**Table of Contents**



**FPT International School**

--- o0o ---

**PROJECT REPORT**

**<Airlines Reservation System>**

Batch: T2004M

Semester: 2

Group: 1

1. NGUYEN TIEN DAT
2. DUONG XUAN BINH

**Hanoi - 04/2021**

[1. Problem Definition 4](#_Toc67432801)

[1.1 Overview 4](#_Toc67432802)

[*i.* *Introduction* 4](#_Toc67432803)

[*ii.* *Implementation Environment* 13](#_Toc67432804)

[1.2 User Stories 14](#_Toc67432805)

[*Guest* 14](#_Toc67432806)

[a) Register 14](#_Toc67432807)

[b) Check flight availability 14](#_Toc67432808)

[c) Pick a seat 14](#_Toc67432809)

[d) Block a seat 14](#_Toc67432810)

[e) See the total price 14](#_Toc67432811)

[f) Search a flight detail 14](#_Toc67432812)

[g) Cancel blocked ticket 14](#_Toc67432813)

[*Registered user* 14](#_Toc67432814)

[a) Sign in 14](#_Toc67432815)

[b) Check flight availability 14](#_Toc67432816)

[c) Pick a seat 15](#_Toc67432817)

[d) Block a seat 15](#_Toc67432818)

[e) Cancel blocked ticket 15](#_Toc67432819)

[f) Buy ticket 15](#_Toc67432820)

[g) Reschedule 15](#_Toc67432821)

[h) See the total price 15](#_Toc67432822)

[i) See the ticket status 15](#_Toc67432823)

[j) Search a flight detail 15](#_Toc67432824)

[Flowchart 16](#_Toc67432825)

[1.3 Register 16](#_Toc67432826)

[1.4 Check availability 17](#_Toc67432827)

[1.5 Block or Buy ticket 18](#_Toc67432828)

[1.6 Reschedule 19](#_Toc67432829)

[1.7 Cancellation 20](#_Toc67432830)

[1.8 Flight Details 21](#_Toc67432831)

[1.9 Ticket Status 22](#_Toc67432832)

[1.10 Update profile 23](#_Toc67432833)

[2. UI\UX 24](#_Toc67432834)

[2.1 Register 24](#_Toc67432835)

[2.2 Homepage 24](#_Toc67432836)

[2.3 New Arrival 24](#_Toc67432837)

[2.4 Men 24](#_Toc67432838)

[2.5 Women 24](#_Toc67432839)

[2.6 Promotion 24](#_Toc67432840)

[2.7 Blog 24](#_Toc67432841)

[2.8 Contact 24](#_Toc67432842)

[2.9 Compare Product 24](#_Toc67432843)

[2.10 Shopping Cart 25](#_Toc67432844)

[2.11 Checkout Product 25](#_Toc67432845)

[*i.* Address 25](#_Toc67432846)

[*ii.* Delivery Method 26](#_Toc67432847)

[*iii.* Payment Method 26](#_Toc67432848)

[*iv.* Order Review 27](#_Toc67432849)

[Summary 27](#_Toc67432850)

# Problem Definition

## Overview

### *Introduction*

The project is to design an airline reservation system that will take care of all the reservation related issues and will give the result in quick time.

System must:

* Reduce effort and frustration for travelers in scheduling a trip, especially by reducing the search effort for the flight they need to take.
* Show all possible combinations and itineraries available for a pair of origin-destination cities.
* Reduce redundancy in the information required from the customers in order for them to buy tickets, create user accounts etc.
* Check the validity of input data and give a feedback to the user in case of errors or inconsistency.
* Make it easy for travelers to check the ticket status or make changes to their trip.

The ARS will provide the following types of easy-to-use, interactive, and intuitive graphical and telephonic interfaces.

The ARS will provide an easy-to-use, intuitive Graphical User Interface (GUI) as part of the Clerk/Administrator’s working desktop environment.

The ARS will also provide an interactive GUI, on the World Wide Web for the general customers.

The above two ARS interfaces shall help provide the following functionalities to the users - access to the ARS to check the flight schedule, availability of seats, ticket price and to block, reserve, cancel, and reschedule tickets.

***Requirement Specification:***

***User Accounts***

The passenger, who will henceforth be called the ‘user’, will be presented with 3 choices by the reservation system, as the first step in the interaction between them. A user can choose one of these and his choice would be governed by whether he is a guest or a registered user and whether he wants to check the availability of tickets or also block/buy them. The terms ‘registered user’ and ‘guest’ are described below.

A user who has traveled by the airline earlier would have been given a user id and a password. He would have his personal information stored in the database referred to as ‘DB-user’. This ‘personal information’ would be henceforth referred to as ‘profile’. Such a user with a profile in DB-user shall be called a ‘registered user’. A registered user will be able to check the availability of tickets as well as block/buy a ticket by logging into the system.

A new user, on the other hand, would either have to

1. register himself with the system by providing personal information or
2. log into the system as a guest.

In case of ‘a’, the new user becomes a registered user.

In case of ‘b’, the new user would remain a guest.

A guest can only check the availability of tickets and cannot block or buy tickets.

But a registered user can also act as a guest if he only wants to check the availability of tickets. ‘Availability of tickets’ always refers to viewing the flight schedule for given days, the price of tickets and any discount offers. The system shall present the user with an option to exit from the system at any time during the following processes.

***Registration and creation of user profile***

The system shall require a user to register, in order to carry out any transactions with it except for checking the availability of tickets. It will ask the user for the following information at the least – a user id, a password, first name, last name, address, phone number, email address, sex, age, preferred credit card number. The system will automatically create a ‘sky miles’ field and initialize it to zero in the user’s profile.

***Checking Availability***

After logging in a user (either a registered user or a guest), the system shall request him to enter the following details – origin city and destination city. “City’ is a generic term and refers to a city or town as the case may be. The origin and destination cities would be entered as text.

The system shall now refer to the flight schedule database and check if there is any ambiguity with the names of the cities. In case there are more than two cities with same name as entered by the user, the system shall list all of them (with more qualifications) and ask the user to select one of them. In case, either the origin or destination cities are not listed as being directly serviced by the airline, the system shall suggest the nearest city to which service is available, including the distance of the destination city from this nearest city.

After the origin and destination cities are ascertained, the system shall now access the flight schedule database and checks if there is a direct operational service between the two cities. If not, the system shall suggest possible routes and transfer points using a ‘route selection algorithm’. The user shall now be presented with a choice of either selecting one of the routes. In case he selects a route, the system shall fill in the intermediate stop over points and create a multiple trip itinerary for the user.

The system shall now ask the user to enter the following details - class, one-way or round trip, departure date and the number of adult passengers, children and senior citizens.

‘Class’ refers to business class/first class/club class/smoking/non -smoking. This choice shall be made by the user through a drop down menu indicating all the possible combinations of choices.

One-way/round trip shall be either a drop down menu or a check box selection. ‘Departure date’ refers to either a single date or a range of dates, entered through a calendar-like menu. This menu shall not show dates in the past or those dates that are too ahead in the future (as determined by the airline policy). In case, the trip is a round trip, the system shall also ask the user to enter the departure date on the return trip.

Having taken all the above input from the user, the system checks for any false entries like the departure date on the return trip being earlier than the departure date on the onward trip. In case of incompatibility, the system shall display a suitable error message and prompt the user to enter the information correctly.

Having taken all of the information as laid out above in 3.3.1 and 3.3.4, the system shall now access the flight schedule database and queries it using the input provided by the user.

The system queries the reservation database to check which of the flights on the schedule have seats available. The system displays the results in a suitable form (a tabular form) with the following information depicted – for each flight number – the flight number, departure time in origin city, arrival time in destination city, the duration of the flight (taking into account the possibility of a change of time zone) and the number of seats available on that flight.

There can be several flights between two cities and all of them will be listed for the particular date that the user wants to depart from the Origin City. In case, the user has entered a range of dates, the system shall display all the flights for all those dates in the range.

If the user has requested a round trip, the system shall display two tables - one for the onward trip and one for the return trip. There will be a check box in front of each line in the table representing a flight with available seats.

The user is now asked to check one of the boxes reflecting a choice of a flight number and time. In case of a round trip, the user is asked to check one box each in the two tables.

The system shall now display the price of the ticket for the trip. This will be the sum of the prices for all the members of the travel party being represented by the user.

The system shall also list any rules regarding the cancellation of tickets – what percentage of the price will be refunded within what date ranges. This will be displayed as a table.

***Making Reservations/Blocking/Confirmation***

After having taken the user through the step 3.3, Checking Availability, the system will now ask the user if he wishes to block/buy the ticket. If yes, and

1. if the user has been a guest, he will have to first register and become a registered user and then log onto the system.
2. If the user is already a registered user, and if he has logged on already, he can block/buy the ticket, but if he has been acting as a guest, he will have to log on.

Having ensured that the user is logged on validly according to 3.4.1, the system compares the departure date with the system date. If the departure date falls within 2 weeks of the system date, the system informs the user that he has no option to block the ticket and asks him if he would like to buy it.

If the difference between the departure date and system date is more than 2 weeks, the system asks the user if he would like to block or buy the ticket. The system informs the user that he can block the ticket at no cost now. It also informs him that if he chooses to block the ticket, he should make a final decision before 2 weeks of the departure date. The system shall send an email to the user, 3 weeks before the departure date as a reminder, in case he decides to block the ticket now.

Having taken the input from the user in 3.4.2, the system shall now proceed to update the reservation database. It will decrement the number of available seats on the particular flight for the particular class by the number of travelers being represented by the user.

In case of a blocking, the system makes a note of it in the database - to be used if the user doesn’t turn up before 2 weeks of the departure date. It generates a blocking number and displays it for the user to note down.

In case the user buys the ticket, the system accesses his profile and charges the price of the ticket to his credit card number. It simultaneously generates a confirmation number and displays it to the user for him to note down. The ticket has been reserved.

It adds the mileage of the trip (accounting for the number of travelers) to the sky-miles in his profile.

***Confirm Ticket***

A user who has earlier blocked a ticket after going through the steps 3.2 through 3.4, is required to either confirm the ticket before two weeks of the departure date or the ticket stands cancelled.

To let the user confirm a ticket, the system shall first log him on and ask for his blocking number. Then it accesses database and removes the check mark, which so far represented a blocked seat. The seat is now confirmed and reserved for the user.

The system accesses database and charges the price of the ticket to the credit card number of the user. It simultaneously generates a confirmation number and displays it for the user to note down. The ticket has been reserved.

It adds the mileage of the trip (accounting for the number of travelers) to the sky-miles in his profile.

***Reschedule Ticket***

The system shall present the user with an option to re-schedule his travel party’s trip. In order to do this, the system first logs on the user and requests his confirmation number. It will not allow a user to reschedule a blocked ticket but only a confirmed ticket. Using this, it queries database-reservation and presents the details of the trip to the user, including but not limited to origin city, destination city, date of departure and date of arrival (in case the trip is a round trip).

The system shall now ask the user to select new dates from the calendar-menu. It now goes through step 3.3.

In case, there are no available tickets for the dates entered, it displays a suitable message informing him that rescheduling to that date is not possible.

In case there are tickets available, the system asks the user to select the flight number for the trip (another for the return trip if the trip is a round trip) and proceeds to update the database.

The system accesses database and decrements the number of available seats on the flight(s) by the number of members in the user’s travel party. It then increments the entry for the previous flight by the same number to reflect an increase in the available seats on it as a result of the rescheduling.

The system now checks if there is any difference in the prices of the tickets. If so, it accesses database and charges or credits the credit card as the case may be. The system generates a new confirmation number and displays it to the user.

***Cancellation***

The system shall also give the user an option to cancel a confirmed ticket or a blocked ticket.

The latter case is simpler and will be dealt with first – the system shall first log on the user and request the blocking number. Then it accesses database and updates it by incrementing the number of available seats by the number of people in the user’s travel party.

In the former case, i.e., for a confirmed ticket, it asks for the confirmation number and accesses database and presents the details of the trip as in step 3.6.1.

It then lists the applicable rules for cancellation of tickets and depending on the system date and the departure date, it displays the % of the amount that would be refunded if the user cancels the ticket.

After the user cancels the ticket, the system generates a cancellation number and displays it for the user to note down. It accesses reservation database and updates it by incrementing the number of available seats on that flight by the number of travelers in the user’s party. It accesses user database and credits the refund amount to his credit card number. The system then deducts the mileage of the trip (taking into account the number of travelers in his party) from the sky miles in his profile.

***Update Profile***

The system shall enable the user to update his profile at any time. Changes can be made in fields including but not limited to address, phone number and preferred credit card number.

***View Ticket Status***

The system shall allow a user to view all information about his trip. After logging him on, it asks for his blocking number or his confirmation number. It accesses database-reservation and retrieves the details of the trip and presents them to the user in a convenient format, including any last minute changes to the flight timings etc. Such changes will be highlighted.

***Query Flight Details***

The system shall allow any user (registered or non-registered) to access the details about the arrival and departure times of a flight by requesting the user to input the flight number and date. The system accesses database-schedule and presents the time of arrival and departure.

### *Implementation Environment*

* HTML5/ CSS/ JavaScript
* Boostrap / Jquery
* Owl Carousel
* Google Chrome/ Mozilla FireFox/ Internet Explorer
* Methodology: Scrum
* Laravel

## User Stories

### *As a Guest, I want to*

#### Register

I can sign up to be Airline’s member to get the promotion from the Airline.

I can purchase a ticket that I can’t do as a guest.

#### Check flight availability

I can search for the suitable flight by origin, destination, class, type of travelling (one way or round trip), departure date and return date that match my need.

#### Pick a seat

I can take a seat where comfortable to me or I can choose the close seat to my friends when I go by party

#### Block a seat

I can keep the seat in case the airplane is going to be full when the trip is still on plan.

Or I want to make purchase later.

#### See the total price

I can know how much the ticket costs and consider it based on my budget.

#### Search a flight detail

I can have a detailed information about a flight for scheduling or tracking a flight.

#### Cancel blocked ticket

I can cancel the blocked seat if I change my plane.

### *As a registered user, I want to*

#### Sign in

I can sign in to check the flight status, get the promotion, purchase ticket,...

#### Check flight availability

I can search for the suitable flight by origin, destination, class, type of travelling (one way or round trip), departure date and return date that match my need.

#### Pick a seat

I can take a seat where comfortable to me or I can choose the close seat to my friends when I go by party

#### Block a seat

I can keep the seat in case the airplane is going to be full when the trip is still on plan.

Or I want to make purchase later.

#### Cancel blocked ticket

I can cancel the blocked seat if I change my plan.

#### Buy ticket

I can proceed purchase ticket including a new ticket and a blocked ticket to make sure for my plan Cancel purchased ticket

I can cancel the purchased ticket if I change my plan and get refunded refer to the Airline’s policy.

#### Reschedule

I can change to another flight when my plan is changed. The extra fee may be charge by the Airline

#### See the total price

I can know how much the ticket costs and consider it based on my budget.

#### See the ticket status

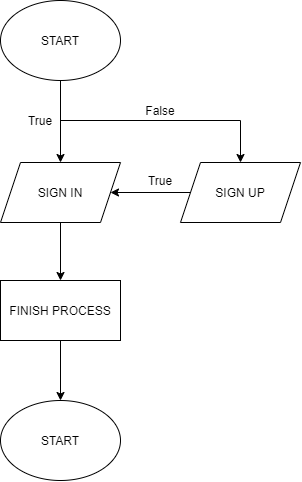
I can get the updated information about my flight whenever a change has been made by the Airline.

#### Search a flight detail

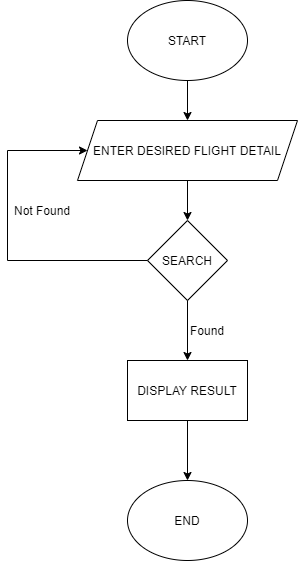
I can have a detailed information about a flight for scheduling or tracking a flight.

# Flowchart

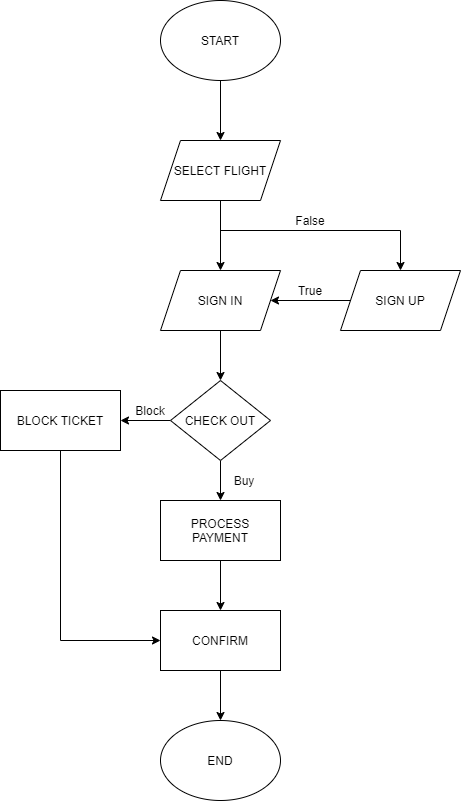
## Register



## Check availability



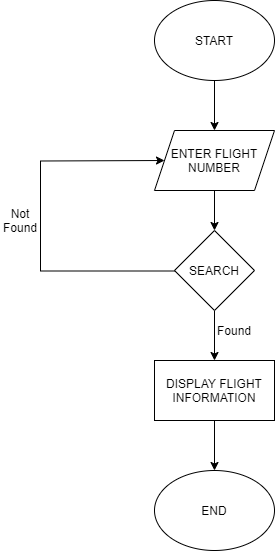
## Block or Buy ticket



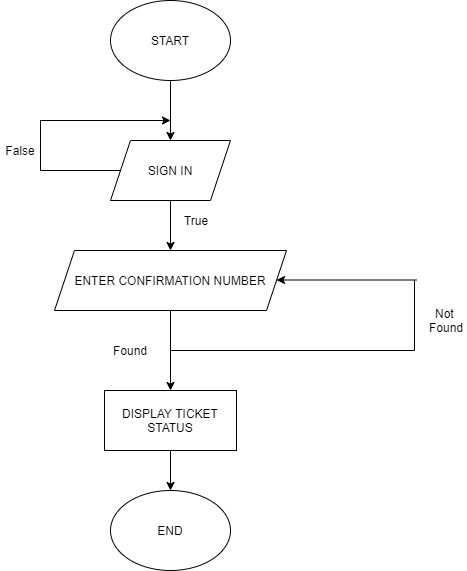
## Reschedule

## Cancellation

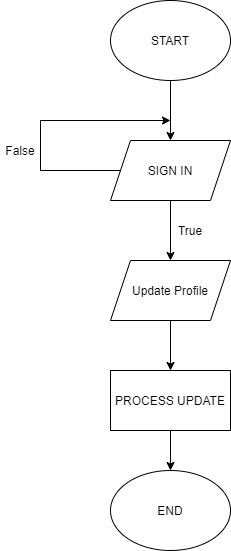
## Flight Details



## Ticket Status



## Update profile



# UI\UX

## Register

## Homepage

## New Arrival

## Men

## Women

## Promotion

## Blog

## Contact

## Compare Product

## Shopping Cart

## Checkout Product

### Address

### Delivery Method

### Payment Method

### Order Review

# Summary

|  |  |  |
| --- | --- | --- |
| **ID** | **Responsible** | **Task** |
| TH2002021 | Nguyen Tien Dat | Develop: Homepage, Men page, Women page, New Arrivals page |
| TH2004002 | Duong Xuan Binh | Develop: Product Detail page, Promotion page Support: JS shopping cart, Checkout Collect contents |