# Introduction

During this group assignment a flight booking system will be produced. The system will be developed for the customer in the form of an on-line system where customers can enter their personal details to search for flights and return a price for the customer depending on destination and frequent flyer details. There will also be an option where the administrator can log in to the system where customer details can be viewed and edited.

This system will be developed by a team that will make use of some enterprise applications to aid in the development process. The applications are as follows: GitHub, Jira, and Javadocs. The system will be developed using agile methodologies and will comprise of several sprints of 1-2 weeks. The team will carry out scrum meetings weekly to communicate tasks that are completed, upcoming and how everyone is getting on with the development.

# Planning

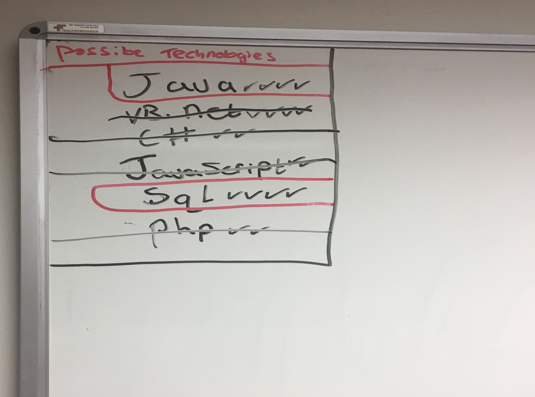
The team had meetings twice a week. One meeting was held in class and this is where the team had their weekly scrum. The second meeting was out of class and usually in a library study room. In these weekly meeting the team discussed the tasks completed, upcoming tasks, and any issues that were encountered during the past week. Jira was used to outline the tasks and to assign tasks to the team members.

## 2.1 Development Environment

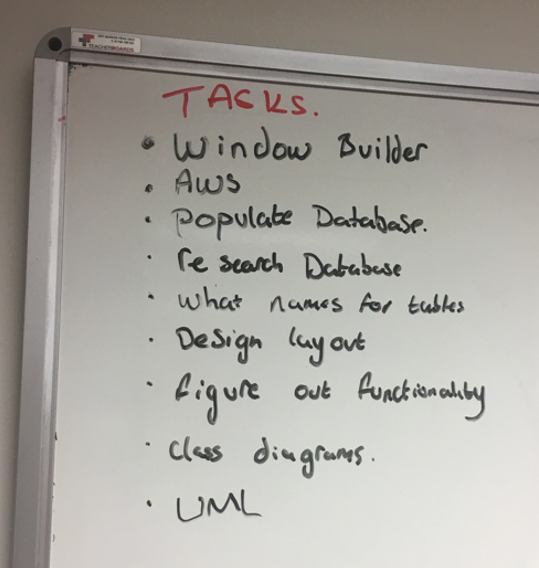
|  |  |
| --- | --- |
| Tools Used | Version |
| Eclipse | Oxygen |
| MySQL Database | AWS RDS |
| MySQL Workbench | 6.2 |
| Jira |  |
| XAMPP |  |
| GitHub |  |
|  |  |
|  |  |

# Design of the System

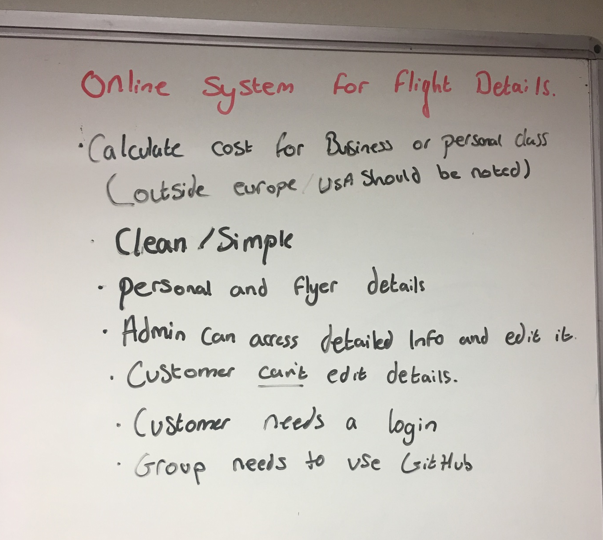
## Designing the system at team meetings



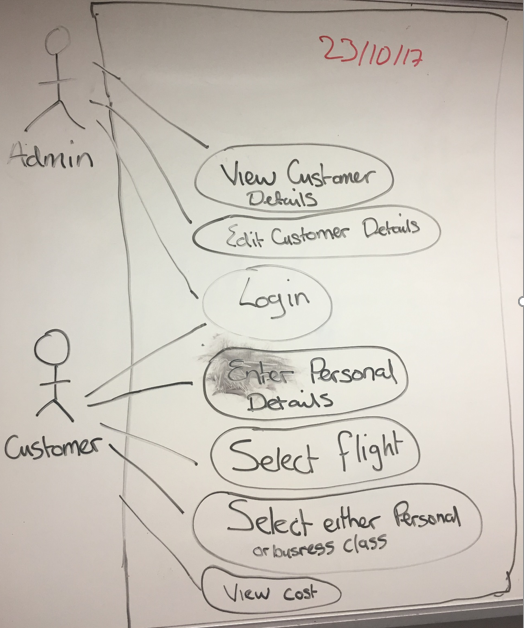
*Fig 3.1 – Possible Technologies*



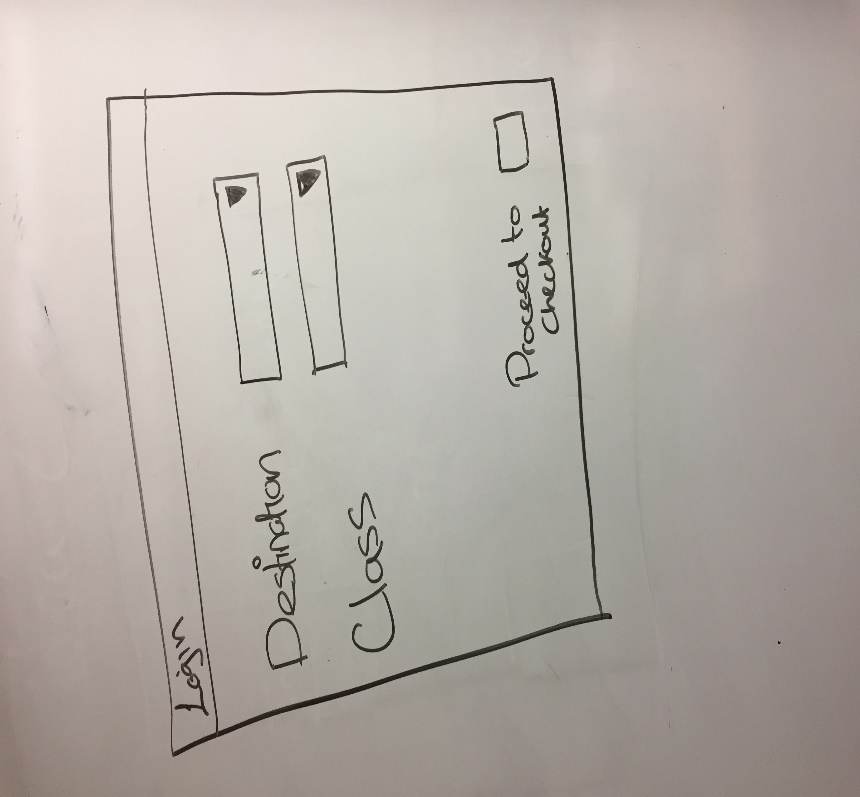
*Fig 3.2 – Tasks to complete the project*



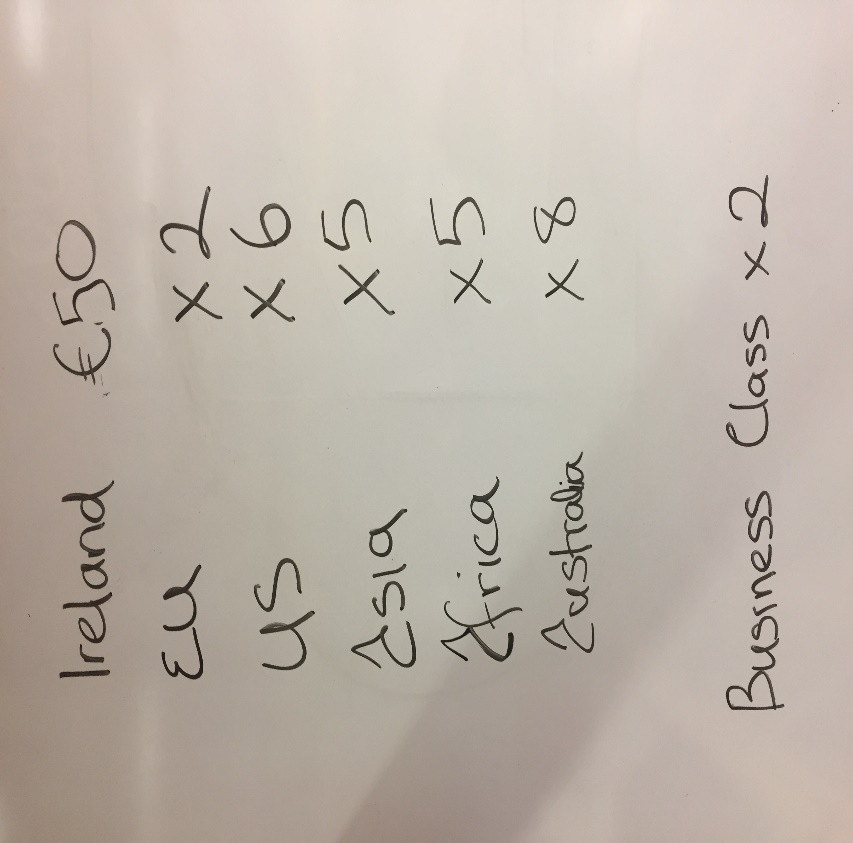
*Fig 3.3 – Group requirements*



*Fig 3.4 – UML Use Case Diagram*



*Fig 3.5 – Customer Login Window*



*Fig 3.6 – Costings for Travel Destinations*

## 3.2 Class Diagram

# 

# Use Case Descriptions

## 4.1 Register Client

|  |  |
| --- | --- |
| Use Case | Register Client |
| Objective | To register a client on the system |
| Precondition | The client wishes to register their details on the system. |
| Main Flow | 1. Client enters details. (Name, Address, DOB, Contact Details) 2. Client registers on system. |
| Alternative Flow |  |
| Post Condition | 1. Client has been registered. |

## 4.2 Select Flight

|  |  |
| --- | --- |
| Use Case | Select flight. |
| Objective | To select a flight. |
| Precondition | A client wishes to select a flight and view the cost. |
| Main Flow | 1. Client logs in. 2. Client views available flights. 3. Client selects flight. 4. Client selects business or personal class. 5. Client views cost. |
| Alternative Flow | 1. No log in details, see use case register client. |
| Post Condition | A client has selected a flight and viewed the cost of the flight. |

## 4.3 Edit Flight Details

|  |  |
| --- | --- |
| Use Case | Edit Flight Details |
| Objective | To edit flight details |
| Precondition | The administrator wishes to edit flight details on the system. |
| Main Flow | 1. Administrator logs in 2. Administrator edits flight details. |
| Alternative Flow |  |
| Post Condition | 1. Administrator has edited flight details. |

## 4.4 Edit Client Details

|  |  |
| --- | --- |
| Use Case | Edit Client Details |
| Objective | To edit client details on the system |
| Precondition | The administrator wishes to edit client details |
| Main Flow | 1. Administrator logs in. 2. Administrator edits client details |
| Alternative Flow |  |
| Post Condition | 1. Administrator has edited client details. |

# Requirements

The requirements for this flight booking system are as follows:

## 5.1 Functional Requirements

### Administrator

* Log in to System
* View Client Details
* Edit Client Details
* View Flight Details
* Edit Flight Details

### User

* Register
* Log In
* View Available Flights
* Select Flight
* Select Business or Personal Class
* View Cost of Flight

## 5.2 Non-Functional Requirements

### Availability

* The system should be 99.999% available.

### Security

* The system should only be available to registered users
* Passwords must contain numbers and letters and at least one capital letter.
* Users will have three attempts to login after which they will be locked out of the system.
* Once the client enters details it should not be able to be changed by the client.

### Confidentiality

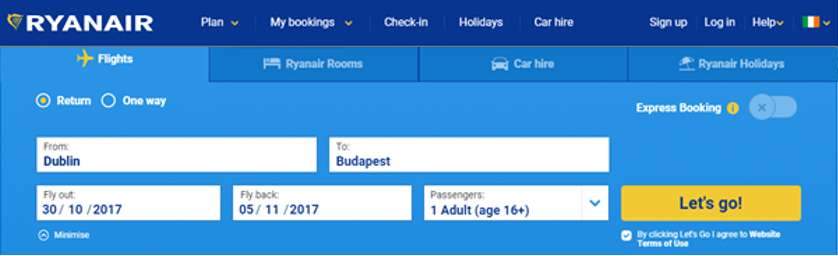
* Only the administrator may view the other client’s details.

### Usability

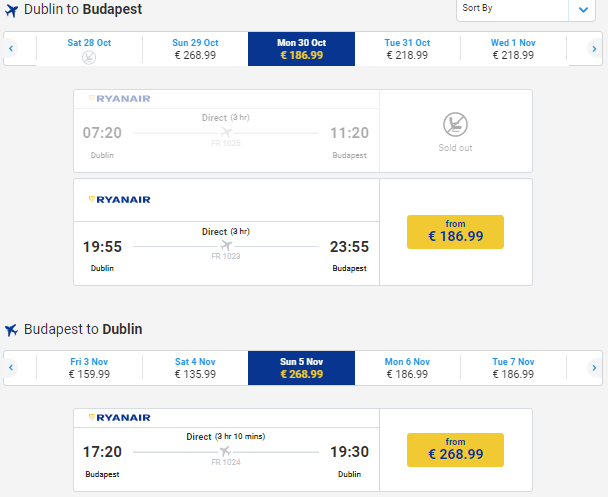
* The system shall be easy to use by people without training.

# Review of Existing Systems

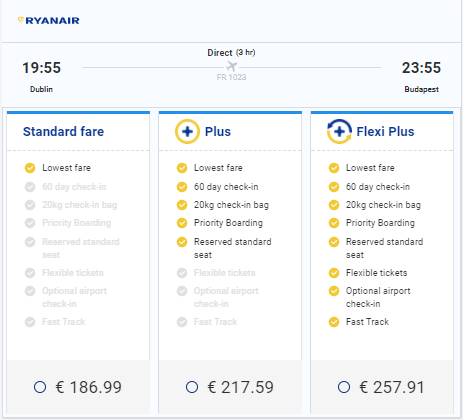
This section samples an existing system which serves a similar purpose to the system we have been assigned for our team project. It has been examined to so that we can create our own GUI that works in a similarly efficient way.



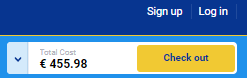
*Fig 6.1 - Ryanair’s Initial Menu*



*Fig 6.2 - Price for flight(s) displayed*



*Fig 6.3 - Additional options displayed*



*Fig 6.4 - Cost displayed with no user login required*

Our system will be developed in Java using eclipse. The GUI above can be similarly represented on a Java GUI platform as the Java components that exist are quite similar to this web based system’s components (e.g. JButton, JTextfield, JCheckBox, JDialog, JRadioButton etc.).

# DevOps Development Tools

## 7.1 Jira

Jira is a software development tool used for agile development. The team used this tool to create sprints that last one week at a time. Inside the sprints are a list of tasks that the team wish to have completed within the sprint. Tasks are assigned to team members and are then marked as done once they are completed.

# 

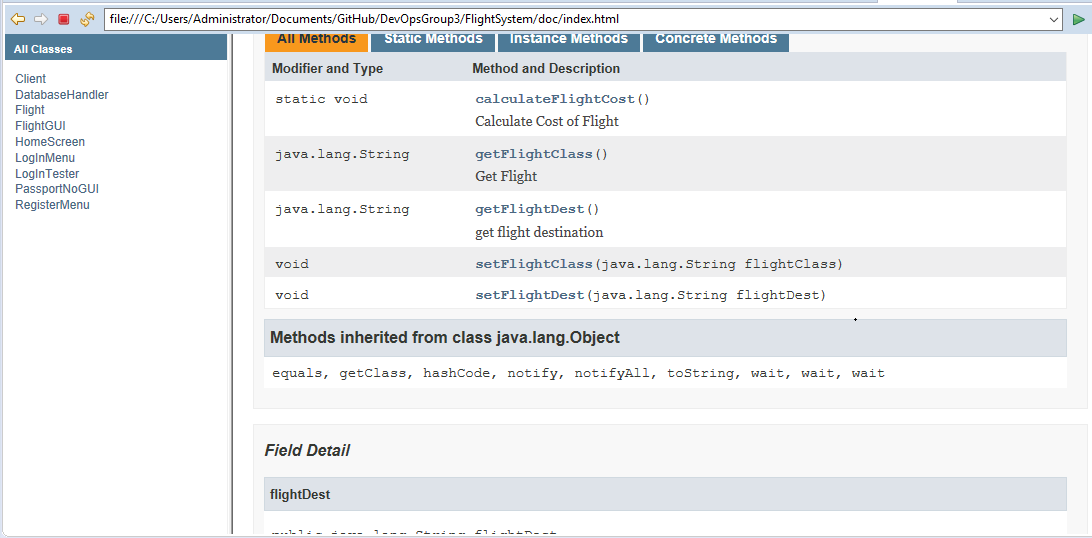
*Fig 7.7.1 Jira Backlog*

## 7.2 Javadoc

Javadoc is a tool which comes with the Java Development Kit which is used to generate documentation from Java source code in html format. Below are examples from our project of how the code is generated and the html format when generated.

# 

*Figure 7.2.1 Example of JavaDoc comments which generate Javadoc*



*Figure 7.2.2 Java Documentation after generation*

## GitHub

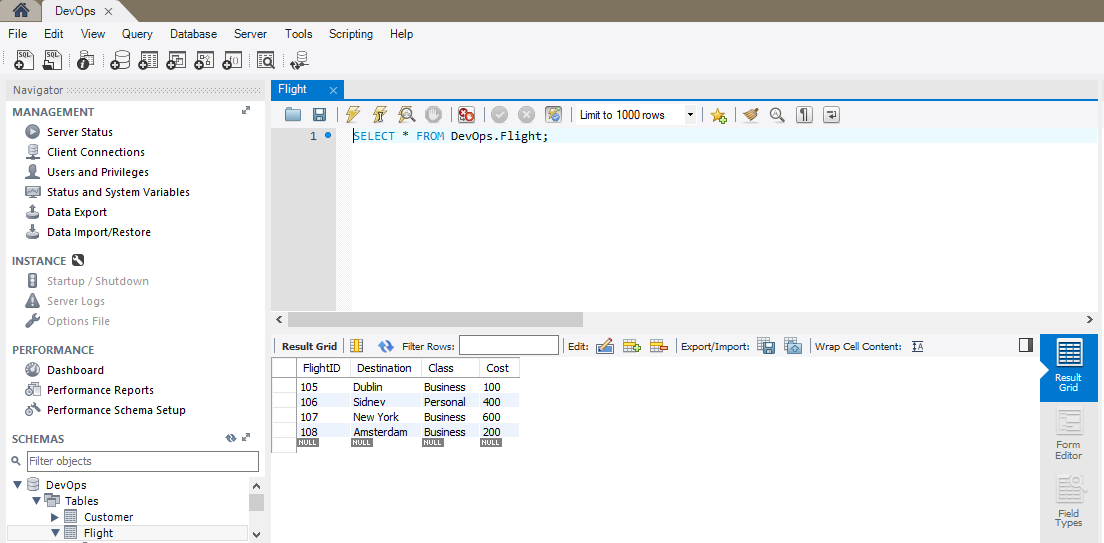
The team used GitHub for version control when building the system. Every time a change was made to the system code or the assignment document, it was uploaded onto GitHub.

## 

*Fig 7.3.1 Team GitHub Repository*

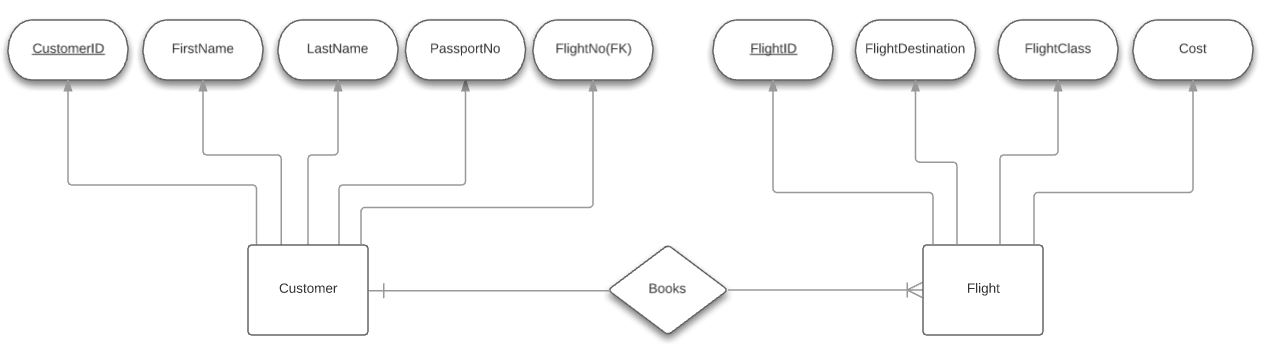
# Database

A MySQL database is used for this system. This database was stored on the AWS cloud.

**

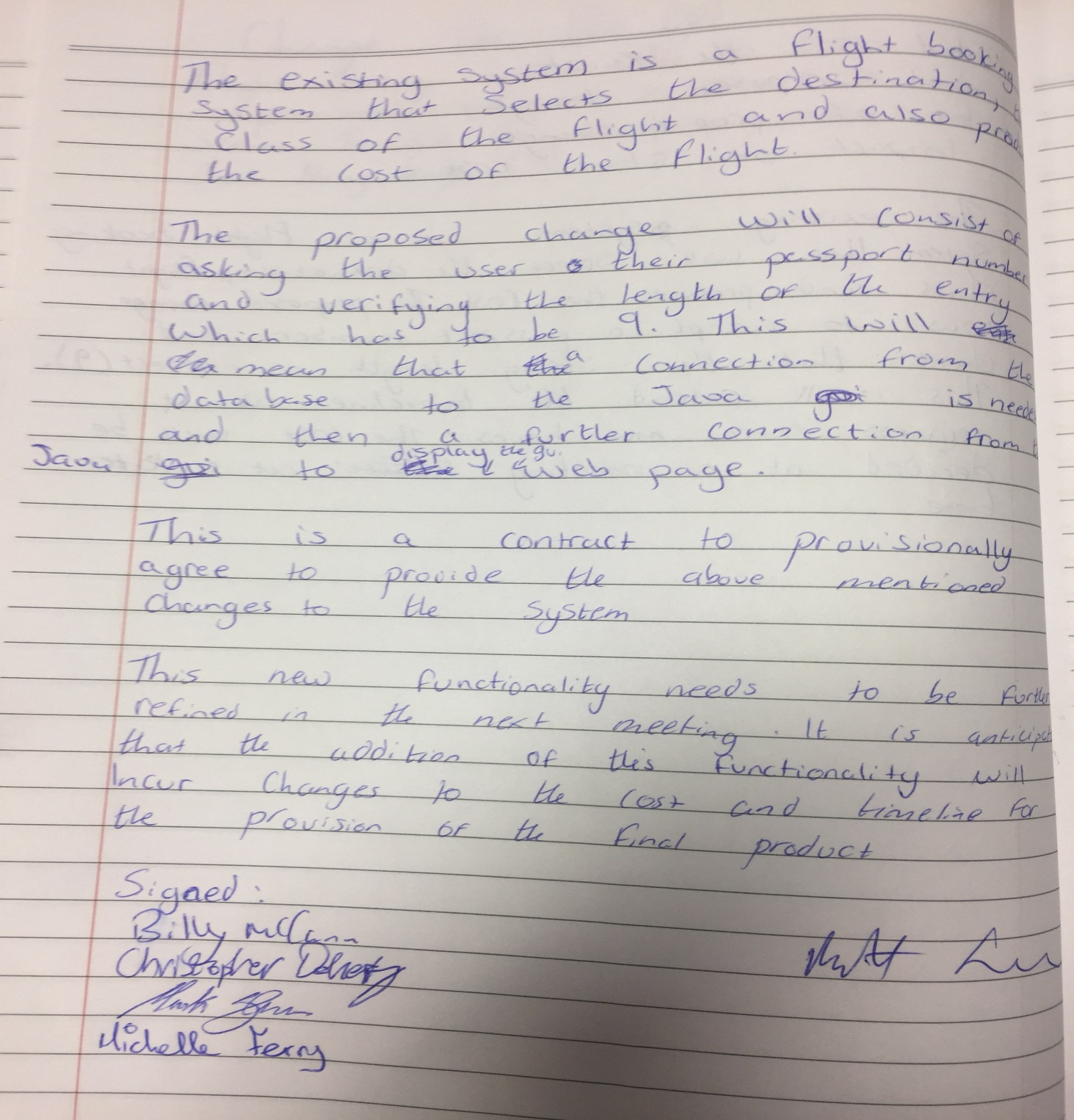
*Fig 8.1 MySQL Database – Flight Class*

## 8.1 Database ERD



*Figure 8.1.1 ERD for System*

# Contract Change



# Individual Conclusions

### Michelle Ferry

### Mark Glen

### Christopher Doherty

### Billy McCann

I found this group project very rewarding and also beneficial for later use in work life. We were asked to create a flight system that would allow the user to select a destination and also what class they would be flying in. Even though our group were one of the smallest in terms of members I felt that we had a great work ethic between the 4 of us and that we divided our work well and developed a great plan for the completion of our project. Personally, I felt that I could have added more functionality to my part of java. In the register page I could have added error handling to stop users entering number in the first and second name sections. Our group worked very well together. Each person expressed their ideas and none of them were metwith negativity.  There was a constant open line of communication via email, which we all utilized. On the day of our presentation, we all played a vital role. We all arrived early to set up.  During the weekly meetings every member took their turn to suggest how to move forward and we all decided what to do in the week that followed. Overall I felt our team worked very well and that our final project done what was asked of it.