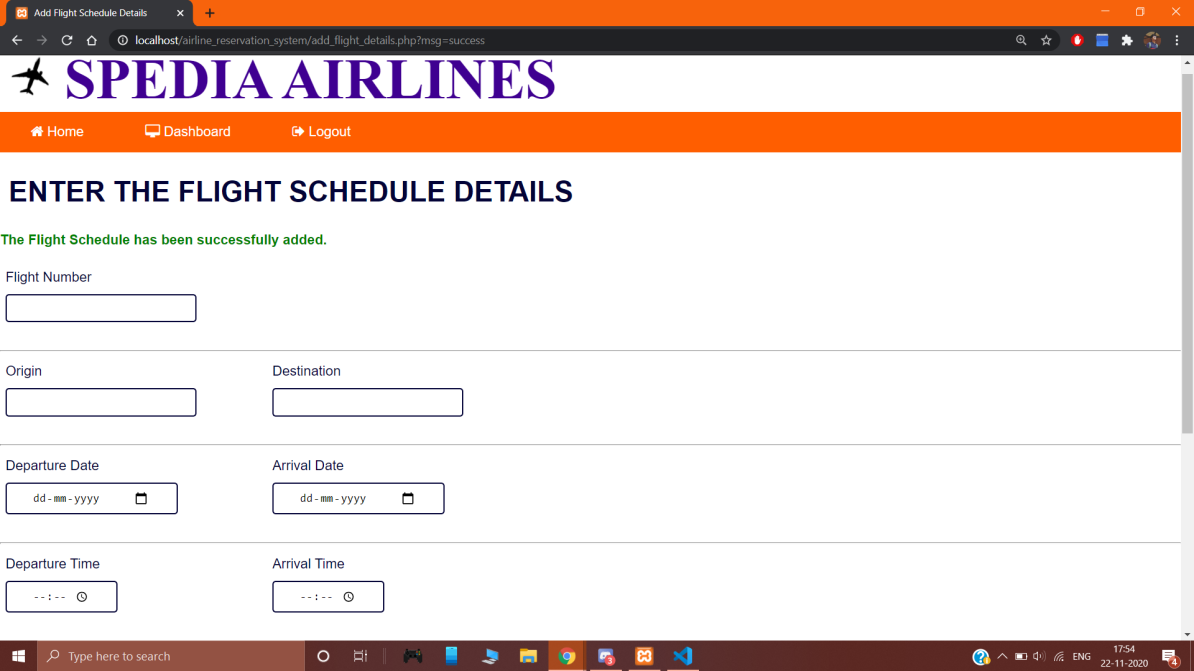


Photo 21: Flight successfully added after appropriate details are filled in every field



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Test Case 12 - Delete Flights**

|  |  |
| --- | --- |
| **Incorrect Input** | User inputs invalid Flight No. or Date of Departure. |
| **Pass Criteria** | An error message is shown to the user indicating that the flight no./date is invalid. |
| **Correct Input** | User inputs valid flight no. and date of departure. |
| **Pass Criteria** | A message is shown to the user informing about successful flight deletion. |

Photo 21: Invalid flight details

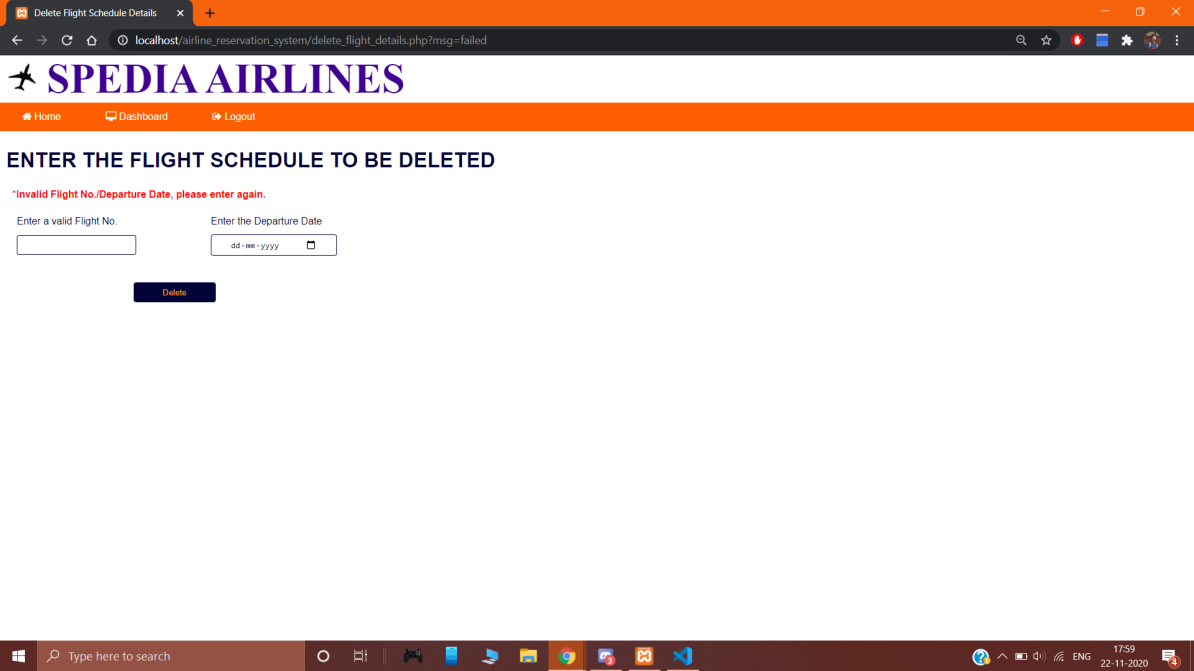
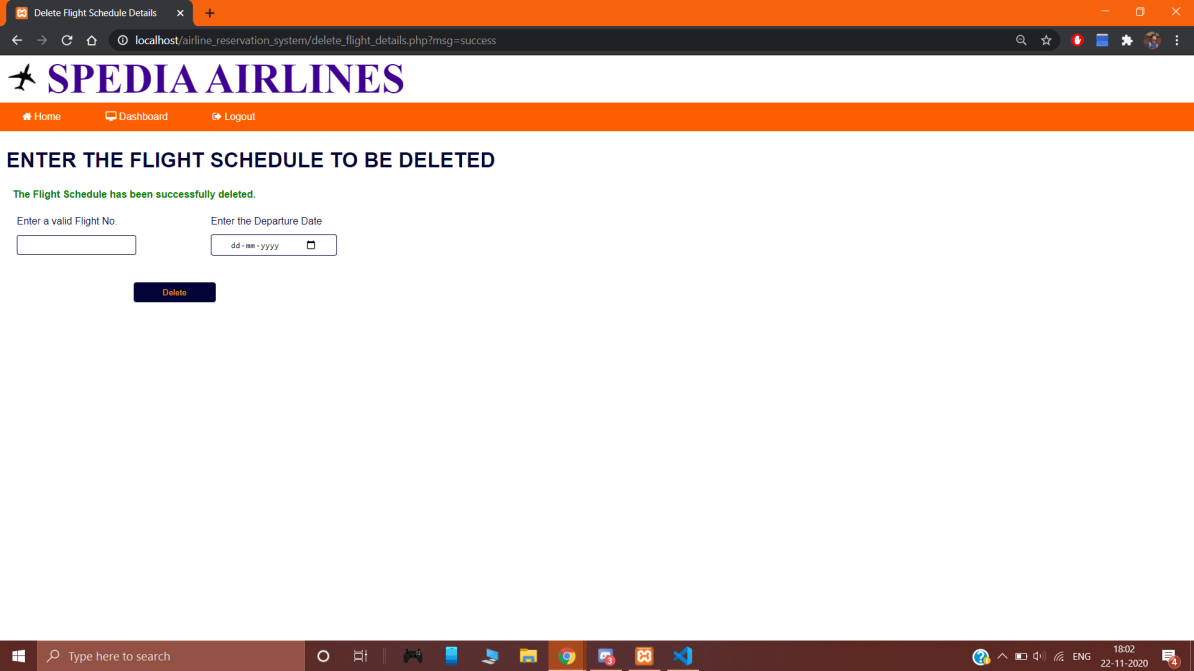


Photo 22: Flight is deleted after correct details are entered



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## SCENARIO MATRIX

**Use case- User Login and Registration**

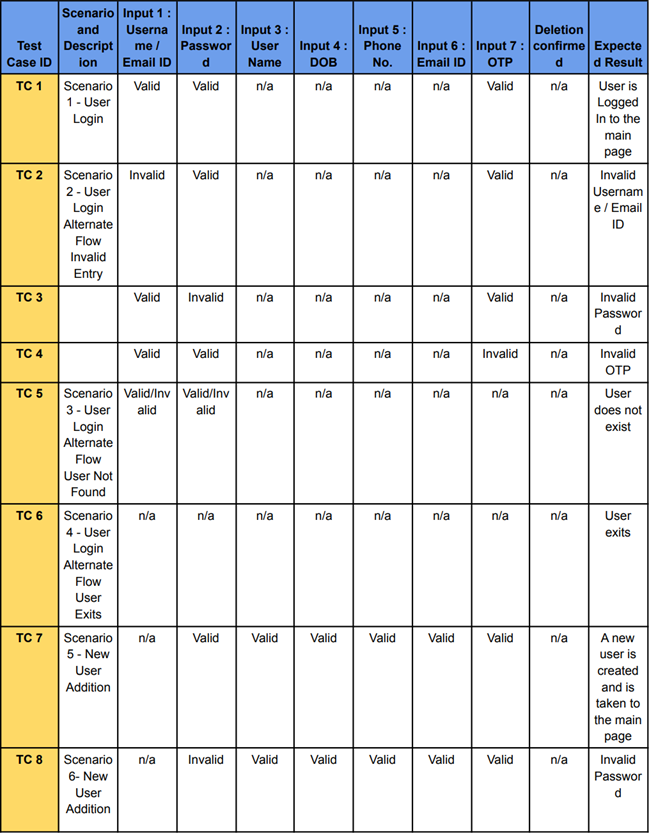
**Variables**

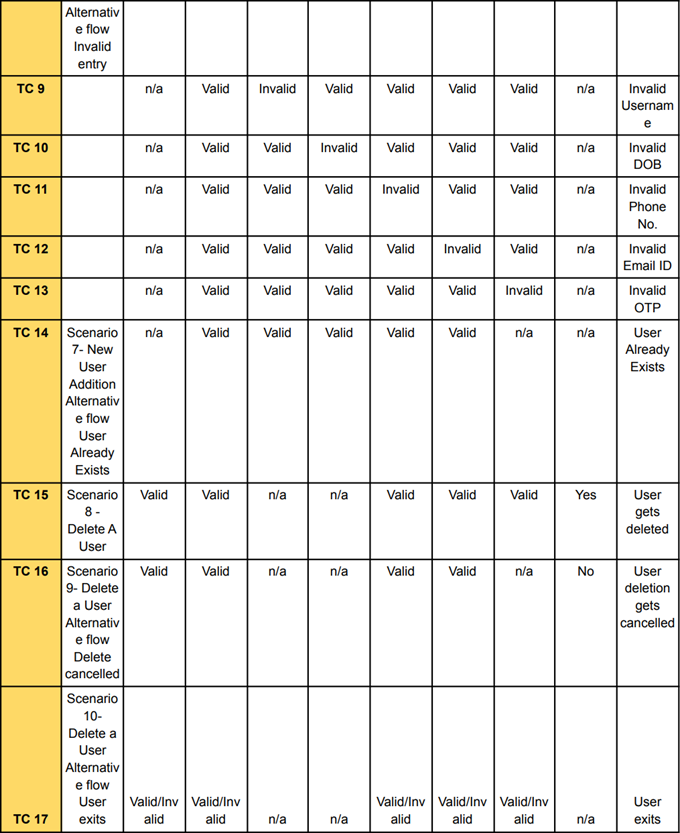
1. Username/Email Id
2. Password
3. Name
4. DOB
5. Phone No.
6. Email ID
7. OTP

## Input States

Valid or Invalid or N/A

## Test Case Matrix for the above Use Case





**Assigning actual values in Test Case Matrix**

