# CMSC 345

### Software Design and Development

# Fall 2013

# System Requirements Specification

The Aviators: Smart Traveler

System Requirements Specification

#### Table of Contents

Page

## 1. Introduction 4

* 1. Purpose of This Document 4
  2. References 4
  3. Purpose of the Product 4
  4. Product Scope 4

2. Functional Requirements 4

3. Non-Functional Requirements 5

3.1 Customer Constraints 5

3.2 External Interfaces 6

3.3 Other 6

4. Deliverables 6

5. Open Issues 7

## Appendix A – Team percent contribution, Team sign off, Customer acceptance

## Document Versioning Control

|  |  |  |
| --- | --- | --- |
| **Version Number** | **Date** | **Changes from Previous Version** |
| 1.0 | 10/4/2013 | N/A |

## 1. Introduction

1.1 Purpose of This Document

The purpose of this document is to break down the requirements gathered from the customer, Professor John Squire, and organize them into an easily readable document intended for the customer and the developers. It will discuss the purpose of Smart Traveler and also the scope of the product. This document will also serve the purpose of defining an end product along with all of the deliverables.

*Note: This document is subject to change based on feedback from the customer.*

1.2 References

1.2.1 CMSC345\_3\_Requirements-Cain.pdf, by Russ Cain

1.2.2 srsTemplate.docx, by Russ Cain

1.2.3 *Basic Use Case Template***,** byAlistair Cockburn, http://members.aol.com/acockburn/papers/uctempla.htm, accessed 1/17/08

1.3 Purpose of the Product

The purpose of Smart Traveler is to provide the end user with a Graphical User Interface (GUI) that will allow the user to choose more than 20 different airports from a list of over 100. The user will be able to choose a start and stop airport and then Smart Traveler will display the shortest path between all chosen airports from start airport to stop airport. Smart Traveler will use the Google Earth display to show the user the shortest path, as well as information about each individual airport and also the total distance traveled. Smart Traveler will allow both users and administrators to log in at a start screen giving administrators the option of adding, editing, or removing airports from the system.

The situation for this product presented itself as being the semester project for CMSC 345, Software Engineering at the University of Maryland Baltimore County. The product will satisfy the need of a semester-long project and also give us, the developers, experience with a customer and all necessary documentation.

* 1. Product Scope



5.1

4.1

3.1

2.1

1.1

This diagram illustrates the top-level use case scenarios. The external actors on Smart Traveler are the User, Administrator, and Google Earth API. The User interacts with Smart Traveler by choosing the different airports the start and stop locations. The user can also interact with the system by accessing the help files provided. The Administrator shares this functionality and can also select any of the airports and modify it, create an airport to be added to the system, and remove an airport. The Google Earth API will be responsible for displaying the airports and information about the airports and also displaying the shortest route after calculation.

2. **Functional Requirements**

Each functional requirement should be represented using a use case.

Refer the reader once again to the top-level use case diagram from Section 1.4. In addition, include separate sub-use case diagrams, where appropriate, for each of the top-level use cases.

In addition to the diagrams, every use case should be documented using the use case specification format below. A suggested format for this section is to begin with a brief introduction of what the section contains, and then alternate sub-use case diagrams with their corresponding use case specifications. Make sure that all use case numbers and names correspond exactly with those in the top-level diagram of Section 1.4. Give all sub-use case diagrams figure numbers and labels (e.g., “Figure 2.1. Administer Exam”).

|  |  |  |
| --- | --- | --- |
| **Number** | *Unique use case number* | |
| Name | *Brief noun-verb phrase* | |
| **Summary** | *Brief summary of use case major actions* | |
| **Priority** | *1-5 (1 = lowest priority, 5 = highest priority)* | |
| **Preconditions** | *What needs to be true before use case “executes”* | |
| **Postconditions** | *What will be true after the use case successfully “executes”* | |
| **Primary Actor(s)** | *Primary actor name(s)* | |
| **Secondary Actor(s)** | *Secondary actor name(s)* | |
| **Trigger** | *The action that causes this use case to begin* | |
| **Main Scenario** | **Step** | **Action** |
|  | *Step #* | *This is the “main success scenario” or “happy path.”* |
|  | *Step #* | *Description of steps in successful use case* “execution” |
|  | *Step #* | *This should be in a “system-user-system, etc.” format.* |
| **Extensions** | **Step** | **Branching Action** |
|  | *Step #* | *Alternative paths that the use case may take* |
| **Open Issues** | *Issue #* | *Issues regarding the use case that need resolution* |

(This template was adapted from *Basic Use Case Template***,** byAlistair Cockburn, http://members.aol.com/acockburn/papers/uctempla.htm, accessed 1/17/08.)

3. **Non-Functional Requirements**

Decide on a standard format for the non-functional requirements (NFRs). Include a unique number for each NFR, a priority (1 = lowest, 5 = highest), and a clear, concise description. It is possible that your system has no NFRs. If this is the case, briefly state so.

3.1 Customer Constraints

These are the non-functional requirements that your customer has specified up front as being non-negotiable. For example, your customer may want the product developed as a desktop application to run on his/her office computer. In this case, the following non-functional requirements would be appropriate:

The system shall be a desktop application.

The system shall run on the customer’s office computer, which is a Dell Latitude D610 running under Microsoft Windows XP Professional, Version 2002, Service Pack 2. Or, it must be a web based application and must operate under the

Firefox browser. Or, it must be a smart phone app for brand XXX.

It is possible that your customer has specified no constraints. If this is the case, briefly state so.

3.2 External Interfaces

If your customer requires your product to read from data files that are external to the system (i.e., you do not have control over), the exact formats of these files (field descriptions, data types, range of possible values, and possible formats) must be specified. The same must be specified for any data files that must be written to that are external to the system. This situation usually arises when the customer keeps his/her own spreadsheet or database and would like your system to be able to share files with the spreadsheet or database. If there are no interfaces to external files or systems, briefly state so. For example, using MySQL, or Cloud access.

3.3 Other

Place the remaining non-functional requirements here. Do not include any NFRs that are related to the user interface. These will be included in the User Interface Design Document. Also, do not include NFRs that relate to hardware or software. Hardware and software specifications will be part of the System Design Document. As stated earlier, if your customer has already restricted you to particular hardware or software, put these NFRs in the Customer Constraints section above.

4. **Deliverables**

Hard copies of each of the following in class:

|  |  |  |
| --- | --- | --- |
| Deliverable | Due Date | Format |
| Systems Requirement Specification | 10/8/2013 | PDF |
| System Design Document | 10/15/2013 | PDF |
| User Interface Design Document | 10/22/2013 | PDF |
| Administrator Manual | 12/5/2013 | PDF |

Final customer deliverables in 3-ring binder:

|  |  |  |
| --- | --- | --- |
| Deliverable | Due Date | Format |
| “Read Me” document | 12/5/2013 | CD/PDF |
| Systems Requirement Specification | 12/5/2013 | CD/PDF |
| System Design Document | 12/5/2013 | CD/PDF |
| User Interface Design Document | 12/5/2013 | CD/PDF |
| Code Inspection Report | 12/5/2013 | CD/PDF |
| Test Report | 12/5/2013 | CD/PDF |
| Administrator Manual | 12/5/2013 | CD/PDF |
| Source Code | 12/5/2013 | HTML - CD/PDF |

5. **Open Issues**

List and briefly discuss issues that do not yet have a conclusion. Give specific target resolution dates. Be honest.

## Appendix A - Team percent contribution, Team sign off, Customer acceptance

**Sign off Agreement Between Customer and Contractor**

Describe what the customer and your team are agreeing to when all sign off on this document. [One paragraph] Include a statement that explains the procedure to be used in case there are future changes to the document. [One paragraph]

**Team Review Sign-off**

Provide a brief paragraph stating that all members of the team have reviewed the document and agree on its content and format. Provide lines for typed names, signatures, dates, and comments for each team member. The comment areas are to be used to state any minor points regarding the document that members may not agree with. Note that there cannot be any major points of contention.

**Document Contributions**

Remember that each team member must contribute to the writing (includes diagrams) for each document produced.