# CMSC 345

### Software Design and Development

# Fall 2013

# **Testing Report**

Code To Joy:

Matthew Barteau

Benjamin Walsh

Sean Hutchinson

Amanda Hartman

John Pham

Code to Joy - Mapping Maestro

Testing Report

#### **Table of Contents**

Page

1. Introduction **3**

1.1 Purpose of This Document **3**

1.2 References **3**

1. Testing Process

2.1 Description **3**

2.2 Testing Sessions **4**

2.3 Impressions of the Process **4**

1. Test Results **4**

Appendix A – Team percent contributions. Team sign off, Customer acceptance

### 1. **Introduction**

* 1. Purpose of This Document

The purpose of this document is to demonstrate to the customer the types of testing that was used. We used a combination of unit testing, integration testing, system testing, and acceptance testing. We will explain why we didn’t use the ideal JUnit testing. We will cover the various ways we tested our program, and reasons why we did it that way. We will also note specific tests that we implemented and mention how well they went.

* 1. References

1. System Requirement Specification

2. Software Design Document

3. Code Inspection Report

2. **Testing Process**

* 1. Description

Our program had 5 major modular parts, shortest distance algorithm, google email sender, location file read/writer, kml file read/writer, and the help file helper. The shortest distance code was used under a GNU license, so we are free to use it for educational purposes. The help file helper and google email sender were snippets of generic example code that were examples for anybody online to use. These 3 pieces of code were lightly tested with just a main method within the file that allowed us to run it individually and verify that the side effect actually happened, or the results were as expected.

The location and kml file read/writers were completely written by us, and we tested them in a main method within their files by reading from some test files and comparing the output to what is actually in the file. We also tested it by writing some example data to a file, and verifying that the file was actually created, and opening the output file to verify the contents were the same.

In retrospect, we could have broken our program into more modular, testable parts. As we are all new to some of the concepts in this project, it took a bit of trial and error to figure out what actually worked and how, and what didn’t. It would have been difficult to follow a test driven development pattern, and most of our testing was after the fact to verify that it was working with our expected output. We weren’t too worried about bad input in these major modular parts since there was no direct user input being sent into those parts. We did not feel it was necessary to spend a lot of time testing for conditions that our code shouldn’t be able to do.

We did a fair amount of integration testing, system testing, and acceptance testing, where we would take our project at the final state and try out every option and path the user could take. We found a fair amount of bugs this way, and our project was small enough to fix the bugs pretty easily. Any bugs that we currently have are not anything major, and if we spent the time to thoroughly fix everything, it would be easy, but fairly tedious.

2.2 Testing Sessions

All tests were conducted with every team member present (except Xuanzhu Zhu) in the library. These were the dates and times that we tested:

-12/3 4:30-5:00pm

-12/5 3:45-4:30pm

-12/9 8:00-8:30pm

-12/10 3:30-4:00pm

-12/10 5:00-5:15pm

2.3 Impressions of the Process

Our testing process was maybe not ideal, but effective enough for what we were doing. Our project was small enough that we could fix any defects relatively quickly, and we didn’t need a lot of tests to help tell us where a problem was. We wouldn’t use our method for any large projects, but it is a quick and dirty way to finish a very small project.

Our best 2 modular units are probably the shortest path algorithm and the email sender, they were modules that we found online that we were free to use, and they were already tested by the authors, and we were able to verify that they work. Our worst two modular units were probably the location and kml file read/writers as far as bugs, because we didn’t get around to thoroughly testing them, but we did test them enough to say that it worked enough for our application.

3. **Test Results**

Our tests consisted of running the program and simulating some actions that the user could take. We simulated the following actions:

Admin controls:

- Adding a location

- Adding another location

- Removing a location

- Modifying a location

We would check that each of the add/remove/modify actions worked in different orders, and confirm that the expected results occurred.

Do all of the same tests that we would do as a regular user, as an admin.

When we tested adding/modifying a location, we would use boundary inputs

User controls:

- Create a user

- Login with the user

- Log out of the user and log back in

With add/remove/modify locations and creating a new user, we would verify in the actual user and location files that the changes occurred.

- Log in

- Select the minimum/maximum/variable number of locations, including recently added locations with random start/stop locations and let our program plot the locations on google earth.

We would repeat this procedure for each test that we did until we no longer had any bugs, the current defects are listed in the Code Inspection Report.

Appendix A – Team percent contributions. Team sign off, Customer acceptance

**Team Review Sign-off**

The customer and the contractors agree to the development of stand alone desktop application which shall find the shortest distance between given locations. It shall have a feature for future trips as per Vishnugopal Rajamanickam. If any future changes are made to the Testing Report document or if any additional features are proposed, we shall notify the developers and their customer via e-mail or in person of the conducted changes and proposals.

Amanda Hartman \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sean Hutchinson \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Matthew Barteau \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Xuanzhu Zhu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

John Pham \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Benjamin Walsh \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Customer:

Vishnugopal Rajamanickam \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Team Review Signoff**

I agree that I have read the above document and reviewed its content. If I do not agree with anything pertaining to the content or format of the document, I shall write it in the comments section below.

**Members of the Team: Signatures: Date:**

Amanda Hartman \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sean Hutchinson \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Matthew Barteau \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Xuanzhu Zhu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

John Pham \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Benjamin Walsh \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_