Enterprise Applications Assignment 2017

Team Alpha

Flight Booking System

Members:

Lindsay Borthwick

Christopher Conaghan

Karin Doohan

Corey Gallagher

Daire McDaid

Ford Rainey

# Table of Contents

[Table of Contents 2](#_Toc501055378)

[Table of Figures 2](#_Toc501055379)

[1. Introduction 3](#_Toc501055380)

[2. System Design 3](#_Toc501055381)

[2.1 Functional Requirements 3](#_Toc501055382)

[2.2 Non-Functional Requirements 4](#_Toc501055383)

[2.3 Activity Diagram 4](#_Toc501055384)

[2.4 Use Case Diagram 5](#_Toc501055385)

[2.5 Sequence Diagram 6](#_Toc501055386)

[2.6 State Activity Diagram 7](#_Toc501055387)

[3. Planning 8](#_Toc501055388)

[3.1 Planning Meeting 1 8](#_Toc501055389)

[3.2 Planning Meeting 2 9](#_Toc501055390)

[4. Customer Change Request 9](#_Toc501055391)

[5. Development Technologies 10](#_Toc501055392)

[5.1 Javadoc 10](#_Toc501055393)

[5.2 Junit 10](#_Toc501055394)

[5.3 JConsole 10](#_Toc501055395)

[6. Contract Amendments 10](#_Toc501055396)

# Table of Figures

[Figure 1 Customer Activity Diagram 5](#_Toc501024150)

[Figure 2 Use Case Diagram 6](#_Toc501024151)

[Figure 3 Sequence Diagram 7](#_Toc501024152)

[Figure 4 State Activity Diagram 8](#_Toc501024153)

[Figure 5 Scrum One 9](#_Toc501024154)

[Figure 6 Scrum Two 10](#_Toc501024155)

# Introduction

Team Alpha was assigned to complete an online Flight Booking system whereby a customer can enter details for flights and the system will calculate the cost based on certain criteria. The system is also used by an administrator to access detailed information and edit as appropriate. The learning outcomes of this assignment was to experience an overview of the work carried out by developers and to utilise the development technologies available to assist teams in the development lifecycle of a project.

# System Design

## 2.1 Functional Requirements

Client:

* **Client registration / creation:** The client will need to register as a user on the system. When registering the client will need to enter the requested personal details and choose a personal username and password. The details entered might include the following – Title, first name, last name, address, contact no, email address and possibly credit card details.
* **Login:** The system will request the user name and password. This will be validated before the client can continue. Once validated the client will be presented with the main menu.
* **Check flights:** Client will choose the origin city and the destination city on selected dates. Client will need to select single or return flights.
* **Select business or personal:** Once the client has selected the flights the system will ask the client whether they want to travel business class or personal class.
* Select amount of tickets??
* **Select flights:** Once the flights are selected the system will calculate and display the price.
* **Confirm booking:** If the client is satisfied with the flights and the price, they can confirm the booking.
* **View Flight Details:** The system will allow the client to view all the information about each booking.

Administrator:

* **Login:** The system will request the user name and password. This will be validated before the administrator can continue. Once validated the administrator will be presented with their main menu.
* **Add, delete, modify client accounts:** The system will have functionality for the administrator to add, delete or modify any client accounts or details.
* Maintain flight schedules??

## 2.2 Non-Functional Requirements

* Calculate the cost of flights
* Keep track of destinations
* Calculate frequent flier miles.

## Activity Diagram

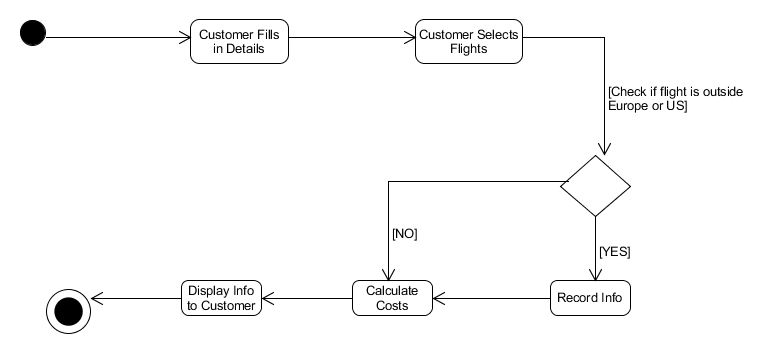


Figure 1. Customer Activity Diagram

The above diagram Fig. 1 demonstrates the actions that the customer will carry out when they are interacting with the system. Once the customer has registered and logged onto the system, he will enter the details for the flight that he wishes to take. The system will take note if the client has selected a flight that is outside either Europe or the USA. The system will record the information of the flight selected, i.e. destination and air miles to be recorded for each customer. The system will record the costs of the flights and display them to the customer.

## Use Case Diagram

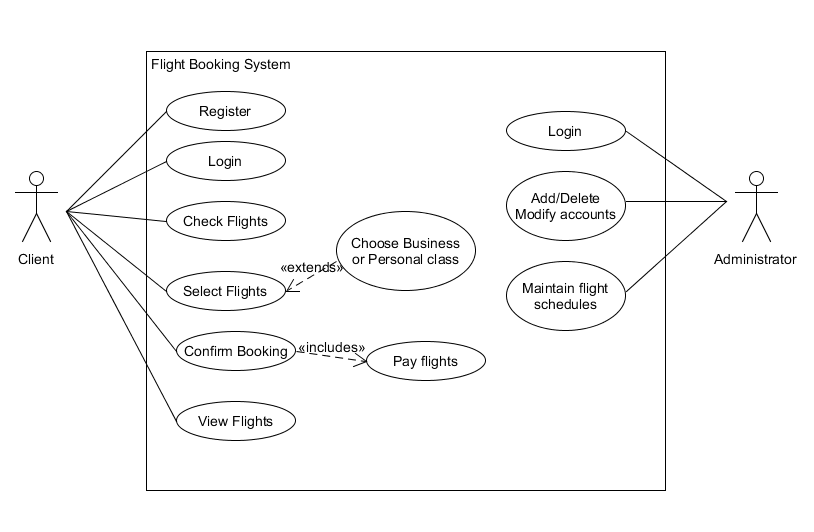


Figure 2. Use Case Diagram

The use case diagram displayed in figure 2 shows the functions of the client and the administrator.

The client has the ability to:

* Register to use the flight system
* Login to use the flight system
* The flights that are available
* Select the required flight, either business or personal class
* Confirm and pay for the flight
* View the confirmation of the booked flight

The administrator has the ability to:

* Log in to the flight system
* Modify the accounts by adding or deleting information
* Maintain the flight schedule system

## Sequence Diagram

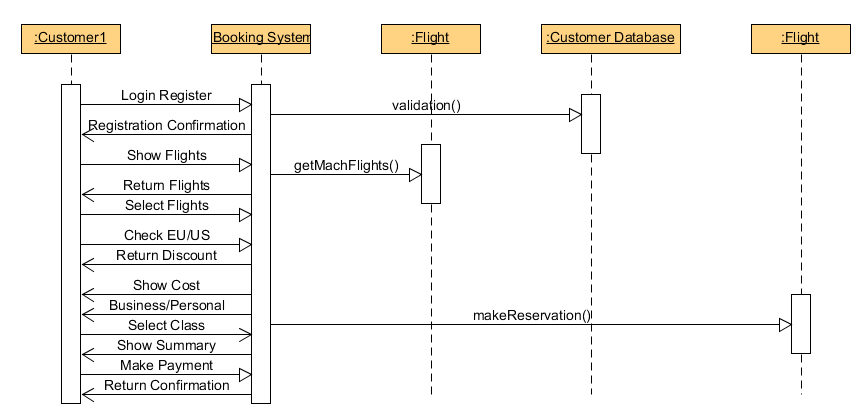


Figure 3. Sequence Diagram

## State Activity Diagram

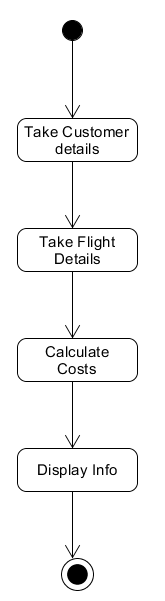


Figure 4. State Activity Diagram

# Planning

## 3.1 Planning Meeting 1

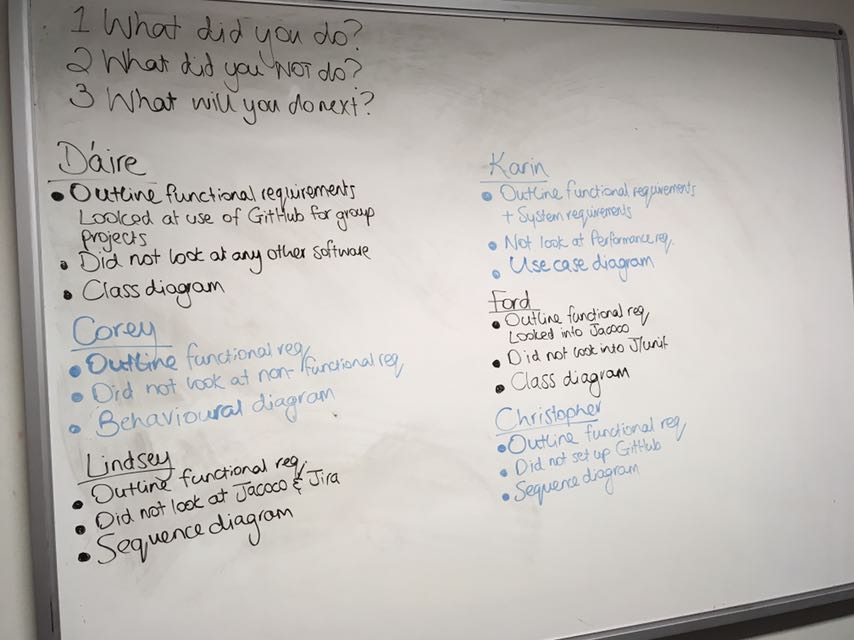


Figure 5. Scrum One

## 3.2 Planning Meeting 2

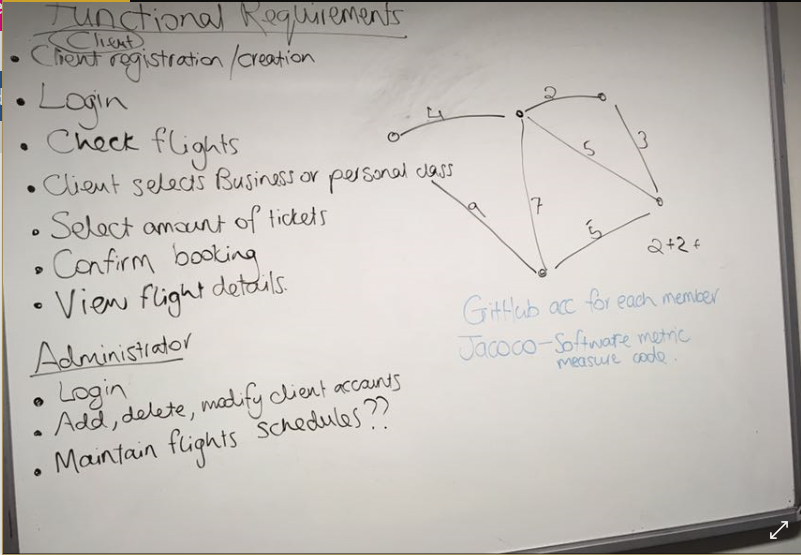


Figure 6 Scrum Two

# Customer Change Request

Customer has established that they would like a change to be implemented to their system, care was taking in the drafting up of a contract and both parties agreed on the change. The new implementation is a currency converter that works from the application so when your price is displayed at the end an option to view this price in different currencies is available.

Insert image of contract

# Development Technologies

The following table outlines the suggested tools to use throughout the project and which ones were utilised by the team.

|  |  |
| --- | --- |
| **Development Technologies** | **Used [Yes/No]** |
| Git / GitHub | YES |
| Javadoc | YES |
| Jacoco | NO |
| Junit/TestNG | YES |
| Jira/Agile Scrum | YES |
| JConsole/jRAT | YES |

## 5.1 Javadoc

Javadoc was generated for the application, outlining available class and methods used. Javadoc is a documentation generator used by most java applets which shows the API documentation in a HTML setup from the java source code.

## 5.2 Junit

Junit testing used to provide tests on the java app, results in increased programming speed and quality of code.

List Features

Sample Tests

Why

## 5.3 JConsole

A jmx compliant monitoring tool. Provides info on performance and resource consumption of apps.

# Contract Amendments

The original agreement that we had with the client has been amended to include the following information:

The client