**Software Requirements Specification**

**(SRS) Document**

**< Project :- Flyaway>**

**<Date of completion :- 01-07-2022>**

**<By- Mohnish Pawar>**

1. **Introduction**
2. **Purpose**: The goal of flyaway flight booking system is to provide airlines with a prototype of how they can go about preparing an online flight management system. Although it might not be a complete prototype, it can give the airlines a blueprint to start working on an online flight management system.

1. **Intended audience**: - This document is intended for anybody from the airline industry looking to make an online flight management system.
2. **General description**

* **Product Perspective**: The idea of the flyaway project came into mind looking at the current advances made in technology and the need to provide a prototype to

said individuals interested in airline companies along with various airline companies to enable them to start an online flight management system of their own.

* **Product features**: The primary operations performed by the flyaway project are booking flights, providing an administrator control over the various operations such as adding airlines, deleting an existing airline, adding places which would be connected by flights, removing existing places which are connected by flights, adding flights for connecting places. It also provides an option for the administrator to change his password. A brief description of the features is mentioned below: -

1. **Booking flights**: - This operation books a flight between two places (source id - the place from where you are booking the flight and destination id – the place where you intend to reach after boarding the flight.)

* In case of no flights, connecting the source and the destination id, a message conveying the same is displayed.
* In case of flights being present which connect the source and the destination id, the agent is forwarded to a page where he can book a flight by choosing the airline (in case of options). On the same page, the agent comes to know the price of 1 ticket as well as the total amount to be paid in case of multiple passengers.
* Afterwards, the agent is forwarded to a page where his personal info is collected along with the details of the flight being displayed.
* Afterwards, the agent is forwarded to a dummy payment gateway where the user enters his details along with the credit card number.
* Finally, the user is thanked with a message showing his booking to be confirmed along with the user’s credit card number which would be used for payment.

A central administrator is the one who controls all the aspects of the project. For that to happen, he attempts to login in with the help of email and password authentication. If the administrator enters the correct details, he is directed to a navigation bar where he can take care of all the operations. If the administrator, enters incorrect details, a message conveying the same is displayed.

1. **Change password:** - The administrator can change his password if he wishes to with the help of this option. The administrator enters the new password wherein then he is forwarded to a message which conveys the password to have changed.
2. **Edit Airlines:** - With the use of this option, the administrator adds or removes airlines. If the admin wants to add an airline, then the admin needs to just mention the name of the airline which he wants to add, and it’s added with the click of a button. Similar is the procedure of removing an airline, the agent can remove an airline by just clicking on the delete button next to the airline which needs to be removed.

1. **Edit Places:** - With the use of this option, the administrator adds or removes places. If the admin wants to add a place, then the admin needs to just mention the name of the place which he wants to add, and it’s added with the click of a button. Similar is the procedure of removing a place, the agent can remove an airline by just clicking on the delete button next to the place which needs to be removed.
2. **Edit flights:** - With the use of this option, the administrator adds or removes flights. If the admin wants to add a flight, he needs to enter details such as ids of the places to which he wants to connect (source and destination id), the airline which he wants to make the flight and finally the price of the ticket per passenger. Just like that with the click of a button, the flight is added.

Similar is the procedure of deleting a flight, the agent can remove a flight by just clicking on the delete button next to the place which needs to be removed.

* **Technical requirements for running the project**: - To run the flyaway project on a system, the system should have JDK installed (mandatory) along with an IDE like eclipse also running (not mandatory). Also needed are jar files for jstl, mysql, javax.api… (mandatory). If you want to avoid downloading the jar files, then we can use Maven and mention the dependencies accordingly as per the jar files. Also, RDBMS such as mysql, oracle
* **Assumptions and dependencies**: - To run the flyaway project, you need to have a basic understanding of English as well as some basic logic regarding the features along with knowledge of basic sql queries.

1. **Flowchart**

The flowchat for flyaway is mentioned on the next page

Diagram

Description automatically generated

1. **Concepts used in the project**
2. Servlets used for backend logic
3. Server :- Apache tomcat server 9.0 for running the web application.
4. Jsp, HTML, a bit of CSS and a bit of bootstrap were also used for front-end development
5. Constructors, getter setters, prepared statements, arraylists, resultset, exception handling, string methods, for loops, etc were the java concepts used.
6. Mysql server for database
7. Jdbc for database connectivity
8. Separation of servlets, model classes and db