Greenway: System Verification and Validation Report

Team #11, Roadkill
Priyansh Shah, shahp36
Utsharga Rozario, rozariou
Jash Mehta, mehtaj8
Bilal Shaikh, shaikb2
Pranay Kotian, kotianp
Sharjil Mohsin, mohsis2

March 25, 2023

Contents

1 Revision History

Date	Version	Notes
March 9, 2023	1.0	Revision 0
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

3 Functional Requirements Evaluation

Tests taken from V and V Plan section 5.1 are evaluated here and the corresponding section names are used as well.

3.1 Start Screen

Start screen covers functional requirements 1-5 from the SRS as all those requirements are related to the inputs and functionality that exists in the program before any processing or output is shown

Preliminary Information Tests

1. preliminary-information-test-1

   Control: Automatic

   Initial State: No input in the start screen

   Input: Start Location

   Desired output: Status Message stating if the start Location was accepted

   Actual Output: No status message found

   Result: Test case failed as the app failed to display a popup of location being validated.

   Impact: The app will assume the use of valid start location as input as the core use of this app is to find the cost of trips not validate locations it is given.

2. preliminary-information-test-2

   Control: Automatic

   Initial State: No input in the start screen

   Input: Location of the Destination

   Output: Status Message stating if the Location was accepted

   Desired output: Status Message stating if the destination location was accepted

Actual Output: No status message found

Result: Test case failed as the app failed to display a popup of the location being validated.

Impact: The app will assume the use of valid end locations as input as the core use of this app is to find the cost of trips not validate locations it is given.

3. preliminary-information-test-3

   Control: Automatic

   Initial State: No input in the start screen

   Input: Car Details

   Desired output: Status Message stating if the Car Information was accepted

   Actual Output: No status message found

   Result: Test case failed as the app failed to display a popup of the status message validating car details.

   Impact: The app with its dropdown menus only allows the user to select valid cars and similarily only asks for mileage as alternative.

4. preliminary-information-test-4

   Control: Automatic

   Initial State: Accepted Car details in the start screen

   Input: Car mileage/fuel economy information

   Desired output: Status Message stating if the Car mileage/fuel economy information was accepted

   Actual Output: No status message found

   Result: Test case failed as the app failed to display a popup of the status message validating mileage information.

   Impact: Small popup known as toast will be added to the app to indicate a valid/invalid value has been added to the mileage.

5. preliminary-information-test-5

   Control: Automatic

   Initial State: Accepted Car details in the start screen

   Input: Car information

   Desired output: Status Message stating if the Car information was updated

   Actual Output: No status message found

   Result: Test case failed as the app failed to display a popup of the status message stating if the Car information was updated successfully.

   Impact: The popup does not need to exist as the app should require a full reset as a lot of different information needs to be updated to make this feature viable for the major goal of this app.

3.2   Map Interactions

Map Interaction covers functional requirements 6-8 as these requirements are concerned with functionality related to the map on display.

Map Tests

1. map-test-1

   Control: Manual

   Initial State: Start Screen finished

   Input: None

   Desired output: Map displaying a route from start to end based on start screen input

   Actual Output: Found full route displayed on the map to indicate the route the app considers most ideal

   Result: Test case passed as the desired output was found in the app.

   Impact: None

2. map-test-2

Control: Manual

Initial State: Start Screen finished

Input: None

Desired output: Map displaying all gas stations that it can possibly encounter along the route from a start to end destination

Actual Output: Found no gas stations along the route being displayed.

Result: Test case failed as the desired output was not found in the app.

Impact: The requirement that was used in the derivation process for this is not important anymore as the app functions on the basis that refueling is done at the start of the trip.

3. map-test-3

Control: Manual

Initial State: Start Screen finished

Input: None

Desired output: Map displaying all gas station prices as they come up along the displayed route

Actual Output: Found no gas stations as such along the route being displayed.

Result: Test case failed as the desired output was not found in the app.

Impact: Gas prices could not feasibly be found for all the gas stations as such the requirements used to derive this test case are not valid anymore.

3.3 Backend Processing

Backend processing covers functional requirements 9-12 as these requirements are concerned with calculations and, external data collection and processing alike.

Backend Tests

1. backend-test-1

   Control: Automatic

   Initial State: Start Screen finished

   Input: Route Details

   Desired output: Route returned to the user is one that is optimizing the distance and elevation perfectly to reduce fuel costs

   Actual Output: Route returned to the user is optimized to minimize fuel costs

   Result: Test case passed as the main goal of the optimization is achieved as minimal fuel costs are being shown.

   Impact: None.

2. backend-test-2

   Control: Automatic

   Initial State: Start Screen finished

   Input: Route Details

   Desired output: The database call on the backend returns correct elevation data for a route.

   Actual Output: The route details for multiple routes was matched with correct reference data in the framework and was found to be identical.

   Result: Test case passed as the desired output was achieved.

   Impact: None.

3. backend-test-3

   Control: Automatic

   Initial State: Start Screen finished

   Input: Route Details

Desired output: Correct mileage is given by backend for a certain coordinate.

Actual Output: Mileage that matched reference value is given by the backend.

Result: Test case passed as the desired output was achieved.

Impact: None.

4. backend-test-4

Control: Automatic

Initial State: Start Screen finished

Input: Route Details

Desired output: Correct total cost is calculated by the backend for a certain route with a start and end decisions as the route details.

Actual Output: All data required for correct total cost is given by the backend which is verified with reference data and equated in the front end.

Result: Test case passed as the desired output was achieved.

Impact: None.

4 Nonfunctional Requirements Evaluation

4.1 User Interface

Usability Tests

1. test-ui-1

Type: Functional, Manual

Initial State: A tester launches the Greenway app.

Input/Condition: A tester will observe the UI of the app to see if it has a modern, minimalistic look.

Desired Output/Result: After 10 testers use the application, there was an average score of 8 scored on Question 1 of the Usability Survey and 70% of testers answered 'Yes' on question 4 of the survey.

Actual Output/Result: There was an average score of 10 scored on question 1 of the survey, and 100% of testers answered 'Yes' on question 4 of the survey.

How test will be performed: This test will be verified using responses to Usability Survey Question 1 and 4. An average score of over 7 on Q1 and majority 'Yes' response to Q4 is considered a pass.

Result: Test case passed as the desired output was achieved.

Impact: None.

2. test-ui-2

Type: Functional, Manual

Initial State: A tester is viewing a page of the user interface.

Input/Condition: A tester clicks on any button that leads them to another page of the application.

Desired Output/Result: After 10 testers use the application, about 70% of testers answered 'Yes' on question 2 of the survey.

Actual Output/Result: 100% of testers answered 'Yes' on question 2 of the survey.

How test will be performed: This test will be verified using responses to Usability Survey Question 2. A majority 'Yes' response is a pass.

Result: Test case passed as the desired output was achieved.

Impact: None.

3. test-ui-3

Type: Functional, Manual

Initial State: A tester is viewing a page of the user interface.

Input/Condition: A tester clicks on any button that helps them navigate around the page.

Desired Output/Result: After 10 testers use the application, there was an average score of 8 scored on Question 3 of the Usability Survey

Actual Output/Result: The average score on question 3 of the survey was 9.

How test will be performed: This test will be verified using responses to Usability Survey Question 3. An average score of over a 7 is considered a pass.

Result: Test case passed as the desired output was achieved.

Impact: None.

4. test-ui-4

Type: Functional, Manual

Initial State: The tester is on the main page of the application.

Input/Condition: After the tester fills up all the field and presses the button that will calculate the route, they will check to see if any of the calculations are visible on the application itself.

Desired Output/Result: After 10 testers use the application, about 70% of testers answered 'No' on question 6 of the survey.

Actual Output/Result: 100% of testers answered 'No' on question 6 of the survey.

How test will be performed: This test will be verified using responses to Usability Survey Question 6. A majority 'No' response is a pass.

Result: Test case passed as the desired output was achieved.

Impact: None.

5. test-ui-5

Type: Functional, Manual

Initial State: The tester is on the main page of the application.

Input/Condition: The tester will try to use the application with colour blind glasses to see if the application is still useable.

Desired Output/Result: After 10 testers use the application, about 70% of testers answered 'Yes' on question 7 of the survey.

Actual Output/Result: First test: 40% of testers answered 'Yes' on question 7 of the survey. Revised test: 80% of testers answered 'Yes' on question 7 of the survey.

How test will be performed: This test will be verified using responses to Usability Survey Question 7. A majority 'Yes' response is a pass.

Result: First test: Test case failed as the desired output was not achieved. Revised test: Test case passed as the desired output was achieved.

Impact: First test: This failure shows that we can remove the requirement of use of application for people with visual impairment as the requirement is not met. Revised test: None.

4.2 User Experience

Quality of Life Tests

1. test-qol-1

   Type: Functional, Manual

   Initial State: The tester will be on the GitHub page of the Greenway app.

   Input/Condition: The tester will follow the installation process on the GitHub page to see how simple it is.

   Desired Output/Result: After 10 testers use the application, about 70% of testers answered 'Yes' on question 7 of the survey.

   Actual Output/Result: 50% of testers answered 'Yes' on question 7 of the survey.

   How test will be performed: This test will be verified using responses to Usability Survey Question 8. A majority 'Yes' response is a pass.

   Result: Test case failed as the desired output was not achieved.

   Impact: The installation process on the GitHub page needs to be simplified in order it easier for all potential users to understand the process.

4.3 Performance

Performance and Robustness Tests

1. test-pr-1

   Type: Functional, Manual

   Initial State: The tester is on the main page of the application.

   Input/Condition: The tester inputs the starting location and the ending location fields and press the route calculation button. They will then observe the total time it takes to complete the calculation.

   Desired Output/Result: After 10 testers use the application, the average time calculated between all testers should be less than 5 seconds.

   Actual Output/Result: First test: The average time calculated between all testers is 9.8 seconds. Revised test: The average time calculated between all testers is 4.9 seconds.

   How test will be performed: Starting and ending location are entered. Calculate route button is selected. Amount of time to complete calculation is measured.

   Result: First test: Test case failed as the desired output was not achieved. Revised test: Test case passed as the desired output was achieved.

   Impact: First test: This failure shows that the calculation time after the route calculation button is pressed needs to be optimized further in order for the requirement to be met. Revised test: None.

2. test-pr-2

   Type: Functional, Manual

   Initial State: The tester is on the main page of the application.

   Input/Condition: The tester will enter details into all the input fields on the UI and press the route calculation button. They will then check to see if the fuel calculation consumption is done as accurately as possible.

   Desired Output/Result: After 10 testers use the application, about 70% of testers answered 'Yes' on question 11 of the survey.

Actual Output/Result: 90% of testers answered 'Yes' on question 11 of the survey.

How test will be performed: This test will be verified using responses to Usability Survey Question 11. A majority 'Yes' response is a pass.

Result: Test case passed as the desired output was achieved.

Impact: None.

3. test-pr-3

   Type: Functional, Manual

   Initial State: The application is already running running.

   Input/Condition: The tester will simply observe to see if the application is available for use 24 hours per day, 365 days per year.

   Desired Output/Result: After 10 testers use the application, about 70% of testers answered 'Yes' on question 12 of the survey.

   Actual Output/Result: 100% of testers answered 'Yes' on question 12 of the survey.

   How test will be performed: This test will be verified using responses to Usability Survey Question 12. A majority 'No' response is a pass.

   Result: Test case passed as the desired output was achieved.

   Impact: None.

4. test-pr-4

   Type: Functional, Manual

   Initial State: The tester is on the main page of the application.

   Input/Condition: Invalid and edge case inputs for input fields are entered by the tester.

   Desired Output/Result: For 70% of testers, they're notified that they need to modify input (error handling).

   Actual Output/Result: Only for 50% of testers, they're notified to change their input due to invalid or unconventional inputs.

How test will be performed: User inputs invalid and edge cases inputs. System's response to these inputs shall be recorded.

Result: Test case failed as the desired output was not achieved.

Impact: The failure of this test shows that the error handling for inputs needs to be fixed in order to account for edge cases.

5. test-pr-5

Type: Functional, Automated

Initial State: A tester launches the Greenway app using a script to run several instances of the program.

Input/Condition: The tester will run the several instances to check if the system can handle an infinite amount of users.

Desired Output/Result: For the majority of testers, they will report back that the system works fine with a very large amount of users.

Actual Output/Result: All the testers have reported that the system works fine with a lot of users in the system.

How test will be performed: Script to run several instances of the program will stress test the system. System's response to a large number of users shall be recorded.

Result: Test case passed as the desired output was achieved.

Impact: None.

6. test-pr-6

Type: Functional, Manual

Initial State: The application is already running running.

Input/Condition: The tester will simply observe to see if the system will operate as long as possible.

Desired Output/Result: After 10 testers use the application, about 70% of testers answered 'No' on question 13 of the survey.

Actual Output/Result: 90% of testers answered 'No' on question 13 of the survey.

How test will be performed: This test will be verified using responses to Usability Survey Question 13. A majority 'No' response is a pass.

Result: Test case passed as the desired output was achieved.

Impact: None.

4.4 Technology

Technological Tests

1. test-t-1

   Type: Functional, Manual

   Initial State: N/A

   Input/Condition: N/A

   Desired Output/Result: The application runs on all on browsers (Google Chrome, Firefox, Safari, Microsoft Edge)

   Actual Output/Result: The application ran on all on browsers (Google Chrome, Firefox, Safari, Microsoft Edge)

   Test Case Derivation: As NFR15 states, the system shall be able to run on all modern browsers.

   Result: Test case passed as desired output was achieved.

   Impact: None

2. test-t-2

   Type: Functional, Manual

   Initial State: N/A

   Input/Condition: N/A

   Desired Output/Result: The users are able to download the Github project.

   Actual Output/Result: The users were able to download the Github project.

How test will be performed: This test will be verified using responses to Usability Survey Question 9. A majority 'Yes' response is a pass.

Result: Test case passed as desired output was achieved.

Impact: None

3. test-t-3

Type: Functional, Manual

Initial State: N/A

Input/Condition: N/A

Desired Output/Result: The application runs from Windows, Mac, and Linux-based (Ubuntu) operating systems.

Actual Output/Result: The application runs from Windows, Mac, and Linux-based (Ubuntu) operating systems.

How test will be performed: This test will be verified using responses to Usability Survey Question 10. A majority 'Yes' response is a pass.

Result: Test case passed as desired output was achieved.

Impact: None

4.5   Maintainability

Maintainability Tests

1. test-m-1

Type: Functional, Manual

Initial State: N/A

Input/Condition: N/A

Desired Output/Result: The users found the necessary information about the application well documented.

Actual Output/Result: 8/10 users found the necessary information about the application well documented.

How test will be performed: Code inspection will be used to validate that all source code changes are documented and maintainable.

Result: Test case passed as desired output was achieved.

Impact: None

2. test-m-2

Type: Functional, Manual

Initial State: N/A

Input/Condition: N/A

Desired Output/Result: The users found the user's guide on how to use the application on GitHub.

Actual Output/Result: All the users found the user's guide on how to use the application on GitHub.

How test will be performed: This test will be verified using responses to Usability Survey Question 14. A majority 'Yes' response is a pass.

Result: Test case passed as desired output was achieved.

Impact: None

4.6   Legal

Legal Tests

1. test-l-1

Type: Functional, Manual

Initial State: N/A

Input/Condition: N/A

Desired Output/Result: The system complies with the Google Maps Platform Terms of Service.

Actual Output/Result: The system complies with the Google Maps Platform Terms of Service.

Test Case Derivation: As NFR21 states, the system should comply with the Google Maps Platform Terms of Service.

How test will be performed: Use Google Maps API to validate that our system is compliant with Google TOS.

Result: Test case passed as desired output was achieved.

Impact: None

5   Testing Results

Requirement #	Test ID(s)	Test Results
FR1	preliminary-information-test-1	Failed
FR2	preliminary-information-test-2	Failed
FR3	preliminary-information-test-3	Failed
FR4	preliminary-information-test-4	Failed
FR5	preliminary-information-test-5	Failed
FR6	map-test-1	Passed
FR7	map-test-2	Failed
FR8	map-test-3	Failed
FR9	backend-test-1	Passed
FR10	backend-test-2	Passed
FR11	backend-test-3	Passed
FR12	backend-test-4	Passed

Requirement #	Test ID(s)	Test Results
NFR1	test-ui-1	Passed
NFR2	test-ui-2	Passed
NFR3	test-ui-3	Passed
NFR6	test-ui-4	Passed
NFR7	test-ui-5	Failed
NFR16	test-qol-1	Failed
NFR8	test-pr-1	Failed
NFR9	test-pr-2	Passed
NFR10	test-pr-3	Passed
NFR11	test-pr-4	Failed
NFR13	test-pr-5	Passed
NFR14	test-pr-6	Passed
NFR15	test-t-1	Passed
NFR17	test-t-2	Passed
NFR20	test-t-3	Passed
NFR18	test-m-1	Passed
NFR19	test-m-2	Passed
NFR21	test-l-1	Passed

6 Changes Due to Testing

6.1 Preliminary Information Tests

To start with there will be some changes incoming to the SRS firstly FR-1 and FR-2 will be removed or modified due to preliminary-information-test-1 and 2. And FR-3 will be modified due to the results of test preliminary-information-test-3. The test preliminary-information-test-4 showed there is a need for small popup with the use of a toast ui component in the app and this will be added to the application. The test preliminary-information-test-5 showed that the SRS needs to be updated with removal or modification of FR-5 due to the changed priorities of the application.

6.2 Map Interactions

The test Mapmap-test-2 showed that the SRS needs to be changed for FR-7 as the refueling functionality has been modified from original concept of the application so this requirement will be changed or removed entirely. Similar idea for map-test-3 as the refueling functionality of the original concept has changed there is no need to show individual fuel prices in a certain area with the need of it being removed/changed.

6.3 Backend Processing

No changes to the backend from the testing.

6.4 Nonfunctional Requirements

From test-ui-5 it can be seen due to the change in goals for the app accessibility for the visually impaired is infeasible so NFR-7 will be changed or removed. The test test-qol-1 it can be seen that instructions need to be added to the applications for making it simpler to install. The test test-pr-1 failed as the application we have is un optimized for performance optimization and as such the app will focus on reducing processing time and interactions with database to reduce calculation time. From the result of test-pr-4 it is that the app is unable to correctly respond to a lot of invalid inputs and since the focus of our app has changed to assuming correct inputs the SRS will be updated to modify or remove NFR11.

7   Trace to Requirements

Requirement #	Description	Test ID(s)
FR1	The system shall allow the user a way to input a start location.	preliminary-information-test-1
FR2	The system shall allow the user a way to input a final destination.	preliminary-information-test-2
FR3	The system shall allow the user a way to select any car model and make.	preliminary-information-test-3
FR4	The system shall allow the user a way to input the fuel economy of their car.	preliminary-information-test-4
FR5	The system shall have a way to update available car models from either user input or external sources.	preliminary-information-test-5
FR6	The system shall have a map to show journey from start to final destination.	map-test-1
FR7	The system shall display the gas stations along the route.	map-test-2
FR8	The system shall display gas prices at stations on route to destination.	map-test-3
FR9	The system shall have a way to calculate the most fuel efficient route.	backend-test-1
FR10	The system shall have a way to collect elevation data along the suggested route.	backend-test-2
FR11	The system shall have a way to calculate fuel consumption on different terrain elevations.	backend-test-3
FR12	The system must calculate the total cost of the journey.	backend-test-4

Requirement #	Description	Test ID(s)
NFR1	The system shall have a modern minimalist user interface.	test-ui-1
NFR2	The system shall have a consistent style across the different pages (or interfaces) of the platform.	test-ui-2
NFR3	The system shall have a user interface which is intuitive and easy to navigate.	test-ui-3
NFR6	The system shall hide the details of its implementation from the user.	test-ui-4
NFR7	The system shall be accessible to users with visual impairments.	test-ui-5
NFR16	The system shall have a simple installation process for all potential users.	test-qol-1
NFR8	The system shall complete the route and fuel consumption calculation in a reasonable amount of time ($<$5 seconds).	test-pr-1
NFR9	The system shall calculate fuel consumption as accurately as possible.	test-pr-2
NFR10	The system shall be available for use 98% (i.e. The system will be down for 1 day every month for maintenance) .	test-pr-3
NFR11	The system shall be able to respond to invalid or unconventional inputs.	test-pr-4
NFR13	The system shall be able to handle an infinite amount of users.	test-pr-5
NFR14	The system shall operate as long as possible.	test-pr-6
NFR15	The system shall be able to run on all modern browsers.	test-t-1
NFR17	The system shall be available to download for the public via GitHub.	test-t-2

Requirement #	Description	Test ID(s)
NFR20	The system shall be compatible to run between Windows, Mac, and Linux-based operating systems.	test-t-3
NFR18	The system's code shall be well documented and clear to be easily understandable for any developer maintaining the code.	test-m-1
NFR19	Supporting documentation shall be available on GitHub, which includes a user's guide on how to use the application.	test-m-2
NFR21	The system should comply with the Google Maps Platform Terms of Service.	test-l-1

8 Code Coverage Metrics

8.1 Front-end Testing

Coverage Summary	
Statements	76% (777/1022)
Branches	100% (2/2)
Functions	100% (9/9)
Lines	76% (777/1022)

8.2 Back-end Testing

Coverage Summary	
Statements	100% (108/108)
Branches	100% (1/1)
Functions	100% (13/13)
Lines	100% (108/108)

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

- Utsharga: A lot more external testing was required compared to what we had planned. These changes had occured due to us lacking of insight. Through our user survey we verfied and validated our results and we plan to make changes according to that.

- Bilal: The VnV plan was different from the VnV as the requirements implemented have changed for the implementation so the test results are different from expectations. A lot of changes occured due to infeasible implementation of certain core features and changes in focus for the app. All these changes are not possible to predicted before hand unless multiple small poc are made for each feature which is obviously not doable in the timeframe of this project.

- Pranay: The testing we conducted for the VnV differed considerably from the original Verification and Validation Plan. Since creating our VnV plan nearly four months, the Greenway app has undergone several considerable changes with regards to the variety of features we included as well as minor changes to the core functionality of the app. Although the changes to the core functionality felt minor, we failed to meet many of the functional requirement tests outlined in our VnV Plan. These changes were hard to anticipate and it seems unlikely that we will be able to anticipate these changes in future projects. Some issues are simply impossible to predict until you have progressed further and garnered more insight.

- Priyansh: The VnV plan differed from the actual VnV process because the requirements that were implemented had been changed over the course of several months. This resulted in test results that were different from what was expected. Many changes had to be made due to the change in requirements for certain core features and a shift in focus for the application. Anticipating all of these changes beforehand was not feasible without creating multiple small prototypes for each feature, which was not possible within the project's timeframe.

- Sharjil: The activities that were conducted for VnV differed significantly from the original Verification and Validation Plan. Over the past four months since we created our VnV plan, the Greenway application has undergone several modifications in terms of the range of features included as well as minor tweaks to its main functionality. Even though the changes to the main functionality were minor, we were unable to meet many of the functional requirement tests that were specified in our VnV Plan previously. These changes were difficult to predict, and it appears unlikely that we will be able to anticipate such changes in future projects. Some problems are simply impossible to predict until we make further progress and gain more insight.

- Jash: There were several differences between the Verification and Validation (VnV) plan and the actual activities that are conducted for VnV. Since the VnV plan is typically developed early in the project lifecycle when there is limited information about the system being developed, it makes sense that the plan is revised as more information becomes available. In our case, the system requirements have changed, new risks were identified, or issues arose during the VnV process that require additional testing or analysis which resulted in our VnV process being different. We also made changes to the test cases, test procedures, or acceptance criteria as some tests failed. To anticipate these changes in future projects,we need to regularly review and update the VnV plan throughout the project lifecycle as new information becomes available as we develop our app. This can help to identify any changes that may be required and ensure that the VnV activities are aligned with the project objectives.