1. **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.

1. **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.

1. **Application Name**

FlyAway (An Airline Booking Portal)

1. **Developer Details**

Ashwin Mathankar, 216 Java SL (Evening Batch), Phase-2 Final Assessment

https://github.com/ashwinmathankar/Phase-2-Projects.git

1. **Sprint Planning**

To complete this application, three sprints were planned to develop different aspects of the web-based application.

**Sprint 1 –** Designed the index page which displays three links, Admin Login for admin login, Customer Login for customer login and Info page for Developer Details. In admin login, user is asked to provide username and password which are then validated with the username, password stored in the database. If wrong details are given, then an error message is displayed and user is asked to login again. After successful login, a Dashboard page is shown with seven links, display Flight Details, add new flights, display Admin Details, display Customer Details, display Order details, change password and Logout.

To display Flight Details, the values are taken from the database using Hibernate and displayed in a table format. Along with that, each flight detail has two additional links, delete or update. Delete will delete that particular flight details from the database using Prepared Statement and for Update, user is asked for new flight details and those details are then updated into the database using Prepared Statement.

To add new flights, user is asked for appropriate details which are inserted into the database using Transaction and then committed. These new flights will also show up in Flight Details list.

For Admin Details, the values are simply taken from the database using Hibernate and then displayed in a table format. This same method of taking values from the database using Hibernate and then displaying it in a table format is also followed for displaying Customer Details and Order Details.

To change password, user is asked for the username and new password. As this is the admin side, the user can change password for any admins present in the database, provided the admin username is known the user. The new password is then updated into the database and the user is asked to login again.

Finally for logout, the user is redirected to the login page.

**Sprint 2 –** In customer login, the customer is asked to provide username and password. If the user is not registered, then that user needs to click on the link present in the login page. During registration, user is asked for details which are then stored into the database using transaction and committed. Then the user is asked to login with new credentials. After user provides the username and password, it is validated with the values in the database. If wrong details are given, then an error message is displayed and user is asked to login again. After successful login, a Dashboard page is shown with three links, Profile page for the user, Book Flight and Logout.

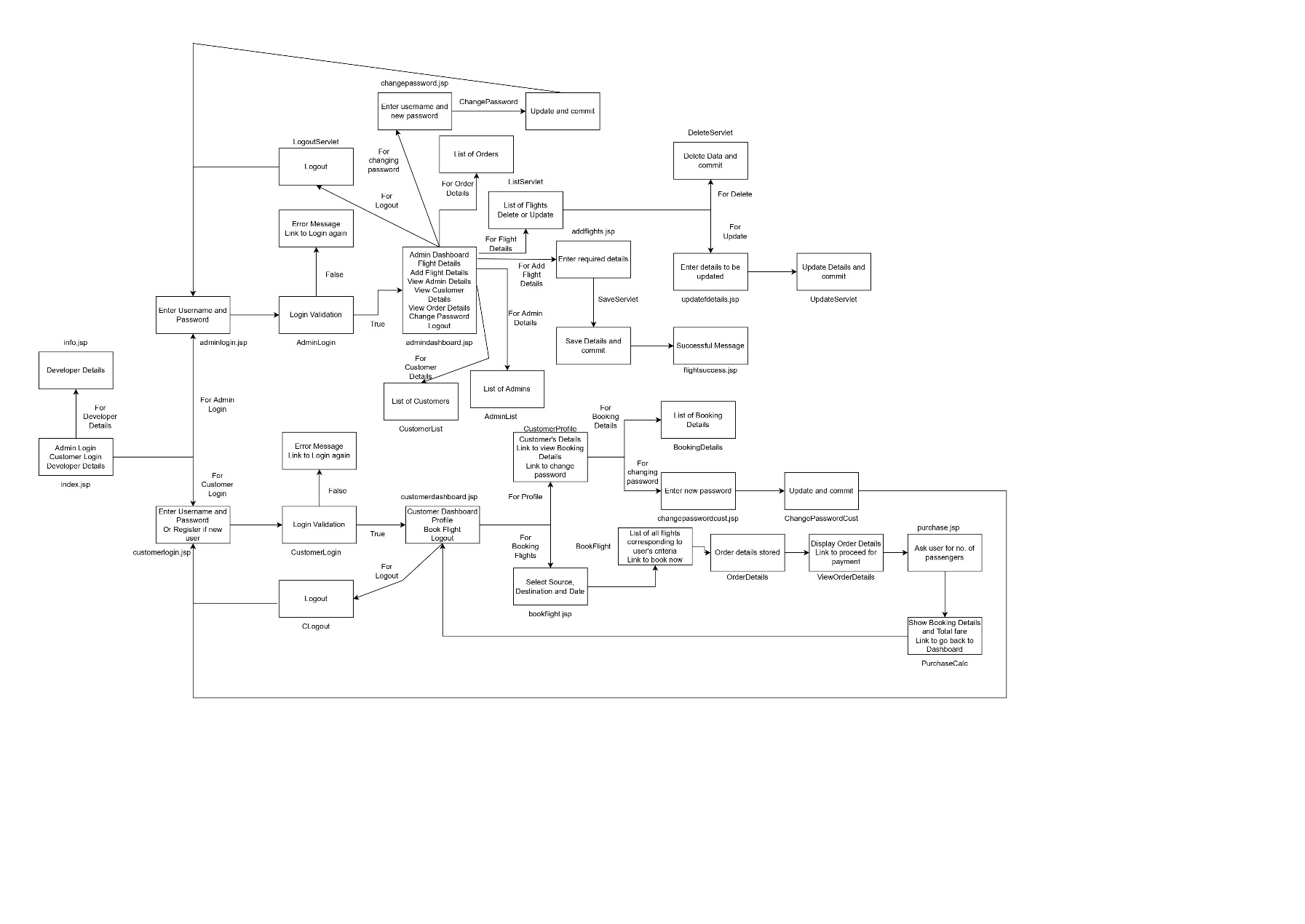
In the Profile page, the user is shown their details which are taken from the database. A Update link is also given beside the details. Using update, the user can update all their details except their password. After providing new details, these are then updated into the database and these new details are then shown in the profile page. The profile page also contain two more links, Booking Details and Change password. Booking Details shows all the past booking details of that user, along with new or upcoming booking details. To change password, user is asked for the new password which is then updated into the database and the user is asked to login again.

**Sprint 3 –** For flight booking, the user is asked for source, destination and date of travel. Taking these criteria in a query, the user is then shown a list of flights available according to the given criteria. Also, a Book Now link is displayed beside the flight details. After the user clicks on Book Now, Order Details are displayed and the user is asked to proceed with payment. The user is then asked for the number of passengers, and using that number, the total fare is calculated. The user is then shown the final Booking Details along with the Total Fare. These booking details are also displayed in the profile of the user.

Finally for logout, the user is redirected to the login page.

All the sprints were of 1 hour or 60 minutes each.

1. **Flow Chart**



1. **Core Concepts Used**
2. Database and Table creation done in MySQL.
3. Front end created using JSP pages and back end is created using servlets.
4. The database is mapped using hibernate config and it is used through Hibernate Util.
5. Total four tables were created and those four tables were mapped with servlets using Entity classes.
6. Displaying values from a table is done using Hibernate.
7. Update and Delete functionality for the values of a table are developed using Prepared Statement and the updated values are then committed into the database using Transaction.
8. Three tables, namely CustomerDetails, FlightDetails and OrderDetails are mapped into the Order Entity class using Hibernate Annotation. The relation between the tables are as follows – One-to-one mapping between OrderDetails and FlightDetails, Many-to-one mapping between OrderDetails and CustomerDetails.
9. **Conclusion**

This web-based application is designed using JSP, Java Servlets and Hibernate. MySQL is used for database and table creation. The front end can be improved using CSS so the design aspect will be more attractive.

**Unique Selling Points (USPs)**

1. User friendly interface helps users to guide through different options without any hassle.
2. Adding flights, changing password, or updating details can be done smoothly.
3. Admins have control over all the aspects of the application.
4. Customers can seamlessly book flights and check on those details later as well.
5. Customers can also check all their previous and upcoming booking details under one section.