Stakeholder Review II

Daniel Campos, Zoe Konrad, Nick Marton, Daniel Yoon

YOCO, Hopefully

22 April 2013

***Table of Contents***

Testing Doccuments…........................................................................................................2

Code Review………...........................................................................................................3

Status Report.......................................................................................................................7

Contribution Summary........................................................................................................8

Testing Documents

Test Plan:

Our goals for the tests are to show that the system meets user requirements and works under normal circumstances. We will measure this by whether or not specific sets of actions produce the promised functionality results.

The resources used for the tests are people, who will need to be using the app on an android phone, and a brief amount of time (our test cases are not time consuming.)

Our testing plan is to have various users serve at testers of the app. They will each follow all of the test cases which are made up of actions and the expected results (as seen by the user) for each action. If one of the expected results does not occur properly, the entire test case does not pass.

The plan is to have each YOCO member complete all test cases and recruit 2 friends per member that have android phones to complete them as well.

The schedule is to complete the testing during the week of April 22 and collect the results from the friends on April 26. The friends will be encouraged to first complete each test case and record results and then for them to continue to use the app at their own leisure and provide us with additional feedback.

Testing Template:

**Testing Iteration: \_\_ Date: \_\_\_\_\_\_\_**

**Tester: \_\_\_\_\_\_**

|  |  |  |
| --- | --- | --- |
| Test Case | Result (Pass/Fail) | Comments |
| Test Case 1 |  |  |
| Test Case 2 |  |  |
| Test Case 3 |  |  |
| Test Case 4 |  |  |
| Test Case 5 |  |  |
| Test Case 6 |  |  |
| Test Case 7 |  |  |
| Test Case 8 |  |  |

**Other Comments/Recommendations:**

Testing Specification:

**Test Case 1: Plot a Route to Final Destination**

Preconditions: None

Flow of events:

1. Open the app
   * map of current location displayed
2. In the start field: click ‘My Location’ button
   * your location is entered into the field as geo coordinates
3. In the finish field: enter address of destination
4. Click ‘Navigate’
   * map showing route outline
5. Click ‘Itinerary’
   * displays the step by step driving instructions of your route

**Test Case 2: Change Final Destination**

Preconditions: User plotted a route

Flow of events:

1. In the finish field: enter different address of destination
2. Click ‘Navigate’
   * map displays a different route
3. Optional: click ‘Itinerary’
   * displays the step by step driving instructions of your route

**Test Case 3: Bad Destination Address**

Preconditions: None or User has already plotted a route

Flow of events:

1. In the finish field: enter incorrect address
2. Click ‘Navigate’
   * error notification is displayed

**Test Case 4: Search for Stop options coming up**

Preconditions: User has created a route

Flow of events:

1. Click ‘RR’ icon
   * display Search screen with search parameters
2. Do not change search parameters from their default values
3. Click ‘Search’
   * display Results screen with specific results listed in order by distance

**Test Case 5: Search for Stop options based on Search Parameters**

Preconditions: User has created a route

Flow of events:

1. Click ‘RR’ icon
   * display Search screen with search parameters
2. Change search parameters from their default values
3. Click ‘Search’
   * display Results screen with specific results listed in order by distance
   * \*should be different from TC2 Results list

**Test Case 6: Change Search parameters**

Preconditions: User has created a route and has searched for stop options and is viewing the Results page

Flow of events:

1. Click ‘Modify’ button
   * redisplay Search screen with search parameters
2. Change search parameters
3. Click ‘Search’
   * redisplay Results screen with specific results listed in order by distance
   * \*again, should be different from TC4 or TC5’s Results list

**Test Case 7: Add a stopover**

Preconditions: User has created a route and has searched for stop options and is viewing the Results page

Flow of events:

1. Click on a place from the Results page
   * Display map with route plotted to new destination

**Test Case 8: Redirect from stopover back to Final Destination**

Preconditions: User has created a route and has searched for stop options and selected a result

Flow of events:

1. User reaches destination
   * Prompt window alerts that you have reached your destination
2. Click on ‘Continue to Final Destination’
   * Display map with route plotted to original destination

Preliminary Test Results

**Testing Iteration: 1 Date: 22 April 2013**

**Tester: Daniel**

|  |  |  |
| --- | --- | --- |
| Test Case | Result (Pass/Fail) | Comments |
| Test Case 1 | Pass |  |
| Test Case 2 | Pass |  |
| Test Case 3 | Pass | Should make the error message more explanatory |
| Test Case 4 | Pass |  |
| Test Case 5 | Pass |  |
| Test Case 6 | Pass |  |
| Test Case 7 | Pass |  |
| Test Case 8 | Fail | Functionality we still need to add |

**Other Comments/Recommendations:**

Usability could be improved with things as little as renaming the buttons

We have a bit more functionality to add but we are otherwise on track to complete by May 2

Code Review

During our code review, we received a few useful recommendations. The first, and rather obvious observation, was that our code was poorly organized. The classes didn’t really fit into well-organized files and our class diagrams were not correctly representative of the actual code. We tried

Another comment was that our commenting was very high-level which is okay, but we chose to expand more to document the inputs/outputs of each function as a result of this observation.

Our code was otherwise very readable. We followed good practices as far as naming and coding. Although we had very consistent code naturally, we had not formally established a coding standard so we were also recommended to do so.

---

We reviewed Chattermap’s code. Their code was well-organized, well-commented, and very readable (especially since we were very familiar with the environment: coding for Android in Eclipse.) It was very consistent throughout all of the code and they explained to us how they had used Android development Best Practices guidelines. They had high cohesion, low coupling, and strong data encapsulation. We also learned of a useful Eclipse plug-in that generates class diagrams automatically which would be able to solve one of the problems from our code review.

Status Report

*Achieved:*

Got settings worked in to the search

Functionality to divide and search all the locations on route based on specific parameter or categories

Minor changes to layout of interface

Reorganized code after code review

*Risks:*

None

*Plan:*

Begin testing as per our plan and schedule

Begin polishing usability of app (and aesthetics!)

Prepare for Transition deliverables and final presentation

Contribution Summary

*Daniel Campos*: Worked on XML parser and reorganizing the code.

*Zoe Konrad:* Completed testing documents including goals, plan, and schedule. Helped reorganize code and minor improvements to the UI.

*Daniel Yoon:* Worked on testing documents. Began preliminary testing.

*Nick Marton:* Worked on searching by category functionality and adding more to the UI.