**FlyAway (An Airline Booking Portal)**

**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 the web pages, databases, and others  
● SQL: To create tables for admin, airlines, 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

**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 it.  
● The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository. You can add a section in your document.   
● Document the step-by-step process starting from sprint planning to the product release.   
● The 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 will include:   
● Project and developer details   
● Sprints planned, and the tasks achieved in them   
● Algorithms and flowcharts of the application   
● Core concepts used in the project   
● Links to the GitHub repository to verify the project completion

GitHub REPO:  
https://github.com/catiafsantos/PhaseIIFlyAwayAirlineProject

* All relevant information, such as this file, print screens, the code, etc... is provided on this REPO.

Project Name: PhaseIIFlyAwayAirlineProject

Project Developer: Cátia Santos ([catia.santos3@vodafone.com](mailto:catia.santos3@vodafone.com))

Company Name: Vodafone

**Sprint Planning:**

The following information is not accurate, the project and the tasks did not take the time that is being mentioned bellow, the following is being provided as data to implement the Agile knowledge acquired during the course.

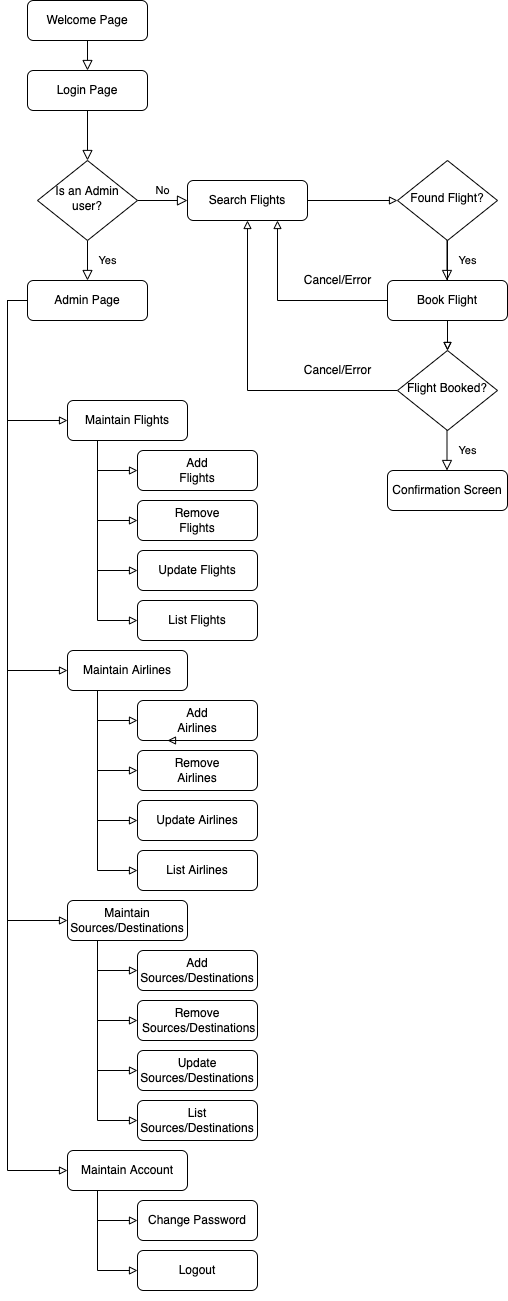
* Hereby the Sprint Tasks and estimates in story points, let’s assume that the average is 10 SPs per sprint:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Task Code + Title** | **Simple Task Description** | **Estimation (SP)** |
| 1 | TASK-134 – Env Prep | Download the required tools, such as IDE, Java, JDK, Git, GitHub etc.  Install the software | 1 |
| 1 | TASK-156 – Project Design | Design the application flowcharts, algorithms and behaviors and document them. | 2 |
| 1 | TASK-204 – Entity Model | Create MySQL Connection DB. | 1 |
| 1 | TASK-205 – Entity Model | Create the Login Model and DB. | 1 |
| 1 | TASK-206 – Entity Model | Create the Customer Model and DB. | 1 |
| 1 | TASK-207 – Entity Model | Create the Place Model and DB. | 1 |
| 1 | TASK-208 – Entity Model | Create the Airline Model and DB. | 1 |
| 1 | TASK-209 – Entity Model | Create the Payment Model and DB. | 1 |
| 2 | TASK-210 – Entity Model | Create the Flight Model and DB. | 2 |
| 2 | TASK-211 – Entity Model | Create the Order Model and DB. | 2 |
| 2 | TASK-352 – Webapp | Create the Logout Servlet and JSP | 1 |
| 2 | TASK-353 – Webapp | Create the Login Servlet and JSP | 2 |
| 2 | TASK-354 – Webapp | Create the Admin Servlet and JSP | 3 |
| 3 | TASK-355 – Webapp | Create the Register Servlet and JSP | 3 |
| 3 | TASK-355 – Webapp | Create the List Flights Servlet and JSP | 3 |
| 3 | TASK-356 – Webapp | Create the Search for Flights Servlet and JSP | 3 |
| 3 and 4 | TASK-357 – Webapp | Create the Checkout page Servlet and JSP | 3 |
| 4 | TASK-358 – Webapp | Create the Confirmation page Servlet and JSP | 3 |
| 4 | TASK-359 – Webapp | Create the Change Password page Servlet and JSP | 3 |
| 5 | TASK-724 – Testing | Test the code functionalities - test all inputs and options. | 5 |
| 5 | TASK-1200 – Documentation | Document the test behaviors with examples.  Document the algorithms used. | 3 |

So, the conclusion is that it will take around 4 full sprints and 8 more SPs (story points) which is roughly half more sprint, therefore 4/5 sprints.

Flowchart:

* Flowchart was designed used diagram.io online:



Technologies/Tools used:

The following technologies were used on the creation of this project:

* Java 11 – OpenJDK Temurin 11 + SDK 11
* Git version 2.31.1
* GitHub
* Mac OS Monterey Version 12.2.1
* IDE – IntelliJ IDEA 2021.2 Community Edition
* Microsoft Word for Max OS X – Version 16.57
* Diagram.io Online - <https://app.diagrams.net/>
* TomCat 9.0.58
* Hibernate
* Maven
* MySQL conector 8.0.28
* MySQL Workbench 8.0
* HTMP,CSS and .JSP

Examples of the application running can be found into the GitHub Repo on the Documentation and Examples Folder.

Comments:

This project is not fully finished from my side, and I don’t know if I can fully finish it without help, there was a big gap of information, between what we’ve learnt in the sessions and what is being requested on this project.

The DB and Models are done, most Servlets and Pages also started at least, do let me know, what is the way forward.

Don’t take me wrong, the trainer is fantastic, and has great knowledge, but for sure wasn’t aware of what we were requested to do in the Project, until the last day, after we requested her to check the project with us.

Without mentorship and previous knowledge is very hard to fulfill the requests you’ve made on the project, since we learn the concepts but not how to join them in the ways you require here, I will continue adding to the GitHub repo.

END