Analysis, Design & Testing

SEG2012GP9

|  |  |
| --- | --- |
| Seabrook, Edward J F | [ejfs1g10@ecs.soton.ac.uk](mailto:ejfs1g10@ecs.soton.ac.uk) |
| Elliott, Kristian | [ke1g10@ ecs.soton.ac.uk](mailto:ke1g10@soton.ac.uk) |
| Yu, Brian | [by2g10@ ecs.soton.ac.uk](mailto:by2g10@soton.ac.uk) |
| Mariani Elola, Oscar N | [onme1g10@ ecs.soton.ac.uk](mailto:onme1g10@soton.ac.uk) |
| Chan, Yun Chak | [ycc1g11@ ecs.soton.ac.uk](mailto:ycc1g11@soton.ac.uk) |

# Notes to the Authors:

* Max 12 sides A4 in total for this report, excluding "wrapper" pages (e.g., title page, contents page, document control page) and appendices.

# 0 Wrapper Pages

## 0.1 Document Control

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Changes** |
| 1 | ejfs1g10 | Initial document |
| 2 | onme1g10 | Added conceptual model of HCI |
| 3 | ke1g10 | Added the approach taking |

## 0.2 Table of Contents

Notes to the Authors: 1

0 Wrapper Pages 2

0.1 Document Control 2

0.2 Table of Contents 3

1 Approach Taken 4

1.1 Technical 4

1.2 Modelling 4

2 Analysis Documentation 5

2.1 Requirements Analysis 5

2.2 Design Analysis 5

3 Testing 6

3.1 Test Plan 6

3.2 Partial Test Report 7

4 HCI 8

5 Screenshots 9

# 1 Approach Taken

## 1.1 Technical

The project specification has required us to work in the Java programming language; this works out well as we have all had experience working with Java and are familiar with it.

We have chosen to use Eclipse as our integrated development environment, this is because we have all had experience using it before and the majority voted for the use of Eclipse as their preferred IDE over anything else.

We will use Microsoft Word 2010 to produce the documentation for the project.

We will use Google Docs to circulate agendas before meetings.

We will use Forge for our project repository, each member of the group has used it before so we are familiar and it allows us to easily share the work we have made on our tasks.

We will use Microsoft Excel 2010 to produce a time expenditure report that will be updated by each individual member of the group, but will be monitored by Kelvin to check that everyone is keeping on tract with their allocated work.

We will use [www.smartsheet.com](http://www.smartsheet.com) to produce a Gantt chart and keep it updated; Edward found this piece of software to be good at producing Gantt Charts. It will allow us to monitor if we are at the right stage of the project and what tasks need to be completed next. Keeping it updated will allow us to have a middle Gantt chart and a final Gantt chart to show how the work breakdown changed over the course of the project.

The types of diagram we have chosen to include for our design of the software are, Use Case diagrams to identify all of the use cases from the requirements, class diagrams, so we have a way to break the software into smaller classes that can be coded and sequence diagrams to see how these classes are meant to interact within the system. There will also be wireframes so that we can design the UI of the system to be implemented in the later stages of development.

## 1.2 Modelling

In the project specification one of the requirements it to structure the code using the Model-View-Controller architecture, so our design process will start with the architecture and move on down to the individual components and build up from that.

# 2 Analysis Documentation

Analysis documentation: This section should include as appropriate class diagrams, statecharts and sequence diagrams, and explanatory narrative. This narrative is important - you need to make it easy for your examiner to see what's going on. Assume he knows nothing about the design of your system. The narrative should include a list of your significant class & method names and definitions.

**Proposal:**

## 2.1 Requirements Analysis

UML such as stakeholder diagram

## 2.2 Design Analysis

UML such as class diagram

# 3 Testing

## 3.1 Test Plan

Detailed test plan including numbered tests, test scripts, expected results. To include the demo UAT.

## 3.2 Partial Test Report

Partial test report including a good representative selection of test results, including UAT. We understand that it is difficult to complete your testing by this date but expect coverage of at least 75% of plan. Ideally, this and the test plan will be combined in tabular format.

# 4 HCI

***4.1 Conceptual Model.***

This software product will provide visual and graphical tools to recalculate and present revised runway parameters, visualizations and summarized calculations to the customer, given and obstacle located in it.

This team will take advantage of the widely known components of Java’s Graphical User Interface libraries and their high affordance and ease of use (e.g. drop down menus, menu bars, buttons, text areas, etc.) and allowing focus on those considered as the main usability goals.

While designing this system, the following usability goals will be regarded as most important:

* ‘Effective to use’; the aim is to build a set of tools that effectively ‘solve the problem’.
* ‘Safe to use’; the aim is a system with low error rate and easy recovery.
* ‘Good utility’; the system will count with a very appropriate set of functions to aid the customer when dealing with obstacles in a runway.

The system assumes a set of users familiarized with the concepts that they will be exposed to, including CAA’s (UK Civil Aviation Authority) rules and regulations, runway redeclaration, and runway parameters, including TORA (Take-off run available), ASDA (Accelerate Stop Distance Available), TODA (Take-Off Distance Available) and LDA (Landing Distance Available).

With the aid of this software product, rapid indication of the effects of an obstruction can be obtained and used to decide if operations in the runway can continue and if performing the manual calculations is worthwhile.

# 5 Screenshots

A small number of screenshots to enable your examiner to make the link between your demonstrated and your paper submissions (as an appendix).