



**NANYANG TECHNOLOGICAL UNIVERSITY**

**Individual Project-1  
Airline Booking System**

**FUNCTIONAL SPECIFICATION**

<b>COURSE TITLE</b>	: CI6225 – ENTERPRISE APPLICATION DEVELOPMENT
<b>LECTURER</b>	: Mr. HARIHARAN VAIDHYANATHAN
<b>STUDENT</b>	: Adam Myrén

## **1. Introduction**

This document serves as the functional specification for the Airline Booking System that has been developed as a part of the course CI6225 – Enterprise Application Development. It will introduce the reader to the most important aspects and operations of the system and describe responsibilities as well as the limitations that the system has.

## **2. Background**

This airline booking system will be able to run on web servers supporting the latest Java Runtime Environment (JRE). The persistence handling mechanism is using a mysql database. This makes the system OS-independent.

Furthermore, the coding need to be of high quality and the EJB framework must be used.

## **3. Overview**

The Airline Booking System developed is to be used as a tool for airlines that which want to handle their customer interactions, such as bookings and customer registrations, and manage their routes via a web interface. It provides admin functionality, such as being able to retrieve a list of all customers. Except being able to book new flights, customers need to be able to view and manage all their previous bookings. Airline personnel needs to be able to add new flights and modify existing flights.

## **4. Operations**

### **4.1 Book a new flight ticket**

A user must be able to book a flight through the developed system. The user will search for flights on a specific date and thereafter choose what flight or flights to book. A user needs to be logged in in order to book a flight, this is to ensure that the user can access his or hers tickets and manage bookings later on. When the user has logged in, or created a new profile, he or she must enter all relevant information about the passengers. Thereafter, the customer must be able to review all details about the booking before entering payment details.

If a passenger at any stage enters invalid information, or the system is not able to book any of the passengers on the specified flight, the user must be notified and prompted to change any invalid information.

### **4.2 Check and manage booked flights**

To check an existing booking, the user must log in at the start page and is then directed to a profile page where he or she can either choose to book a new flight but also see all previous bookings. At this stage the user can also choose to cancel a booking. When a flight is to be cancelled to user must confirm the cancellation by pressing an additional button.

### **4.2 Create list of passengers on a specific flight**

To be able to check what passengers are booked on a specific flight, one must be logged in as a user with admin privileges. This will direct the admin user to a page where the admin can enter a flight id and date in order to get a list of all passenger on the specified flight. If there are no passengers booked on the specified flight, the list will be empty and the user can simply press the back button to make a new search. The list can also be downloaded as a xml file.

### **4.3 Create list of customers**

As no information is needed from the user to fetch all the users from the system, the admin user only has to press a button in order to retrieve a list of all active users on the system. The list can also be downloaded as a xml file.

#### **4.4 Edit an existing flight**

To be able to edit a flight, one must be logged in as a user with admin privileges. First, the admin must enter a flight number and a date for the flight to be edited. The admin will then be presented with all information of that flight and can choose to edit any field before saving the flight in the database. If a user tries to search for a flight that does not exist, the user should not be presented with a result and if the user tries to change a piece of information to an invalid value the user should be notified and prompted to reedit or cancel the edit.

#### **4.5 Add new flight**

As with creating a passenger report or editing a flight, the user needs to be an admin in order to add a new flight into the system. The user has to enter all flight information in the add flight box at the admin page and then press “add flight” to save the flight in the system. If not all information is entered or if any information is invalid the user must be prompted to reenter the information.

#### **4.6 Add Promotion**

By entering an origin airport and destination airport, an administrator can add a promotion on the route. That means that the route will be displayed on the start page, in order to promote sales. The cheapest available price on that route will automatically be displayed.

#### **4.7 Delete Promotion**

All current promotions will be displayed on the admin page, simply by pressing the “delete” button an administrator can remove a promotion from a route.

#### **4.8 Add new administrator**

Any logged in administrator can create a new user with administrator privileges. This is done by filling out user information button on the dedicated box on the admin page.

### **5. Human Interfaces**

The users interact with the system through a web browser. The application is designed to support user interface in all major browsers, it is however primarily tested using Google Chrome.

### **6. Data Models**

Each booking is related to exactly one user, the user that was logged in when the booking was performed. A user however, can have many bookings and a user is either an admin user or a regular user. A flight keeps track of how many available seats that it has, and each time a booking is made or cancelled on that flight the value is either decreased or increased. Each booking is related to one passenger and one user, the user and the passenger does not have to be the same person as the user is only responsible for managing the booking.

### **7. Robustness and Reliability**

When a user puts invalid information causing an error in the database connection or any other layer in the system, the user should be properly informed and able to correct the error by retyping the input. If an unexpected error arises the system should provide an explanation to why the error occurred so that it can be corrected in future releases. HTML verification has been used to prompt the user to not leave any blank fields where input is required.

### **8. Configuration**

The system should be able to run on any OS that supports J2EE6 and JRE7 and Mysql database server. For all testing of the application Glassfish 4 has been used.

## **9. Future Expansion**

Due to time constraints there are still functionality to be added.

- Allow a user to modify flights
- Allow admins to create financial reports
- Allow users to book multiple destination flights
- Allow admins to add new airports
- Make HTML more adaptable to different screen sizes