|  |
| --- |
| SFT221 Project |
| Test Plan |
| Truck Delivery Service |

|  |
| --- |
| Adrian Bautista, Brian Cheung, Jaaved Dewan, Marcus Brown  3-15-2023 |

Table of Contents

**Introduction1**

Purpose2

Project Overview3

**Scope4**

Type chapter title (level 2)5

Type chapter title (level 3)6

**Introduction**

**Purpose**

This test demonstrates the testing approach in a Truck Delivery Service and more specifically its pathfinding feature. This document will provide:

* Testing Strategy: assigning valid test cases with various tests at defined time frames.
* Execution Strategy: description of how the tests will be performed along with functional implementation.
* Test Management: handling of logistics, risk management and on-demand bug fixing.

**Project Overview**

The software being tested is on a pathfinding application of a Truck Delivery Service. The software will receive input from the user. This input includes the weight and size of a shipment box along with its destination. The program will calculate the best route based on three given coloured routes of blue yellow and green. All three routes start from the headquarters while the yellow and green will branch off the blue route.

The weight would be between 1 to 1000 kilograms while the size would be between 0.25 to 1 cubic meter. Furthermore, the destination is given in coordinates of at most 3 characters. The first two being numbers while the final character being an upper-case letter. Finally, to end the program, the user will input “0 0 x”. This input of setting weight and size to 0 while setting the destination to x will end the program.

**Scope**

The scope of this test will solely be based on the inputs received by the user and hence a lot of limitations and buffers are implemented in the program.

**Test Strategy**

The tests to be done would be:

* Unit Tests: Each function will be tested to make sure there aren’t any bugs or memory leaks.
* Manual Tests: Manual testing will be done through manual inputs.
* White Box Tests: Spaces will be tested in all inputs making sure there is a fail-safe message. This is especially the case for the destination input.
* Black Box Tests: General inputs to make sure there are proper edge cases and fail-safes for when input goes beyond the range.
* Integration Tests: Tests dependence on the integration of various modules and header files. This includes the mapping module along with the input module.
* Regression Tests: Tests will do at set interval times after changes of the code are made.
* Acceptance Tests: To make sure the software is business and consumer friendly and that it is ready to place in the market.

**Environment Requirements**

The testing will be done through the tester’s workstation with minimal requirements of hardware and software so long as the computer can have access to visual studio smoothly. Furthermore, the test can only be conducted after implementing the input module into the software. Finally, a test main module will be used.

**Execution Strategy**

The tests will be reported in a test matrix excel file. The file will classify any sort of bug and their severity. Please see excel sheet for further details.

**Test Schedule**

Unit, Manual, White box, and Black Box testing will be done at certain points of changes within the code. This is at least once a week. This is followed by Integration and Regression if any major changes are made. Finally, the acceptance test will be done at the end. The first few tests would take about 2-3 weeks while the following two will take another week. The final test would take a day.

**Control Procedures**

**Revision and Signoff Sheet**

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of Change |
|  |  |  |  |
|  |  |  |  |

Approver List

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Role | Approver | Approval Date |
|  |  |  |  |
|  |  |  |  |

Reference Documents

|  |  |  |
| --- | --- | --- |
| Version | Date | Document Name |
|  |  |  |
|  |  |  |

**Functions To Be Tested**

The functions to be tested include:

* printMap()
* addRoute()
* addPtToRoute()
* addPointToRoute()
* addPointtoRouteIfNot()
* distance()
* shortestPath()
* getPossibleMoves()
* getClosestPoint()
* Others that are added.

**Resources and Responsibilities**

**Resources/Tools:**

* Visual Studio Code
* Github
* Jira
* Seneca College
* C Program

**Responsibilities:**

* Data privacy
* Business acceptance
* Deadlines

**Deliverables**

The program fully functional.

**Suspension/Exit Criteria**

A 100% Functioning code.

**Resumption Criteria**

A newfound bug or error.

**Dependencies**

None.

**Risks**

* **Business**

Data loss or usability concerns cause customer complaints.  
Risk of privacy loss of client through destination.

Risk of product loss through wrong destination.

* **Technical**

Data loss might indicate bigger problems in the project.

* **Project**

This needs to be tested before the next phase of development.

Tight scheduling.