EGR 401 – Capstone Design

Deliverable 3: Requirements Specification

Team Name: Blueprints

Team Members: Jordan Ziegler

  Jeremy Fischer

  Justus Karenzi

  Sudi Nsengiyumva

  Dylan Shanahan

Client’s Name: Dr. Mark Gordon

Advisor’s Name: Professor Thomas Renck

Version: 2

Date: 10/16/14

**3.1 Constraints**

Constraints involve the limits of variables that create the product. These values are only what is needed to accomplish what is specified by the client.

**3.1.1 Marketing Requirements**

These requirements are the requirements that are related to the feasibility of the project in a business standpoint.

**Efficiency**

The efficiency of the requirements specification is related to the battery life, memory storage, and processor use.

* Battery Life
  + The application should not excessively drain the battery of the phone/hardware in use. The battery of the hardware should not drain in excess of 1% per minute.
* Memory Storage
  + The application should not be excessive in allocation of memory on the phone/hardware. The application should not exceed a memory value of 256 MB.
* Processor Allocation
  + Processor allocation should not be excessive, but will probably be determined based on the choice in SDK.

**Time Constraints**

This project has a completion deadline of April 13, 2015.

**Budget Constraints**

The Team has been given a $500 budget with which to complete all the necessary components of the project.

**Software Constraints**

* Drop Down Menu
  + The user needs to be able to tell the program what they want through a drop down menu. The drop down menu needs to be easily accessible and understandable to the user.
* CSV File
  + The drop down menu needs to be able to be modified through csv files. The user ought to be able to use their own csv file to modify the names of the locations.
* Direction
  + The user needs to view the world through the phone/hardware’s camera, and display arrows or some other object to point the direction the user needs to go. The arrows should give the user course corrections so that they arrive at their destination in an expedient manner.
* Image Referencing
  + At the bare minimum the program needs to reference images found in the engineering building in order to develop a method for displaying the directions the user wishes to follow.
* Map Reference[[1]](#footnote-0)
  + The user needs to have the option of seeing the diagram in a top down format showing the directions rather than the real time view of the area they are in.
* Starting Location
  + The software needs to use a starting location in order to direct the student from one location to the next, or optionally develop a starting location based on the student’s current location in the building.
* Schedule Implementation[[2]](#footnote-1)
  + The program should optionally allow for the user to input their class schedule and use that to direct them to the room they need to go to, at the time they need to be there. The software might take into account the current time of day, day of the week, and where they are located in order to direct them to their class.

**Security Constraints**

**3.2 Standards**

A standard is defined as that which prescribes a concise set of conditions and requirements that must be satisfied by a material, product, process, procedure, convention, or test; and the physical, functional, performance and/or conformance characteristics thereof. Standards are used to ensure that the product meets a minimum performance efficiency, that it meets safety requirements, and that the process for production is consistent and can be repeated.

**Software Development Approach**

Our group plans on using the \_\_\_\_ approach.

**General Coding Practices**

(3.2.1) Coding Standards for Mobile Apps:

* Input Validation and Output Encoding
* Minimise lines of code.
* Use safe languages (e.g. from buffer-overflow).
* Implement a security report handling point (address) security@example.com
* Use static and binary code analyzers to find security flaws.
* Use safe string functions, avoid buffer and Integer overflow.
* Run with the minimum privilege required for the application on the operating system.
* Be aware of privileges granted by default by API's and disable them.
* Don't authorize code/app to execute with root/sa privilege.
* Always perform testing as a standard as well as a privileged user.
* Avoid opening application specific server sockets (listener ports) on the client device.
* Use the communication mechanisms provided by the OS.
* Context aware security: may be able to decrease/increase access based on the context (e.g. location, network).
* Remove all test code before releasing the application.
* Ensure logging is done appropriately but do not record excessive logs, especially including sensitive user information.
* What sort of information should be recorded in the logs. (Keep audit data on the server, no user specific data - link to the Apple Issue - Signed Timestamps).

**List of Standards**

(3.2.2) IEEE 12207 Systems and Software Engineering - Software Lifecycle process

* Addresses the complete software engineering lifecycle, from acquisition and supply, through development, to operations and maintenance.

(3.2.3) IEEE 1228 Standard for Software Safety Plans

* Establishes the minimum requirements for the content of a software plan. This applies to the software safety plan that is used for development, procurement, maintenance, and retirement of safety-critical software.

(3.2.4) IEEE 29119 Software and systems engineering - Software testing

* The purpose of this standard is to provide a guideline for software testing that can be used by an organization.

(3.2.5) Section 508 Standards for Electronic and Information Technology

* The standards are divided in four subparts: General, Technical Standards, Functional Performance Criteria, and Information, Documentation, and Support.
* 1194.21 Software applications and operating systems
  + When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.
* 1194.22 Web-based intranet and internet information and applications
  + Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.
* 1194.23 Telecommunications products
  + Telecommunications products which include voice communication functionality shall support all commonly used cross-manufacturer non-proprietary standard TTY signal protocols.

(3.2.6) Web Content Accessibility Guidelines (WCAG) 2.0

* Perceivable
  + Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
* Operable
  + Provide ways to help users navigate, find content, and determine where they are.
* Understandable
  + Make text content readable and understandable.
* Robust
  + Maximize compatibility with current and future user agents, including assistive technologies.

Security?

Programming Languages?

Data Formats?

**3.3 References**

A list of references for standards.

3.2.1 - Coding Practices:

* <https://github.com/NormanChiflen/Continuous-Integration--QA-Test--Deploy---Feedback/wiki/Coding-standards-for-mobile-apps>

3.2.2 - IEEE 12207

* <http://en.wikipedia.org/wiki/IEEE_12207>

3.2.3 - IEEE 1228

* <http://standards.ieee.org/findstds/standard/1228-1994.html>

3.2.4 - IEEE 29119

* <http://standards.ieee.org/findstds/standard/29119-2-2013.html>

3.2.5 - Section 508 Standards for Electronic and Information Technology

* <http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards>

3.2.6 - WCAG 2.0 Guidelines

* <http://www.w3.org/TR/WCAG20/>

**3.4 Requirements Specification**

Specifications are precise, unambiguous, measurable statements about what the product will do. They contain a metric and a value, and they specify behaviors, functions or attributes. They are the targets that the product must satisfy.

|  |  |  |
| --- | --- | --- |
| Marketing Requirements | Engineering Requirements | Justification |
| 3 | 1. The battery should not drain faster than 1% per minute | Despite relatively short periods of use, the user should not experience a quick depletion of their battery while using this application. |
| 3 | 2. The app should not take up more than 256MB of space on the hardware device. | The program will be designed to store most data on the hardware device, but will limit the space it takes up so that the user is not inconvenienced due to the large space demands of this program. |
| 3 | 3. The processor of the hardware device should not be overly taxed by the program. | This is tied in with the battery use and the desire not to inconvenience the user with the application use. The program should not be developed in inefficient manners which are unkind to the user’s hardware. |
| 2 | 4. The software should incorporate a drop down menu for room selection. | The user needs to be able to quickly and efficiently choose where they want to go. |
| 2 | 5. The software should allow for change to the drop down menu through a csv file | The ability to rename the rooms they frequent regularly is an important function for ease of use. It is convenient to be able to rename rooms based on what you remember them for, rather than what their actual titles are. |
| 1 | 6. The software needs to provide directions to the user using the “augmented reality” style as per the client’s request. | This is the fundamental of the program. The program needs to show the user where they need to go and how to get there. The easiest method for the user is to show them the world and which direction they need to go through the camera. |
| 1,2 | 7. The software needs to be able to recognize images. | The bare minimum software system needs to recognize and utilize images to determine location and arrow directions. |
| 1 | 8. The software should recognize where the user is starting from. | The program recognizes where the user starts out and from there is able to guide the user to their destination. |
| 1,2 | 9. The software will provide a map layout of the building.[[3]](#footnote-2) | The program needs to show the user a top down map of the area and the directions they will be taking. |
| 4 | 10. The software will be able to include the user’s schedule into the program and use it for directions.[[4]](#footnote-3) | The schedule can optionally be implemented for the program to determine where the user needs to go and when they need to go there to create better ease of use. |

Table 3.1

Marketing Requirements:

1. The program should show users where they need to go.
2. The program should be user friendly and easily operated.
3. The program should not strain the user’s hardware.
4. Optional functionality

Discussion:

The program’s engineering specifications are developed from the marketing requirements. They are attempts to fulfill the requirements using measurable values and justifiable, scientific reasons.

The first engineering requirement is to ensure that the program does not require too much energy for the execution of the programs functions.

The second and third engineering requirement is to ensure that the limitations of the user’s hardware is not maximized by the program.

The fifth engineering requirement is to allow the easy integration of additional files into the program and for the ability to change the values of those files.

The sixth engineering requirement is for aesthetics and ease of use.

The seventh engineering requirement is for the background code used to identify where the user is and display the directions for the user.

The eighth engineering requirement is an optional feature that would show a top view of a building and allow users to follow a specific path.

The ninth engineering requirement is to determine a user’s starting location.

The tenth engineering requirement is an optional feature to implement a user’s schedule to create better functionality.

1. Optional [↑](#footnote-ref-0)
2. Optional [↑](#footnote-ref-1)
3. Optional [↑](#footnote-ref-2)
4. Optional [↑](#footnote-ref-3)