|  |
| --- |
| **PHASE 2-PROJECT**  **FLY AWAY**  **AN AIRLINE BOOKING PORTAL**  **(Specifications and Scrum Details)** |

**The sections in this document are:-**

|  |
| --- |
| 1.Project details and Developer details |
| 2.Sprint Planning and Task Completion |
| 3.Algorithms and Flow Diagram of the Application |
| 4.Core concepts used in the Project |
| 5.Link to the Git Hub repository |
| 6.Demonstrating the Project Capabilities, appearance, and user Interactions |
| 7.Unique selling points of the Application |
| 8.Conclusions |

**The code for this project is hosted at**

|  |
| --- |
| **Git-Hub link:-**  https://github.com/swathikanduri7/Fly-Away-Airline-Booking-Portal.git |

**The project is developed by Swathi Kanduri**

**1.Project Details:-**

|  |
| --- |
| FlyAway is a ticket-booking portal that lets people book flights on their website  **The user can do the following:-**  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.  **The admin user can do the following:-**  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.  **Developer Details:-**  **This project is developed by Swathi Kanduri** |

**2.Sprint Planning and Task Completion**

|  |
| --- |
| The project is planned to be completed in 2 sprints. |

**Tasks completed in this sprint are**:-

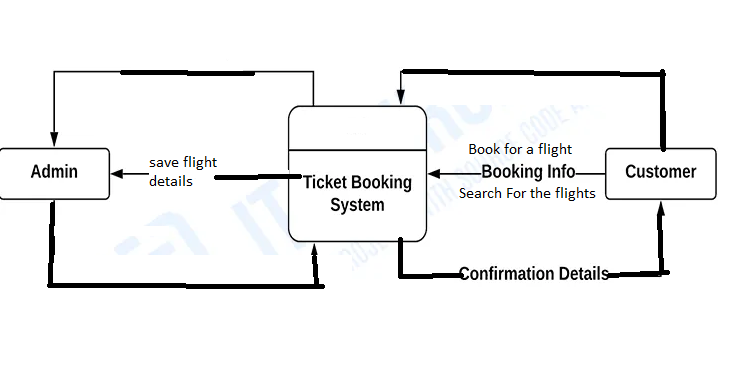
|  |
| --- |
| * Creating the flow of the application. * Initializing the git repositories to track changes as development progresses. * Configuring Web Application using Maven tool * Configuring Hibernate utility to manage data base connection and perform data base actions * Completed below stories   Admin Login to login as a admin  Develop a page to save the Flight information by Admin  Develop a page to login as a user  Develop a page to register new user  Develop a page to search flight details availability for booking  Develop the pages to handle successful/failure search  Develop a page to book a flight  Develop successful registration page     * Testing the program with different types of user input. * Pushing the code to Git Hub. * Creating this specification document highlighting application capabilities, appearance and user interaction. |

**3.Algorithms**

|  |
| --- |
| **Algorithm for users for flight booking**  **Step1:-Register as a new user**  **Step2:-After successful registration login to the webportal**  **Step3:-Provide flight details and search for the flight**  **Step4:-If Flight available, then user can book the flight, If Flight not available,**  **then go to step 3 and search for other flights**  **Step5:- After successful booking will display the message as flight booked successfully.** |

|  |
| --- |
| **Algorithm for admin for flight booking**  **Step1:-Provide login details and click on submit button**  **Step2:-after successful login it will display flight details form, to save the flight**  **details**  **Step3:-admin can provide the flight details and click on submit button**  **Step4:- After successfully saving the details to the database continue with the**  **step 2.** |

**Flow Diagram of the Application**

****

**4.Core concepts used in the project are:-**

|  |
| --- |
| * **IDE: Eclipse is used to code for the application** * **Core Java:-A programming language to develop web pages, databases and others.** * **Databases:-Mysql.** * **WebTechnologies:-HTML,CSS,JSP** * **J2EE:-Servlet/Jsp are used to collect data from the users to create the web pages dynamically and present the results** * **SVN Tools:-Git-Hub** * **Scrum:-An efficient agile framework to deliver the product.** * **Specification document:-Ms-word is used for specifications for open-source documentation.** |

**5.Link to the Git Hub Repository:-**

|  |
| --- |
| **Git-Hub Link:-**  **https://github.com/swathikanduri7/Fly-Away-Airline-Booking-Portal.git** |

**6.Demonstrating the product capablilites, appearance and user interactions:-**

**The user can do the following:-**

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.

**The admin user can do the following:-**

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.

**7. Unique Selling Points of the Application:-**

1. The application is designed to keep on running and taking user inputs even after  exceptions occur. To terminate the application, appropriate option needs to be  selected.

2. The application can take any file/folder name as input. Even if the user wants to  create nested folder structure, user can specify the relative path, and the application  takes care of creating the required folder structure.

3. User is also provided the option to write content if they want into the newly created  file.

4. The application doesn’t restrict user to specify the exact filename to search/delete  file/folder.

5. They can specify the starting input, and the program searches all  files / folder starting with the value and displays it. The user is then provided the option to select all files or to select a specific index to delete.

6. The application also allows user to delete folders which are not empty.

7. The user is able to seamlessly switch between options or return to previous menu even after any required operation like adding, searching, deleting or retrieving of files is performed.

8. The application is designed with modularity in mind. Even if one wants to update the  path, they can change it through the source code. Application has been developed  keeping in mind that there should be very less “hard coding” of data.

**8.Conclusions:-**

|  |
| --- |
| Further enhancements to the application can be made which may include:   * Conditions to check if user is allowed to delete the file or add the file at the  specific locations. * Asking user to verify if they really want to delete the selected directory if it’s not  empty. * Retrieving files/folders by different criteria like Last Modified, Type, etc. • Allowing user to append data to the file. * Admin can manage flight details to offers better productivity and hence more progress towards development. |

**Pushing the code to Git Hub Repository**

* Open your command prompt and navigate to the folder where you have

created your files.

**cd <folder path>**

* Initialize repository using the following command

**git init**

* Add all the files to your git repository using the following command

**git add .**

* Commit the changes using the following command

**git commit . –m<commit message>**

* Push the files to the folder you initially created using the following command

**git push -u origin master**