**FLOAT**

**Test Plan Document**

Aaron So 17150137

Clarence(Junwei) Su 36387132

Ming Hin Matthew Lam 33056145

Rachel Yeo 30032122

Richard Xie 55786140

Samuel Farinas 17721144

Selina Suen 14022140

**Introduction**

The concept of Float is to create a fundraising system where users can view, contribute to and start charity campaigns. As a consequence of how these charity campaigns are structured, the project itself is to rely heavily on the locations of each user, user interaction and secure payments. As our implementation of Float will be in the form of an application targeted towards Android devices, the previously mentioned functionalities will be supported by the following: a Facebook login system for users, a PayPal interface for receiving payments, a MySQL database for storing information regarding users and campaigns, as well as Google Maps to connect users and campaigns to locations. An extensive user interface will also be created to connect all subsystems together.

**Verification Strategy**

In the beginning stages of verification and testing, we plan to talk to charity organizations regarding the idea of the app and our stated requirements, as to receive feedback on if the idea itself is something that will be of interest to the common population. If this proposed meeting goes well, we will also aim to have the chosen charity organizations be part of later user acceptance testing. We have chosen to focus on charity organization workers as our first point of verification as, even though they may not cover the entirety the project’s target audience, they themselves are well versed in the manner of charity work and what motivates people to become involved and donate.

To continue on the previously mentioned topic of user acceptance testing, we plan to further verify that the application meets user’s real needs by running multiple instances of beta testing during the course of the project. Such testing will be directed towards our main target audience of reasonable well-off young to middle aged adults. Even though, statistically speaking, those that are elderly or aged are more likely to show interest in charitable acts and donations, such a demographic usually does not show much interest in electronics and mobile applications, which will be the main platform of expansion for our project.

With regards to the multiple instances of user acceptance testing, we are currently planning to have three testing/feedback sessions which are as follows: testing of the prototype, testing of the pre-release of the application as well as testing of the actual release of the application. During these testing sessions, we will strive to receive detailed feedback from all participants. Though we have not fully developed the questionnaire we plan to provide to the participants, we will be looking to find feedback concerning if the app is something the public would be interested in, and what changes/add-ons should be made to the idea of the project. On top of such questions regarding the general overview of the project idea, we will also seek feedback regarding the app’s user interface and any bugs that are apparent. Upon receiving this feedback, we will then do our best to integrate suggested changes into the application.

**Non-functional Testing and Results**

Reliability testing: We plan to test our application for reliability amongst our group members. One measure of reliability is that our application will show the same campaign points for users of the same location. We will test this by initializing a few campaigns ourselves and making sure that if two or more users are in the same location, they will all see the same campaigns. If users are in a different location, they should therefore see a different set of campaigns in their area.

Usability testing: We will obtain feedback on the usability of our application through input from classmates and friends. We will mainly focus on how easy it is to understand and learn to complete the main functionalities of our application: (1) Browse campaigns (2) Select a campaign (3) Spread a campaign with the option of donating (4) Start a campaign. We will also take their input on the overall satisfaction of our application.

Load testing: We will simulate load in our application by starting a large cluster of campaigns in an area. A user should log into the application without being impacted by the volume of the campaigns (there should not be a significant impact on the performance of the application such as lags, and the user should be able to still easily navigate through the campaigns).

Compliance testing: We must ensure that our application stores data safely and securely.

**Functional Testing Strategy**

We plan on testing the java project files through the Android debugging framework. We will be performing unit testing, integration testing as well as system testing. A unit test will be performed at a preliminary stage, following any new code or modules written for features. Following this, integration testing will be performed on the feature as a whole. As a final step, we will then integrate the new feature into any existing code for the app, before performing a system test, to ensure that all new and previous features are working properly. This above described process will be performed at least once a sprint, but may be performed more frequently depending on pace of implementation and project progression.

Our test approach will consist of writing JUnit tests and running them either on the JVM or on a physical device. As system testing is black box testing, we will write a sequence of actions and their expected result, running them through the app to see if it performs as desired.

As we are currently using a team organization website called Asana to track our progress and assign tasks, we will also use this platform to track any bugs we may find, and assign them to a certain team member. In this system, any posts regarding bugs will also come with tags indicating the status of the bug as well as its category.

**Adequacy Criterion**

We have set a following list of test obligations that must be satisfied in order for our application to meet its adequacy criterion.

Amongst each subgroup, members will initiate their own testing to ensure all methods written are tested before combining the components together. This includes separate testing for the Facebook login, the database, PayPal payment, and the GUI.

The use cases will be tested upon combining the application together. We will cover each use case we have provided in our Project Requirements documentation. Additional, to assure the quality of the test suit, we would aim for a high statement and branch coverage(i.e cover all essential statements and branches of the production code). We understand that high coverage does not necessarily means better test suit but it is one of the necessary conditions for a test suit to be good. For method with complex boolean expression, we will employ MCDC (Modified Condition Decision Coverage) to cover as many possibilities as possible. In the end, we would run our test suit against the mutation test to ensure the ability of our test cases to detect bugs. We will do that by using PIT.

**Test Cases and Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Requirement Purpose | Action/Input | Expected Result | Actual Result | P/F | Notes |
| 1 | To ensure the user has an account | User logs into the application using Facebook | The user successfully logs on and is brought to the main page of the application. The information provided by their Facebook account can be accessed by our application for processing |  |  |  |
| 2 | Payments can be made to charities | User enters the amount of money they wish to donate to a campaign | The user is prompted to submit their preferred payment option and the transaction successfully goes through via PayPal |  |  |  |
| 3 | User can start a campaign | The user presses the “Start Campaign” button | The user is brought to a new page in the application where they can fill a form to start a campaign |  |  |  |
| 4 | User can select a charity through the map | The user taps a marker on the map | A window fills half of the screen displaying the campaign’s picture, title, user, charity. A ‘details’ button and a deselect ‘x’ button is also in the window. |  |  |  |
| 5 | User can view charity information | The user taps the “list” button in the bottom-right of the screen. The user scrolls and taps on a campaign. | The map is replaced by a list of campaign, each entry containing same elements as in test 4. The list scrolls to reveal more charities. The list view is replaced by the campaign page with same info as in test 4 and a text description. |  |  |  |
| 6 | User can view charity information | After selecting a campaign, the user clicks on the charity’s logo/name button. | The campaign page is replaced by the charity page, containing the charity logo, a “Verified charity” status, statistics, and a blurb (small write-up). |  |  |  |
| 7 | User can view account information | At any page, the user clicks the person icon in the top-right of the screen. User slides finger up and down the screen | The present page is replaced by the user’s page containing the user’s username, thumbnail picture, statistics, blurb (small write-up), and settings (revealable through scrolling). |  |  |  |