FLYAWAY – AN AIRLINE BOOKING PORTAL

This document contains the following.

• Project and developer details

• Sprint planning and tasks achieved

• Core concepts used in the project • Flowchart of the application

• Links to the GitHub repository

• Demonstration of product capabilities, appearance, and user interactions

• Unique Selling Points of the application

• Conclusion

Project and Developer details:

Project objective:

As a Full Stack Developer, design and develop an airline booking portal named as FlyAway. Use the GitHub repository to manage the project artifacts.

Background of the problem statement:

FlyAway is a ticket-booking portal that lets people book flights on their website.

The website needs to have the following features:

- A search form in the homepage to allow entry of travel details, like the date of travel, source, destination, and the number of persons.
- Based on the travel details entered, it will show the available flights with their ticket prices.
- Once a person selects a flight to book, they will be taken to a register page where they must fill in their personal details. In the next page, they are shown the flight details of the flight that they are booking, and the payment is done via a dummy payment gateway. On completion of the payment, they are shown a confirmation page with the details of the booking.

For the above features to work, there will be an admin backend with the following features:

- An admin login page where the admin can change the password after login, if he wishes
- A master list of places for source and destination
- A master list of airlines
- A list of flights where each flight has a source, destination, airline, and ticket price

The goal of the company is to deliver a high-end quality product as early as possible.

The flow and features of the application:

- Plan more than two sprints to complete the application
- Document the flow of the application and prepare a flow chart
- List the core concepts and algorithms being used to complete this application
- Implement the appropriate concepts such as exceptions, collections, and sorting techniques for source code optimization and increased performance

You must use the following:

- Eclipse/IntelliJ: An IDE to code for the application
- Java: A programming language to develop web pages, databases, and others
- SQL: To create tables for admin, classes, students, and other specifics
- Maven: To create a web-enabled Maven project
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the application code and track its versions
- Scrum: An efficient agile framework to deliver the product incrementally
- Search and Sort techniques: Data structures used for the project
- Specification document: Any open-source document or Google Docs

The following requirements should be met:

- The source code should be pushed to your GitHub repository. You need to document the steps and write the algorithms in them.
- The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link to the repository. You can add a section to your document.
- Document the step-by-step process starting from sprint planning to the product release.
- Application should not close, exit, or throw an exception if the user specifies an invalid input.
- You need to submit the final specification document which includes:

o Project and developer details

o Sprints planned and the tasks achieved in them

o Flowchart of the application

o Core concepts used in the project

o Links to the GitHub repository to verify the project completion

Developer Details:

Mahendra Kumar Singh

mahendrakumarsingh9893@gmail.com

Spring planning and Task completion:

Sprint 1:

I. Create a welcome page or home page displaying FlyAway and name of developer.
II. Take user input for filling source and destination, date of travel, no. of travellers etc.
III. Clicking on Admin login tab will redirect to admin login page for asking input as username and password.

Release product for feedback over the frontend part of home page and admin login page.

Sprint 2:

After successful admin login -

I. Add options to admin login page for further activities like
    a. Updating list of all the flight airlines.
    b. Change password.
        i. Add webpage to set up new password.
    c. Updating list of all the Flights source and destination.
    d. Display consolidated flight details with flight details.
II. Create webpage for the above-mentioned activities and interlink them as necessary.

Release product for feedback with all added webpages.

Sprint 3:

Create a program to store all the inputs given by the admin to a database (assigned to the FlyAway). Develop relation between frontend developed in sprint2 with the database to store and retrieve data dynamically.

Release the product for feedback with all required database activities.

Core concepts used in the project:

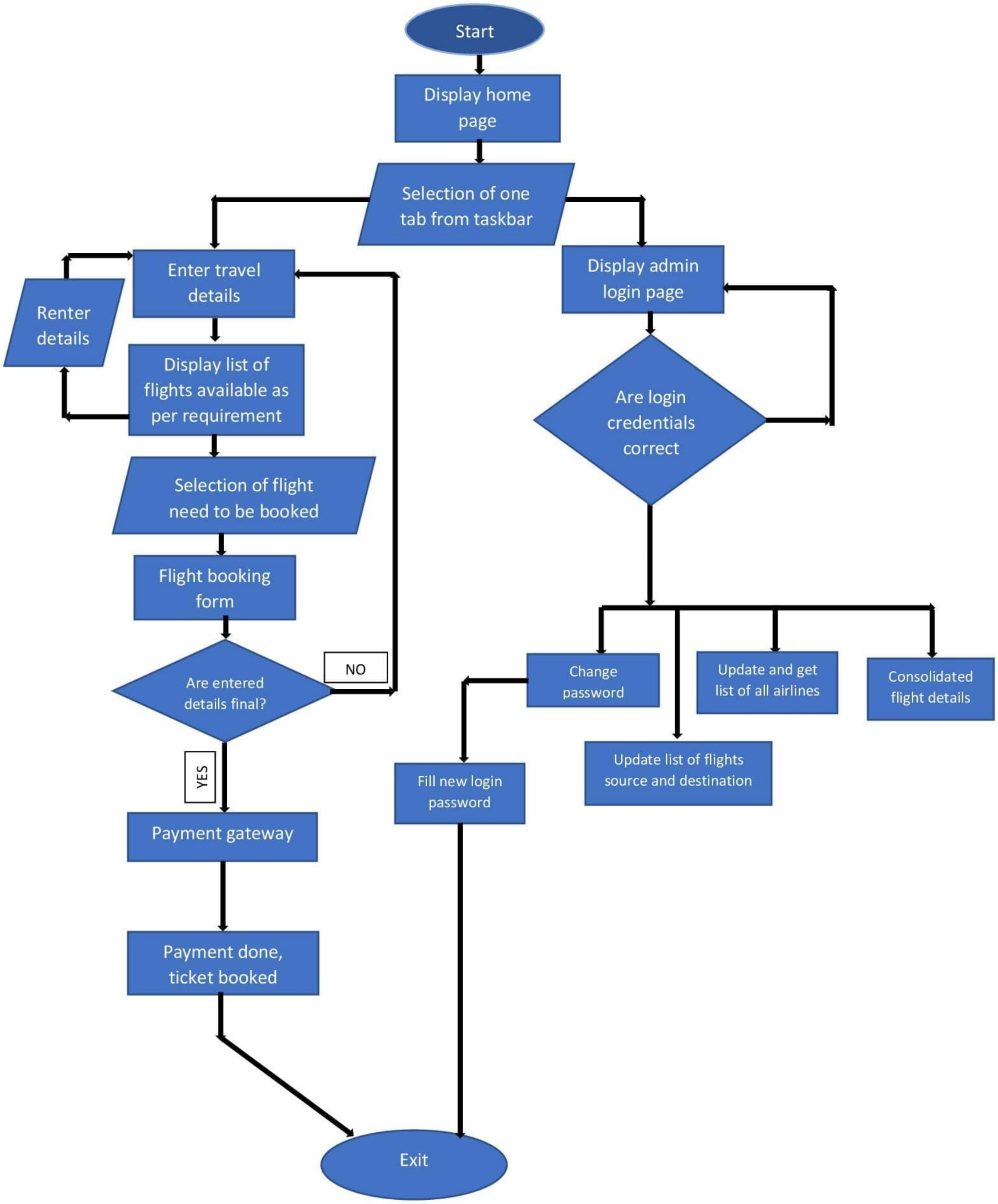
- CRUD Operations
- Exception Handling
- Authentication and Authorization
- Session Management • Frontend technologies like HTML, CSS

- Java Server Pages (JSP)

Links to the GitHub repository:

https://github.com/Mahendra1272/FlyAway.git

Flow Chart


Start
Display home page
Selection of one tab from taskbar
Enter travel details
Renter details
Display list of flights available as per requirement
Selection of flight need to be booked
Flight booking form
Are entered details final?
NO
YES
Payment gateway
Payment done, ticket booked
Exit
Display admin login page
Are login credentials correct
Change password
Update and get list of all airlines
Consolidated flight details
Update list of flights source and destination
Fill new login password

Flow of Application:

1. Welcome screen with name of website owner and developer.
2. Ask for user input to fill travel details like source and destination location, date of travel or select admin login tab from the taskbar present at the top.
    1. If user put travel information and submit it, using these details display flights available on the screen for user to choose one of these flights.
    2. Once user select a flight, show total price to be paid and flight details on screen, confirm the entries and forward to payment gateway.
3. Open admin login page to ask for login credentials, on successful login credentials redirect the client to further set of webpages else display the login page again. Take user input of admin to choose from the following activities by clicking on the desired tab from the webpage:
    1. Change user password.
        1. Display a window to take new user password details.
    2. Updating or getting list of all flights source or destination.
    3. Updating list of all the classes.
    4. Retrieving consolidated flight details with name to and fro station date and price altogether.
4. For each of the above-mentioned choices interlink the webpages.
5. Store or retrieve the data into the database or from the database of the FlyAway as input given by the client/admin in this case.

Pushing the code to GitHub Repository:

- Open Git Bash and navigate to the folder where you have created your files.

  cd Simplilearn\Spring Tool Suite\Phase2SecondProject\FlyAway

- Initialize the repository using the below command **git init**
- Add all the files to your git repository using the below command **git add .**
- Commit the changes using the below command **git commit -m “Initial commit”**
- Add the URL for the remote repository where your local repository will be pushed **git remote add origin https://github.com/ Mahendra1272/Fly-Away.git**
- Push the files to the folder you initially created using below command **git push -u origin master**

Conclusion:

An application with a sprint duration of three sprints has been developed in the programme. This application manages the FlyAway data. The user can create account and search or book a flight. The administrator can manipulate the flights data by logging in with an email ID and password.

This application will be improved further by adding following features:

➢ Form/Input Validation

➢ Better CSS/Frontend Styles

➢ Modify available flights data

➢ Show previous flight bookings

➢ Ticket Payment