2015

Test Plan & Results Document

VERSION 1.0.0

XAVIER THOMPSON, RICHARD RUSLI, KAI LIN, ANDREW LOH

TEAM 22

Contents

Introduction

Testing of the app will include running through each individual feature multiple times to judge whether or not it satisfies the expectations. Running the application as a whole will also take place to test if each individual feature works as expected with each other. The most important features to test are the tabs (where all the crucial information is stored in each page), the GPS integration (to record routes) and the start/stop button (to record routes). The other features to be tested are the speed calculations, save and search functions.

Each feature should be tested multiple times while the code is being developed, and once the feature is finished, it should be tested at least 5 times to ensure that results are consistent. Once another feature is complete, testing should include the other features as well, to progressively check that the features are able to work together.

Types & Approaches of Testing

Unit Testing
Unit Testing is used to test if each individual feature/function is able to work on its own. Time period is for the majority of the development cycle, and will only cease to occur once all the features have been coded. Xavier Thompson is in charge of this.

Integration Testing
Integration Testing is used to test if each feature is able to work with each other. This type of testing should not take very long (e.g. a couple of minutes each test after every new feature made). Richard Rusli is in charge of this.

Usability Testing
Usability Testing is used to test if the interface is user friendly. Should take around 30 minutes to an hour to decide on favourable designs and colours. Andrew Loh is in charge of this.

Regression Testing
Regression Testing is used to test if modifications do not damage other parts of the code. Timeframe is similar to integration testing. Kai Lin is in charge of this.

Test Plan

The functional requirements that tested will be the location tracking function, visual display of the route through the Google Maps API, clearing the displayed route (from both the map and memory), saving the route with complementary data to localstorage, retrieving the data from local storage and displaying the retrieved route and data.

Unit testing will be commonplace throughout the development of the application as it is imperative that each function will be able to perform as intended every time it is called. This is completed by passing known quantities through the functions to get a known result, if the function doesn't return the expected result then the code is revised until it does. This ensures that as the functionality of the application grows, all of the completed functions will, by themselves output what is required of them, nullifying the function as a source of error.

The other three testing types (Integration, Regression and Acceptance), while important, rely on the fact that the unit testing has been completed. Integration tests are completed by observing how the functions react with each other and if they output expected results and needs to be completed each time a new function is added to the website. Usability testing plays a vital role in the look and feel of the application. Without this testing the application could be nigh unusable due to a lacking interface or horrible color palette. A number of designs and color palettes will be tested against each other until a best fit is decided upon. This test only needs to completed once when the rest of the code has been completed. Regression testing is functionally similar to integration testing however is more aimed towards how the code builds up over time rather than how new completed functions interact. This test needs to be done constantly as new functions are added to more sure previously completed functions do not break upon the addition of new the new code.

Test Cases and Results

Test Case: Test that the Track Location Function tracks accurately.
Description: This test case tests to check whether the track location function tracks the location of the user, continues to when kept on and is able to be turned off. Reliant on console logs to know if passed or failed.

Step	Description	Expected Result	Pass/Fail
1	Click the 'Start' button to start the tracking function	The tracking function starts and begins track the current position.	Pass
2	Do nothing	The function continues to track the user	Pass
3	Click the 'Stop' Button to stop the function	The function should stop tracking	Pass

Test Data - Set Track_Location 1		
1	The function Tracks as intended	Fail

Remark: The function would not stop, however tracks accurately.

Test Data - Set Track_Location 2		
1	The function Tracks as intended	Pass

Test Case: Display Function.
Description: This test case tests to check whether the display function outputs the stored route into the Map Displayed by the Google Maps API. A marker needs to be placed at the start and end of the route.

Step	Description	Expected Result	Pass/Fail
1	Click the 'Start' button to start the tracking function	The tracking function starts and begins track the current position. The first position is marked with a marker.	Pass
2	Do nothing.	The function continues to track the user and plot the path accordingly	Pass
3	Click the 'Stop' Button to stop the function	The function should stop tracking and plot the final point with a different coloured	Pass

		marker.	

Test Data - Set Display_Route 1		
1	The function Tracks as intended	Fail

Remark: The display function did not plot the route, only the first marker.

Test Data - Set Display_Route 2		
1	The function Tracks as intended	Fail

Remark: The display function plotted the route and both markers, however the last marker was the same colour.

Test Data - Set Display_Route 3		
1	The function Tracks as intended	Pass

Test Case: Clear function
Description: This test case tests to check whether that deletes the displayed route, markers and route from memory.

Step	**Description**	**Expected Result**	**Pass/Fail**
1	Have a route saved in the memory, and press the clear button.	The displayed route will be cleared from the map.	Pass
2	Clicking start again	The path will not show the previously cleared path joining to the new coordinates.	Pass

Test Data - Set Clear 1		
1	The function doesn't clear the markers or polyline.	Fail

Remark: Upon further research you have to implement the google Maps own command to remove it from the map, reloading the map won't work.

Test Data - Set Display_Route 2

1	Works as intended except only the first marker is deleted	Fail

Remark: It appears that it only remembers one of the markers, will add the marker coords to an array in hope that that will clear all of them.

Test Data - Set Display_Route 3		
1	The function clears as intended	Pass

Test Case: Save to Memory function.
Description: This test case tests to check whether the name is inputted, whether the time, distance, speed, calories burnt and the save to local storage saves properly.

Step	**Description**	**Expected Result**	**Pass/Fail**
1	While not having a route on memory, click the save button	Text is outputted saying ”Nothing to Save” and nothing is saved.	Pass
2	While tracking, click the save button	Text is outputted saying “Can’t save while tracking” and nothing is saved	Pass
3	While having a route on memory, click the save button	An alert box appears with the caption “Please enter a name” and nothing is saved.	Pass
4	While having a route in memory, enter a name and click the save button	The function should save with correct values for time, distance, average speed, calories burnt and includes the route.	Pass

Test Data - Set Display_Route 1		
1	No text outputted, not saved	Fail
2	No text outputted, not saved	Fail

3	Works as intended	Pass
4	saved correctly, however with incorrect values	Fail

Remark: Something is wrong with the .innerHTML which is not allowing the text to be displayed

Test Data - Set Display_Route 2		
1	Works as intended	Pass
2	Works as intended	Pass
3	Works as intended	Pass
4	Saved correctly, still with incorrect values	Fail

Remark: Upon further investigation the units for the additional data are off.

Test Data - Set Display_Route 2		
1	Works as intended	Pass
2	Works as intended	Pass
3	Works as intended	Pass
4	Works as intended	Pass

Test Case: Retrieve from Memory function
Description: This test case tests to check whether the function pulls the object from local storage and deposits in an array with all the other data from local storage to be put into an array to be used later.

Step	**Description**	**Expected Result**	**Pass/Fail**
1	The data is retrieved on page loading	An object is added into the array with an element for each piece of data.	Pass

Test Data - Set Display_Route 1		
1	The function returns undefined	Fail

Remark: The code to acquire the key does not work correctly.

Test Data - Set Display_Route 2		
2	The function returns [object object]	Fail

Remark: JSON is not configured correctly as it is not returning the correct thing however is returning something.

Test Data - Set Display_Route 3		
3	The function Tracks as intended	Pass

Test Case: Display Retrieved Data function
Description: This test case tests to check whether the function displays the data in a table, in which data is able to be selected, and upon a button press, either have the route displayed or the the data deleted.

Step	**Description**	**Expected Result**	**Pass/Fail**
1	On loading the page (when data is available in local storage).	Data is displayed in the table and the map loads in however nothing is displayed.	Pass
2	On loading the page (when no data is available in local storage).	The table loads in with 'No data' in the row and the map loads in however nothing is displayed.	Pass
3	When no radio button is selected and either the display or delete button is pressed	The delete and display function display a warning alert that no run is selected.	Pass
4	When a radio button is selected and the display or delete button is pressed.	The run's data should be displayed or deleted respectively.	Pass

Test Data - Set Display_Route 1		
1	Works as intended	Pass
2	Works as intended	Pass
3	Buttons don't work, don't recognise the selection inside table	Fail

4	Buttons don’t work, don’t recognise the selection inside table	Fail

Remark: Originally using checkboxes, the function didn’t work well on android and with jQuery which was used to make the selection possible, so the swap was made to linked radio buttons.

Test Data - Set Display_Route 2		
1	Works as intended	Pass
2	Works as intended	Pass
3	Radio buttons aren't linked	Fail
4	Radio buttons aren't linked	Fail

Remark: The radio buttons weren’t linked, allowing multiple rows to be selected

Test Data - Set Display_Route 2		
1	Works as intended	Pass
2	Works as intended	Pass
3	Works as intended	Pass
4	Works as intended	Pass

Conclusion

The current testing goals have been met, the unit testing for each function has been completed and as such each function performs as intended. However just because they all work in an isolated environment does not necessarily mean that the website will work in all situations, external factors can and will disrupt the flow of the application. In this way further testing may be necessary to ensure the application will work in all environments, because of the testing done however, it is known that each function works on its own.