

# **Software Requirements Specification**

**for**

## **Washington State University Vancouver Interactive Map**

**Version <2.0>**

**Prepared by**

**Group Name: Team 12**

**Daniel Brown  
Derek Cheung  
William Harer**

**11655489  
11660299  
11687920**

**daniel.j.brown@wsu.edu  
derek.cheung@wsu.edu  
william.harer@wsu.edu**

**Date: 10/25/2019**

# Contents

<b>CONTENTS.....</b>	<b>II</b>
<b>REVISIONS.....</b>	<b>III</b>
<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1 DOCUMENT PURPOSE.....	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	2
1.5 DOCUMENT CONVENTIONS.....	2
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	2
<b>2 OVERALL DESCRIPTION.....</b>	<b>3</b>
2.1 PRODUCT PERSPECTIVE.....	3
2.2 PRODUCT FUNCTIONALITY.....	4
2.3 USERS AND CHARACTERISTICS.....	4
2.4 OPERATING ENVIRONMENT.....	4
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	5
2.6 USER DOCUMENTATION.....	5
2.7 ASSUMPTIONS AND DEPENDENCIES.....	5
<b>3 SPECIFIC REQUIREMENTS.....</b>	<b>6</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	6
3.2 FUNCTIONAL REQUIREMENTS.....	7
3.3 BEHAVIOUR REQUIREMENTS.....	8
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS.....</b>	<b>10</b>
4.1 PERFORMANCE REQUIREMENTS.....	10
4.2 SAFETY AND SECURITY REQUIREMENTS.....	10
4.3 SOFTWARE QUALITY ATTRIBUTES.....	10
<b>5 OTHER REQUIREMENTS.....</b>	<b>12</b>
<b>APPENDIX A – DATA DICTIONARY.....</b>	<b>13</b>
<b>APPENDIX B - GROUP LOG.....</b>	<b>14</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1 Rough Draft	Daniel Brown Derek Cheung William Harer	Initial version. Largely blank.	10/20/2019
1.0 Rough Draft	Daniel Brown Derek Cheung William Harer	Almost all information was added.	10/24/2019
2.0 Final Draft	Daniel Brown Derek Cheung William Harer	Finished adding all necessary data. Edited content to be as accurate as possible and professional. Added general diagram, use-case diagram.	10/25/2019

# 1 Introduction

The Washington State University Vancouver Interactive Map is an online web application that displays a map of the campus that the user can interact with. The map provides various functions, such as directions from point to point, generating exercise routes, or highlighting recreation spots. This section is intended for a quick overview of what the software is, who the software is for, and what the software's broad functions are.

## 1.1 Document Purpose

The purpose of this document is to create a detailed outline of the Washington State University Vancouver Interactive Map and all of the requirements needed to create it. Through the use of this document, the software should be understood enough to not have to make any revisions throughout development. Any questions on the direction of development and future support should be answered through this document, as it covers the entire scope of the project as it is currently understood.

## 1.2 Product Scope

The Washington State University Vancouver Interactive Map is intended for students, faculty, staff, and visitors of Washington State University Vancouver campus. The goal of the software is to provide people with efficient routes for their traveling, quick options for discovering new walking, jogging, or running trails, as well as the ability to discover different things on campus, such as food, water fountains, and landmarks.

The benefits of this tool are many. This is a short, but not all-inclusive, list of benefits:

- It allows people who are unfamiliar with the campus to get directions on how to get where they need to be.
- It helps people who focus on fitness to create routes for exercise.
- If a person needs to find out where they can purchase food or drink, they can easily look it up to find out where they would like to go.

## 1.3 Intended Audience and Document Overview

The intended audience for this document is the developers behind the program, the professor, and perhaps future students who may have an interest in reproducing or altering this software later on down the line.

- **Developers:** the most relevant sections, aside from the general knowledge of what the program is about, are sections 3 and 4, *Specific Requirements* and *Non-Functional Requirements*.
- **Professor:** All of the sections are relevant, but for a general overview of who this product is for, see section 1, *Introduction*, and section 2, *Overall Description* to get a good idea of what this software is all about.
- **Students:** Section 2.6, *User Documentation*, should help give an overall introduction on how the product is used.

The rest of the SRS document contains specific details about the software, such as the software requirements, hardware requirements, limitations, and general use. The best way to read this document is simply from beginning to end, as it is organized to be in a comprehensive order.

## **1.4 Definitions, Acronyms and Abbreviations**

- SRS - Software Requirements Specification
- WSU - Washington State University
- WSUV - Washington State University Vancouver
- WSUV Map - Washington State University Vancouver Interactive Map

## **1.5 Document Conventions**

We will be using IEEE style standards for this document, of which there are a few conventions.

- Author names are listed first name and last, or first initial and last name.
- Any titles for journals or books are written in italics.
- The name of an article, chapter, patent, or other similar documents are in quotation marks.

If there are any citations for documents within this report, it will be listed in accordance to IEEE style standards. A helpful document has been provided to the authors of this report, and it will be the citation guide used throughout this document.

## **1.6 References and Acknowledgments**

### **1.6.1 References**

*There are no required references for this particular software.*

### **1.6.2 Acknowledgements**

Washington State University Vancouver campus maps  
<https://www.vancouver.wsu.edu/campus-map-directions>  
<https://www.vancouver.wsu.edu/about/cougar-trails>

## 2 Overall Description

### 2.1 Product Perspective

The WSUV Map is a self-contained web application that contains a map in conjunction with metadata for WSUV parking lots, building locations, trail routes, restrooms, and other locations of importance. Users will be able to interface with the WSUV Map to create personalized routes according to their schedule needs.

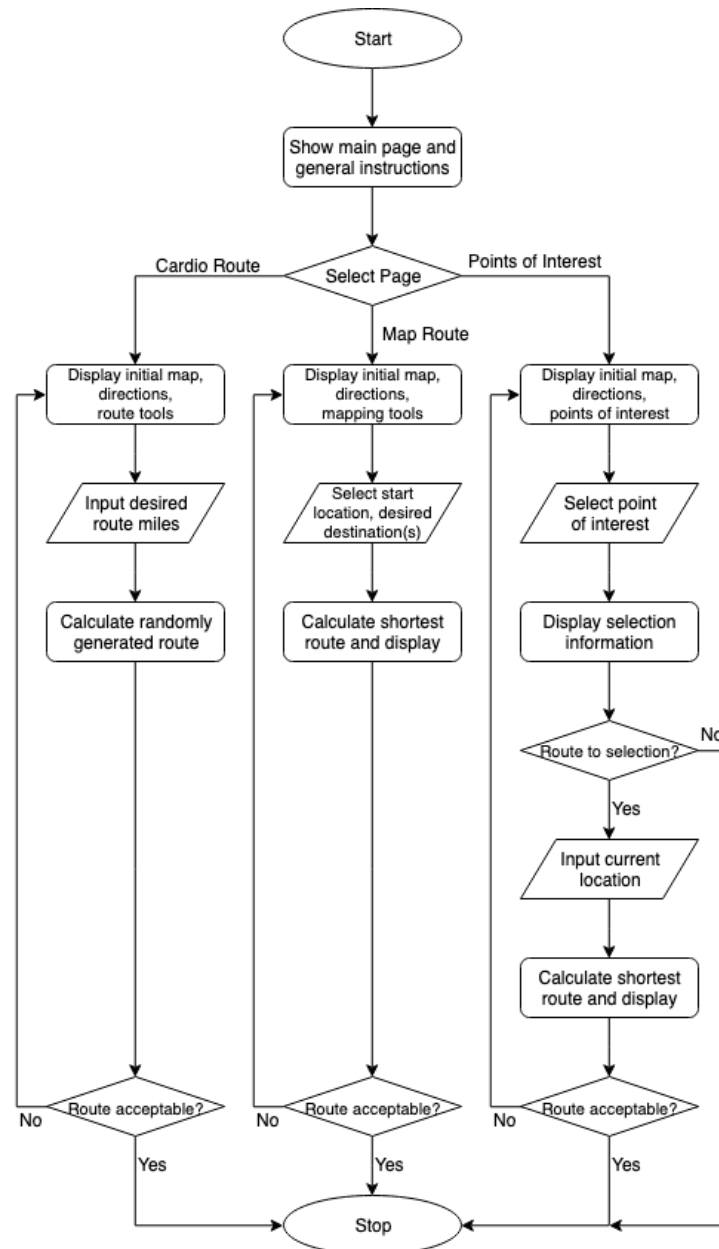


Figure 1. Flow chart of software.

## **2.2 Product Functionality**

- The system shall display a landing page containing information for application usage.
  - Additionally, the system shall display links for navigating between various pages.
- The system shall contain files containing important locations, linking their names with their location.
- The system shall have a page that displays a search bar for searching for key areas, such as buildings, food, restrooms, or landmarks.
- When the user inputs symbols into the search bar, the system should attempt to auto-complete the input with locations matching the given symbols.
- When the user selects a location, the system shall add that location to a list.
- Once a list is created and submitted, the system shall calculate a route that links the locations, in order, based on shortest distance/time.
- The system shall display a route to the user, drawn by lines on a map of WSUV.
- The system shall display a button for the user to print their calculated route.
- The system shall allow for the ability to create jogging trails and calculate the distance the route provides.
- For certain areas, such as a cafeteria, the system shall state what the normal operating hours of that location are.

## **2.3 Users and Characteristics**

### **Student and Faculty Users:**

We anticipate that WSUV students and faculty will be the most frequent users of this product. Both students and faculty are adherent to their individual weekly schedules, which require them to be in designated locations at designated times. The WSUV Map should assist students and faculty in efficiently maneuvering to and from classes and help them find specific locales if they need to deviate from their daily routes.

### **Local Pedestrians:**

This group includes people such as exercisers, dog-walkers, hikers, and other persons local to the WSUV area. We expect minimal to moderate product use from this group. Local pedestrians are less concerned with classroom locations; most product use will be centered on outdoor routes and other locations of interest (e.g., restrooms, vending machines).

### **Passersby:**

This group includes people who visit WSUV infrequently - event staff, extended family of students, and the like. We expect this group to utilize this product the least. Passersby may use the WSUV Map once or twice to determine a location of interest.

## **2.4 Operating Environment**

The WSUV Map shall be restricted to a web-based, JavaScript application. To operate consistently and efficiently, the user should have an updated, modern web browser installed on their device (e.g., Google Chrome, Firefox, Safari, Internet Explorer 3.0+). There exist no specific hardware requirements as long as a modern web browser is available; however, faster machines will better aid in route processing.

## 2.5 Design and Implementation Constraints

As per instructor request, this web application will be primarily JavaScript-based. We expect to limit route processing time to a maximum of five (5) seconds, leaning towards a shorter time with less client-side input. There should be no interaction with the WSUV website - all functions and usage will be localized to the web application. If, down the line in development, time allows to implement interfacing with WSUV's website to grab course specific information, that would be a useful feature, but not the goal of this software.

## 2.6 User Documentation

The landing page of the software will give a quick introduction to what the tool is, what the tool can do, and where to go for each use case. When the user goes to a separate page, there will be a quick summary of what the user is allowed to do on that web page, and how it will help them. The software itself will provide useful tips along the way, either in the form of labels, or by messages that pop up when the user hovers their mouse over the point of interest.

For users who are interested in developing their own versions of this software, there will be a detailed *README* file provided in the GitHub page of the software.

## 2.7 Assumptions and Dependencies

Assumed Factors:

- Users will not be creating huge routes. No more than five (5) classes in a single route will be considered, so the route calculations will cap out at seven (7) locations, to allow for transportation between parking lots and buildings.
- When creating an exercise route, the route should not exceed more than 5 miles in length.
  - Carryover length may be routed as X loops of Y length, where the carryover length is divided into Y mile segments X times.
- If an exercise route is created, it is assumed that the user does not care if it is not the exact distance specified.
  - Example: User requests a 5-mile route and is delivered a 4.6-mile route instead.
- The map routing system can be developed from scratch with the use of JavaScript, HTML, and CSS. If this becomes too much of a challenge, the development may have to include a third-party tool to assist in the mapping and creation of routes.

Dependencies:

- The WSUV Map shall be operationally constrained to a web application. There will be no database, so all data required will have to be contained in files and have a way to reliably read from those files.
- If the software ends up needing a third-party tool to reach completion, there could be constraints put on the project by the tool.



## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

The most important interface for the user will be a map of the school campus. This map will be the central hub of information for the user and will include inputs for starting and ending locations. The map will also output information such as the shortest path, amount of time to complete a given path, and options for alternative routes. Additional updates may also include a separate page that can take a student's schedule and outputs a custom map for the shortest path between each class.

The input that the user interacts with will be in the form of selecting a current location, and either one or many destination routes. This can be done by clicking on locations on the map, selecting from a list the starting and ending location, or searching for the locations with a text box. Other forms of input from the user will be in the form of inputting desired distances for creating a cardio route or interacting with a map to see different locations of interest around campus.

#### 3.1.2 Hardware Interfaces

This will be a website with no associated user data. Since we do not require user data, the only hardware interfaces will be between a computer (laptop/desktop) or a phone. For the website to be live it must also connect to a host server to display the content.

#### 3.1.3 Software Interfaces

The website must correctly display its interface and maps across multiple browsers (Chrome, Firefox, Safari, Edge, etc.). It must also interface either between an external library which controls map creation such as a GMaps (Google Maps) API, or we must write programs from scratch that would be able to create a graphical map, compute distances, and draw paths on the map. The user input also needs to interface with either the map libraries, or our own personal functions to correctly output data. The current goal is to get the map fully functional without any external libraries, but as stated in section 2.7, *Assumptions and Dependencies*, there is the chance that an external library will have to be used, which will change the requirements necessary to complete the software.

#### 3.1.4 Communications Interfaces

The encryption will only have to be strong enough to protect the three underlying structures of the software, which is HTML, JavaScript, and CSS, so the website will not need any encryption past the default HTTPS and will not store or interact with user data.

User input on the website only interacts with data locally stored on the server, such as building locations, points of interest, and the static web pages themselves. As soon as the user leaves the web page, the data they entered is lost. As such, if a user wishes to save a route, the page will have to be printed for future reference, or simply recreated by inputting the route a second time.

## **3.2 Functional Requirements**

### **3.2.1 Landing Page**

The main page for information regarding how to use the tool, with links leading to the other pages. The page also links to the GitHub page with a more technical readme.

### **3.2.2 Navigation Page**

The Navigation page gives a short description of what the page is for and how to use it. From here, users can select where they are and where they'd like to go, then calculate the shortest route.

#### **3.2.2.1 Selecting Locations**

The user can select their current location from this page, and up to 6 other locations to calculate a route. For the current and desired locations, the user can either type it, find it from a list, or select it on the map.

#### **3.2.2.2 Calculating Shortest Route**

After the user selects their desired locations, they can calculate the shortest route and have it displayed on the map. The map will be printer friendly so they can save the route for later.

### **3.2.3 Cardio Route Page**

The Cardio Route page will allow the user to calculate a route for their desired form of cardio exercise (e.g., running, jogging, or biking).

#### **3.2.3.1 Selecting Desired Distance**

The user can select the desired distance they want to travel. If it's higher than 5 miles, there is a large possibility the route will not loop perfectly. The user can also select if they want their route to be within the smaller campus map, or the larger Cougar trails map.

#### **3.2.3.2 Calculate Random Route**

Once the user has selected their desired distance and map size, they can calculate the route. There will be randomly generated values within the function that make the route different every time, so if the user doesn't like the route they received, they can calculate a new one.

### **3.2.4 Points of Interest Page**

The Points of Interest page will allow the user to search for and/or select a more specific location (e.g., restroom, vending machines, student services office).

#### **3.2.4.1 Displaying Information**

Once a location is searched for, the page will display the name and type of location alongside information relevant to the specific location. For example, if a bookstore is searched for, this page will note the name of the bookstore and its normal hours of operation.

### 3.2.4.2 Calculating Route

On the same page as the displayed information, there will be an option to create a route to the user's selection. Selecting this option will transfer the selection to the destination in the Navigation page (see section 3.2.2).

### 3.2.5 User Accessibility

The software is meant to be as accessible and easy-to-understand as possible, with shortcuts to functions like printing the current page so they can save their route for later. The whole software itself will be easily understandable to the average user and need very little instructing.

Whenever the user types into a search box for finding different locations on any of the three web pages, it will attempt to auto-complete what they're typing in, so the user can have a better understanding of whether or not that location exists before they've even finished typing.

### 3.2.6 Function Reusability

Many of the functions in the software will be reused for all main pages. The main calculation for the shortest distance can also be used for calculating the navigation for points of interest, as well as calculating routes for cardio trails.

## 3.3 Behaviour Requirements

### 3.3.1 Use Case View

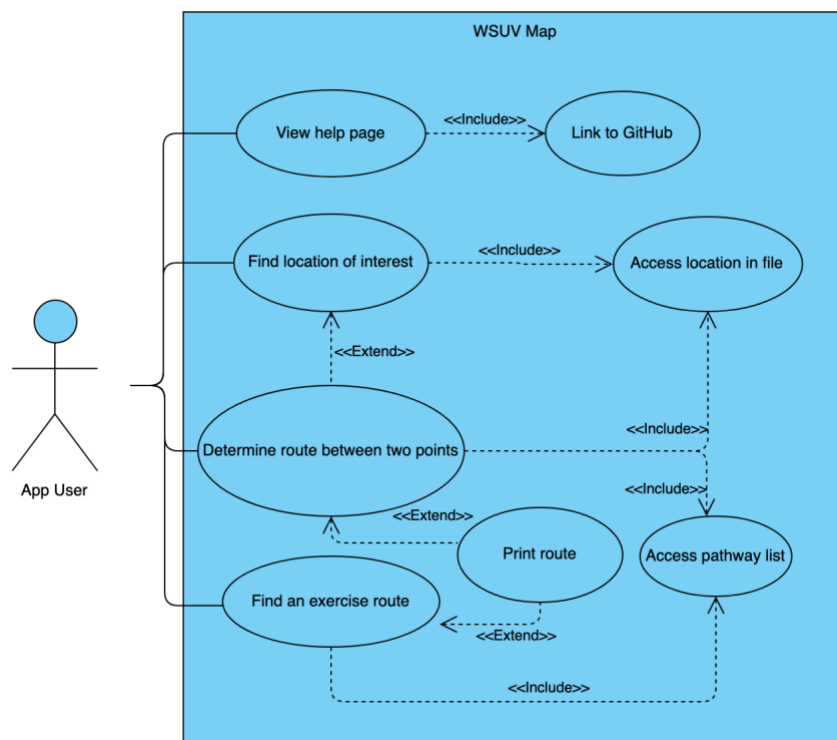


Figure 2. Use Case Diagram for WSUV Map.

### **3.3.2 Use Case Summary**

The actor in our use case is any user who needs to use the map to calculate directions from one location or another, create a cardio route, or locate various points of interest on WSUV's campus. This can be professors, students, visitors, or anyone who is on Washington State University Vancouver's campus and needs directions or ideas on what there is to do around campus.

There are three main pages to the software, aside from the landing page, that provide different functionalities for the user. The landing page encompasses all of the different options provided to the user in the use case.

## 4 Other Non-functional Requirements

### 4.1 Performance Requirements

This software will require a modern browser with HTML5 and JavaScript that can support the basic protocols necessary for the main functions of the software. There will be no specific functions that require cutting-edge hardware, but all of the route calculations will be done client-side, so a faster machine will perform better. Average hardware will be able to calculate routes in no more than 5 seconds once the current location and destination are set.

### 4.2 Safety and Security Requirements

The software is very straight-forward and simple. There are no user accounts, no data is stored server-side, and all calculations are done client-side. Based on this information, the expected list of security requirements are as follows:

- No users will have data collected, stored, or transferred.
- All potential data that could be stored is immediately lost as soon as the user leaves the web page.
- The software is operated under the principle of least privilege, and no special permissions are required to use it.

### 4.3 Software Quality Attributes

The following software quality attributes are not an all-inclusive list of what the software is, but it comes very close to explaining the different attributes necessary and what is expected of them.

#### 4.3.1 Usability

The software is intended to be as easy to use as possible. One of the demographics for the software are students who need to figure out where to go to get to their next class, so the main page needs to easily direct the person to where they should go to calculate what route they should take. The routing page itself needs to be simple and straightforward so they can quickly receive the directions to their next class.

As for the other pages, they should be straightforward and require little instructions on how to use. Certain features should exist to make it easier for people to refer back to, such as in the case of creating a jogging route, the page should allow the user to easily print that page for later use, since the website itself does not store any data and will lose that data as soon as they close the page.

#### 4.3.2 Portability

Since the software is processed through a browser, almost any device with modern HTML5 and JavaScript support can run the software without having to install anything. This allows someone who uses the software frequently to be able to access it from other devices seamlessly.

#### 4.3.3 Reliability

The software is only as reliable as the server it is hosted on. If the server goes down, the software will be unavailable. The goal is to host this software on a server that is reliable and secure, so the software will be available all the time.

#### **4.3.4 Discoverability**

Since the software is only geared towards those who come to the Washington State University Vancouver campus, the software should only be discoverable through their own website or by word-of-mouth. There is no plan to make the map widely known outside of the intended campus.

#### **4.3.5 Maintainability**

The software will be relatively lightweight, and the use of hosted web pages and JavaScript will allow for easy changes if requirements change. If, for instance, WSUV adds a new building to their campus, it would be relatively easy to update the map to reflect the changes.

## **5 Other Requirements**

- There will be a need for a server to host the WSUV Map on, since it is meant to be accessible across devices.
- If use of the WSUV campus map is restricted due to reserved use of intellectual property, a different method of route overlaying will need to be found.

## **Appendix A – Data Dictionary**

*This page intentionally left blank.*



## Appendix B - Group Log

10/24/2019:

- Daniel Brown
  - Drafted section 1, section 4.
- Derek Cheung
  - Drafted section 2, edited section 1.
- William Harer
  - Drafted section 3

10/25/2019:

- Daniel Brown
  - Edited and fleshed out sections 1-4.
- Derek Cheung
  - Created use-case diagram in section 3, added sections to 3.2.
- William Harer
  - Finished section 3.