FlyAway (An Airline Booking Portal)

Phase -2 Project

Project and Developer details :

The code for this project is hosted at *https://github.com/Aman9392/Flyaway_project* Project is developed by Aman Pawar

Sprint Planning :

This project is intended to be delivered in three sprints.

Sprint 1: Analysed the application's features and prepared a flow chart and Git Repository. Implement the database schema for admin, airline, flight, place, user, booking.

Sprint 2: Create basic CRUD operations for admins, users, airlines, flights, booking, places. Implement the Search Flight feature to display information about flights available based on a specific criteria selected by user.

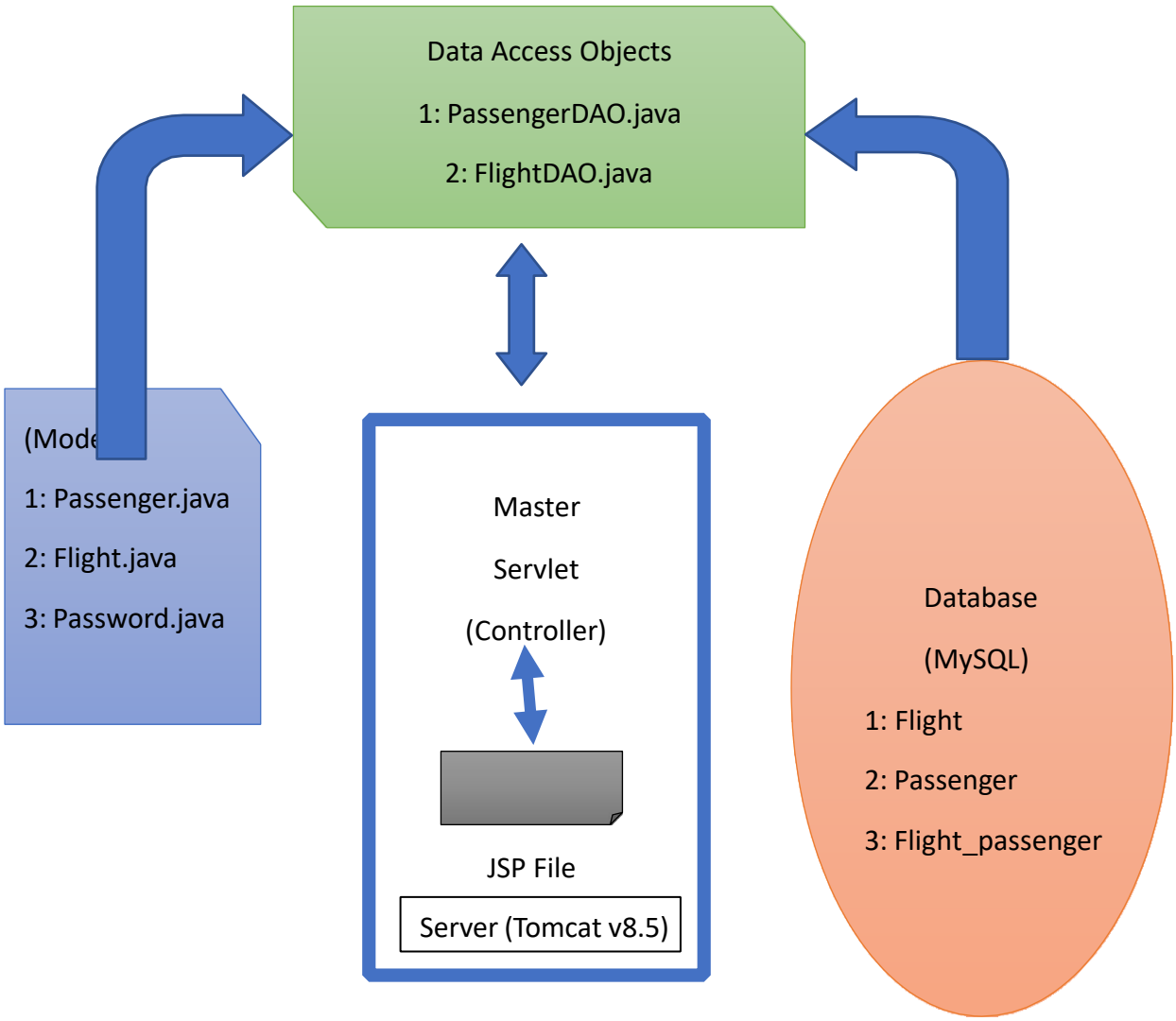
Sprint 3: Tested the application numerous times to ensure a high-end quality product and pushed it to GitHub. Prepared this document highlighting the application's capabilities, appearance and user interactions.

Core Concepts :

The concepts used in the project are:

- CRUD Operations
- Exception Handling
- Authentication and Authorization
- Session Management
- Frontend technologies like HTML, CSS
- Java Server Pages (JSP)

Flow Chart :



GitHub Repo :

The code for this project is hosted at *https://github.com/Aman9392/Flyaway_project*

Inferences :

- Mainly got to learn about the sql connection with my application and how can manipulate the data.
- Using bean classes and servlets. Using techniques like exceptions, collections, and sorting techniques for source code optimization and increased performance

Conclusion :

1: The prototype is robust and platform independent.

2: User can easily use the prototype and safely exit out of it.

3: As a developer, we can enhance it by introducing several new features such as guards along each web pages as currently its statically connected with each along with backend as will not allow to go back once admin has been logout, routing, custom validators and can have more user-friendly by adding styling (CSS, Bootstrap), custom loaders.

4: Though this prototype is tightly connected, the data will only persist in database until server is running and gets reset with each restarting of sever because of manual configuration of hibernate.

5: This prototype can also be implemented with multithreading to enable better performance.

6: And lastly, this prototype can be upgraded by implementing with securities patches to make it more versatile and secure in both local environment and global and later can be configured dynamically with connection of database through hibernate.