Quality Assurance

Table of Contents

1. Overview....3
2. Automation/System Testing....3
3. User Testing....4
4. Integration testing....5
5. Software Tools Used....5
7. General Quality Assurance Procedure....6

1. Overview

We will be using different types of tests to ensure the application is functional and provides a friendly user experience. One of the main types of testing that we'll be using, is automation unit testing. Automation unit testing can be done using a third-party software which will be able to execute tests, report outcomes and compare results with earlier test runs.

There are many different tools available to help in performing these automatic testing's. Some of them are XCTest, KIF(Keep It Functional), and APPIUM. We will be going with APPIUM for our application. It allows you to write UI tests for your mobile app. Some other types of testing that we will be using, besides automatic unit testing, are Manual Testing, and Integration testing.

Due to time constraints – in Version 3 we have done extensive simulator testing but didn't get around to using APPIUM.

2. Automation/System Testing

For our Unit and System testing, we decided to implement all of these tests by March 21st. By this time, we will have all our deliverables and most important features completed and functioning. We will review the results and implement the required changes.

We will be reviewing the following properties using automatic testing:

1- ***Diary:***

a. We will Inject data into it and test to see if the following information is saved:

i. Post

ii. Date

iii. Mood

2- ***Quiz:***

   a. There are 3 choices of quizzes, we will inject each quiz with answers and analyze if we receive the proper “Grade”.
      i. We will apply this test with the same answers multiple times for each quiz to assure the answers are consistent.
   b. Then we will check if the following information is saved and accurate:
      i. Quiz type
      ii. Date
      iii. Result
         1. We will apply the same test to each quiz multiple times; in order to assure we receive the same answer every time for an accurate result.

3- ***Cue Creator:***
   a. Inject different tags into the cue creator and asses how relevant the suggest solutions are

We will use Manual Testing for the following properties:

1- ***Timer Meditation:***
   a. We will input a time and assess if time countdown is maintained and accurate.
   b. Assure the user can pick the playlist they desire from the provided library and test to check if every playlist is functioning

2- ***Guided Meditation:***
   a. Assure the step tags are displaying when they need to and are clear and easy to read.
   b. Check the rating system and check if the result is saved.
   c. Again, assure the user can pick the playlist they desire from the provided library.

3- ***Haptic Meditation:***
   a. Assure timer accuracy and that you can stop when desired

3. User Testing

Finally, after implementing all the above testing, we will do our User testing. Between March 25-28th we will bring in 3 different groups of 5’s to test the app. It will be a group of 5 per day. These individuals will be people we already know (friends). We will bring them into the CSIL Labs one at a time during the afternoon. The user will be provided with a device with this app and we will have them do the following:

1- *Attempt each Meditation type (Timed meditation, Guided Meditation, Haptic Meditation)*
   a. Then answer the following questions after each:
      i. Was it easy to navigate/use this meditation type?

ii. Was there enough variety of music to choose from?
iii. ***ONLY FOR GUIDED MEDITATION:***
   1. Were the steps clear and easy to follow?
iv. How was your general experience? Rate from 1-5

2- *Allow them to attempt each test and view their results (3 tests in total) then ask the following:*
   a. Where the questions straight forward?
   b. Was it easy to navigate through the questions and submit your answers?
   c. Are the results easy to understand the way they're displayed?
   d. After viewing your results what do you think?
   e. How was your general experience? Rate 1-5

3- *Allow them to write in a diary and then pull up the diary again to view, after it is saved. Then ask them the following:*
   a. Do you think this would be useful to you as a user?
   b. Is it easy to navigate through creating a new diary entry and opening an older entry?
   c. How was your general experience? Rate 1-5

4- *Allow them to press the panic button and view the contacts list available. Then ask:*
   a. If you were to press this during an emergency, would you be able to use this?
   b. Do you think it would be simple enough to press on a contact and call them?
   c. How was your general experience? Rate 1-5

5- *Review the available resources, and search around for different topics which we will provide at the time. Then ask:*
   a. Was it easy to search for a topic?
   b. Was it simple enough to select a resource and read it?
   c. Is everything clear? Font wise.
   d. How was your general experience? Rate 1-5

4. Integration testing

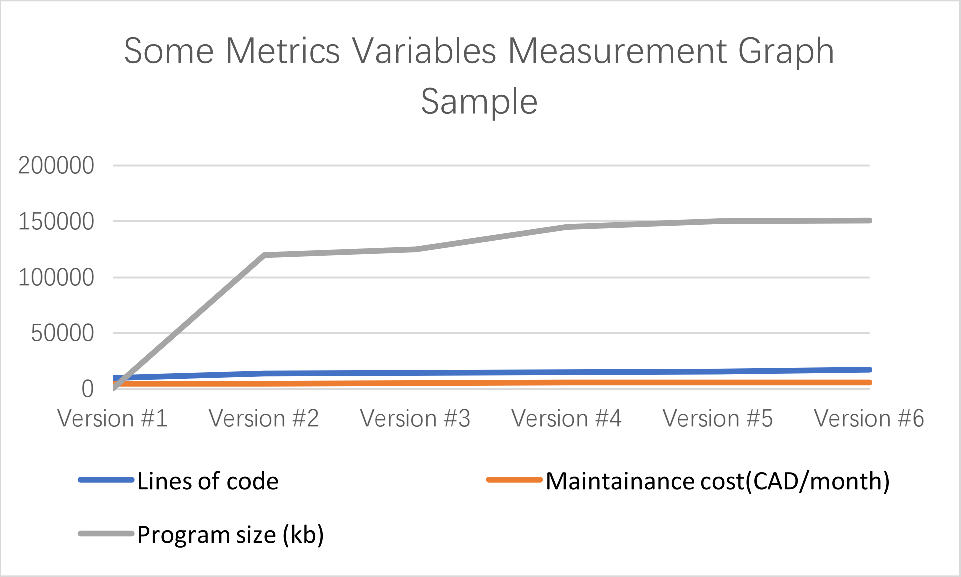
For our application, we do the unit testing to make sure the individual modules running in the expected ways. Then, we use the top down approach (see the reference) to integrate each of the separated modules together without the GUI testing. If the integration testing run perfectly, then, we test the GUI finally. http://www.softwaretestinghelp.com/what-is-integration-testing/

We will also incorporate integration test cases that properly mimick the results that a user would typically input in previous sections and display it accordingly.

5. Software Tools Used

We pick “Taylor” to measure the software metrics. It is available on Github: https://github.com/yopeso/Taylor/.

Here is a sample of metrics measurement graph, I pick some simple, easy variables. For our final project measurement, we would use some complex metrics variables that provides from our tools.



7. General Quality Assurance Procedure

In-house testing will occur for version 1, 2, 3 5 days before each assignment deadline.
User testing will occur 10 days before April 5th 2017.

In addition, we need to check whether the application is easily accepted by public. For example, we will pick some testers to use our application and request for their feedback, just like the game testers. At the end, we may try to find some better ways to improve the code, for example, we could use less RAM, less CPU iterations(performance).