# CMSC 345

### Software Design and Development

# Fall 2013

# System Requirements Specification Development Team: “Team Twix” System Name: Tailored Travels Supervisor: Russ Cain Customer: Piyush Godbole

Team Twix’s Tailored Travels

System Requirements Specification

#### Table of Contents

Page

## 1. Introduction 4

* 1. Purpose of This Document
  2. References
  3. Purpose of the Product
  4. Product Scope 5

2. Functional Requirements 6

3. Non-Functional Requirements 7

3.1 Customer Constraints

3.2 External Interfaces

3.3 Other

4. Deliverables 8

5. Open Issues 9

## Appendix A 10

## Document Versioning Control

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

## 1. Introduction

* 1. Purpose of This Document

Herein are detailed the requirements associated with the geo-plotting software requested by the customer. All functional requirements, defined as those that refer to a technical performance of the software, are detailed as use-cases in *section 2*. Non-functional requirements, being the remaining non-technical requirements, are detailed under a priority list in *section 3*. Intended for general readership by those involved with the development of this project, this document will also list all intended deliverables to the customer in addition to an ongoing issues list.

***1.2*** References

Use Case Template: *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 software referred to by this document is being developed for the sole purpose of completing a class project at the University of Maryland, Baltimore County. The software shall be used to plot the shortest possible land route from the user’s chosen location to a destination selected from a stored list, while visiting every non-selected waypoint in that same list. This route is to be plotted on Google Earth as a series of straight lines connecting the various locations. All users of this software must log in to a personal account in order to access this functionality. If registered as an administrator, the user will be presented with controls to add, remove and edit the list of waypoints.

In our proposed solution to this task, Team Twix has so far outlined a Java application that will fulfill all later mentioned requirements. Using a graphical user interface, the application will give visual access to all controls and mechanisms of the app within the permissions of the current user. Upon selection and confirmation of the start and end waypoints, the user will be redirected to an instance of Google Earth. This instance will be summoned through the Java application after the computing of an ideal path has completed.

* 1. Product Scope

**This section identifies the boundary between the system to be developed and the outside world. That is, it identifies what will be included in the system and what will not. You will use a top-level use case diagram for this purpose. In addition to referring the reader to the diagram, give a brief description of how the diagram illustrates the system’s scope and its relationships to any external systems. Briefly describe the classes of users (the primary actors) and their relationships to the various system functionalities. Make sure to number the use cases in the diagram. Note: You do not need use cases for logging in or out of the system. [One to two substantial paragraphs]**

[Mariama – Diagrams]

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* |

|  |  |  |
| --- | --- | --- |
| **Number** | *1.1* | |
| Name | *Logs in* | |
| **Summary** | *User shall be able to log into the system* | |
| **Priority** | *5* | |
| **Preconditions** | *Username and password need to be entered* | |
| **Postconditions** | *The user has access to the system* | |
| **Primary Actor(s)** | *User* | |
| **Secondary Actor(s)** | *GUI* | |
| **Trigger** | *User submits login information* | |
| **Main Scenario** | **Step** | **Action** |
|  | *Step #1* | *User enters username* |
|  | *Step #2* | User enters password |
|  | *Step #3* | *User clicks button to login* |
| **Extensions** | **Step** | **Branching Action** |
|  | *Step #1* | *Step 1 and Step 2 can be performed in opposite order* |
| **Open Issues** | *Issue #1* | *Login information must be secure* |

|  |  |  |
| --- | --- | --- |
| **Number** | *1.2* | |
| Name | *Logs off* | |
| **Summary** | *The user shall be able to logout of the system when finished with application* | |
| **Priority** | *3* | |
| **Preconditions** | *User clicks logout option* | |
| **Postconditions** | *User is successfully outside of the system* | |
| **Primary Actor(s)** | *User* | |
| **Secondary Actor(s)** | *Application's GUI* | |
| **Trigger** | *User selects logout option from GUI* | |
| **Main Scenario** | **Step** | **Action** |
|  | *Step #1* | *User clicks logout option* |
|  | *Step #2* | User is no longer in the system |
| **Extensions** | **Step** | **Branching Action** |
|  | *Step #1* | *N/A* |
| **Open Issues** | *Issue #1* | *User's information is secure* |

|  |  |  |
| --- | --- | --- |
| **Number** | *2.1* | |
| Name | *Adds locations* | |
| **Summary** | *Administrators shall be able to add locations to the map* | |
| **Priority** | *5* | |
| **Preconditions** | *Users are prompted for locations (3.1), User must be admin* | |
| **Postconditions** | *The location is added to the map* | |
| **Primary Actor(s)** | *User* | |
| **Secondary Actor(s)** | *Map location* | |
| **Trigger** | *Administrator selects option to add a map location on GUI* | |
| **Main Scenario** | **Step** | **Action** |
|  | *Step #1* | *Admin selects location defined on map* |
|  | *Step #2* | User selects “add” option from GUI |
|  | *Step #3* | *Map location is successfully added to map* |
| **Extensions** | **Step** | **Branching Action** |
|  | *Step #* | *N/A* |
| **Open Issues** | *Issue #* | *Location must exist* |

|  |  |  |
| --- | --- | --- |
| **Number** | *2.2* | |
| Name | *Deletes locations* | |
| **Summary** | *Administrator shall be able to delete a location from the map* | |
| **Priority** | *2* | |
| **Preconditions** | *Location must be already added, user must be admin* | |
| **Postconditions** | *Location is successfully deleted from the map* | |
| **Primary Actor(s)** | *User* | |
| **Secondary Actor(s)** | *Map location* | |
| **Trigger** | *Admin selects “delete” option from GUI* | |
| **Main Scenario** | **Step** | **Action** |
|  | *Step #1* | *Admin selects a map location* |
|  | *Step #2* | Admin selects delete option |
|  | *Step #3* | *Location is deleted from map* |
| **Extensions** | **Step** | **Branching Action** |
|  | *Step #* | *N/A* |
| **Open Issues** | *Issue #* | *N/A* |

|  |  |  |
| --- | --- | --- |
| **Number** | *2.3* | |
| Name | *Edit location information* | |
| **Summary** | *Admins shall* | |
| **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* |

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

All user interactions with the system shall be performed through a graphical user interface (GUI)

It is expected from the customer that the system will present users with an intuitive GUI in order to minimize the technical knowledge needed to execute any external function of the software.

3.2 External Interfaces

The system shall communicate with Google Earth to display the calculated path.

It is required that upon computation of the shortest route between al waypoints that this route is drawn on an instance of Google Earth. This instance shall be summoned from within the delivered software. As such, KML output is required from the software in order to complete this communication.

3.3 Other

No other NFRs are currently evident.

4. **Deliverables**

Provide a list of all deliverable items (that is, all artifacts that you will deliver to the customer). This list will include items such as the product itself (What format? Source code? Executable code? Object code?), documentation, and training resources (if any). Specify when (date) and in what format (e.g., hard copy, CD) each will be delivered. You may assume that the deliverable items are as follows, although you may have more (e.g., training resources):

Hard copies of each of the following:

* Systems Requirement Specification
* System Design Document
* User Interface Design Document
* Administrator Manual

A CD and 3-ring binder (delivered at the product demo) containing the following:

* Documentation:
  + "Read Me First" document
  + System Requirements Specification
  + System Design Document
  + UI Design Document
  + Code Inspection Report
  + Test Report
  + Administrator Manual
* All source code
* The executable program (if applicable)
* Any other software required for program installation, etc.

Do not simply cut and paste this section into your document. Please come up with a more appropriate format. A tabular format works well.

**Things to hand in a three ring binder:**

* Systems requirement document: Document with what is needed for the program
* System Design Document: Document with how the program is designed
* User Interface Design Document: Document with how the user will interact with the program
* Administrator Manual: Document on how the administrator modifies the program in a reasonable manner
* Read me First document: Document that should be read before using the program
* Code Inspection Report: Document on how team members inspect each other’s code
* Test Report: Document on what test we did on the program
* Source code: Source code of the program if the customer wishes to inspect the code
* Executable: Finished product to allow the customer to run the program

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.

<https://github.com/smoore-ms9/UMBC-CMSC_345_TWIX/blob/master/Documents/Technical Reports/TWIX_SRS.docx>