|  |
| --- |
| **Palmly Mobile Application:**  Software Test Plan Procedure (v 1.0) |

Project: Palmly iOS Application Development

Date: March 2, 2020

Prepared by: Nguyen, Katie; Ruiz, Sofia; Tolliver, Moriah

|  |
| --- |
| **8.1 Introduction** |

This section of the document describes the testing procedures used during development of the Palmly application. The testing procedures include unit, integration and acceptance test levels wherein each tested component is evaluated at each level. This section will also detail how each component is tested and the steps taken to prove the functionality of the application.

|  |
| --- |
| **8.2 Unit Test Plan** |

The computer software components to be tested include the interface on the front end and the neural networks on the back end. The front end consists of the reading capture page, the reading output page, and the breakdown, lifestyle, and personality pages for each line. The back end consists of three separate neural networks for the head, heart and life lines.

**8.2.1 Unit Tests Planned**

The view controllers have automated tests created and run through the XCTest framework. The controllers determine the outputs for the lifestyle and personality pages. The view is manually tested for spacing and reaction to different screen movements. For the controllers, tests will contain stubs for readings that will eventually come from the neural networks hard coded inputs. The pages for these controllers expect many outputs for one input so the user has a new experience each time they open the application. To test this behavior, the controller test will run the main functions of the controllers, many times with the same input and check that the same output only appears 25% of the time.

The neural networks are tested for valid output using the label\_image.py script located in the TensorFlow training files.

**8.2.2 Unit Test Procedures**

Steps to Run Network Tests:

1. Navigate in the terminal or command line window to the directory scripts directory where the networks are stored on the machine.
2. Run the command:

|  |
| --- |
| > python -m scripts.label\_image \  --graph=tf\_files/retrained\_graph.pb \  --image=tf\_files/<line name>/<expected label>/<image name> |

1. The test passes if no reading is at 100%, at least two readings are at or above 0.01%, the reading with the largest percentage is the expected label.
2. Repeat for all images in the test folder.

Steps to Run XCTests:

1. Navigate to the NetworkTests or ControllerTests folder in the XCode project.
2. Find the where the test function is located in the test file.
3. Each test function is noted by a diamond on the left side of the code editor. Click the diamond to run the test.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Module Name | Inputs | Expected Outputs |
| testPersonality | personalityController | An array of tuples: (probability, label) where the first tuple is the life reading, the second is head, and the third is heart. | [] for an empty array or an array of 5 article objects. |
| testLifestyle | lifestyleController | An array of tuples: (probability, label) where the first tuple is the life reading, the second is head, and the third is heart. | [] for an empty array or an array of 3 strings. |

|  |
| --- |
| **8.3 Integration Test Plan** |

**8.3.1 Integration Tests Planned**

Checking connections between the different sections or interfaces of the application.

1. Testing connection on upload screen to:
   1. Upload a picture - click the button that should allow you to access camera roll, which should allow you to do so and select a picture from there
   2. Take a picture- click the button for taking a picture - should open camera
2. Testing connection from uploading a picture to palm reading interface - select image and you should be taken to the palm reading interface
3. Testing connection from taking a picture to palm reading interface - take an image and you should be immediately taken to the palm reading interface
4. Testing connection from reading interface to:
   1. Breakdown of reading page - when clicking the breakdown of reading page - should be taken to that tab
   2. Personality section - when clicking the personality tab - should be taken to that tab
   3. Lifestyle component page - when clicking lifestyle tab - should be redirected there

**8.3.2 Integration Test Procedures**

a.1) Uploading a picture

i) Open application

ii) Click the button on the screen to upload a picture

iii) User should be redirected to camera roll to select a picture

a.2) Taking a picture

i) Open application

ii) Click the button on screen to take a picture

iii) User should be redirected to their camera where they can take a picture

b) Connection from uploading picture to palm reading interface

i) Select a picture from camera roll

ii) Should be redirected immediately to palm reading

c) Connection from taking a picture to palm reading interface

i) Take picture on camera

ii) Should be redirected immediately to palm reading

d.1) Connection between reading interface to breakdown tab

i) On reading interface click “breakdown” tab

ii) Should be redirected to breakdown

d.2) Connection between reading interface to personality section

i) On reading interface click “personality” tab

ii) Should be redirected to personality

d.3) Connection between reading interface to lifestyle component tab

i) On reading interface click the “lifestyle” tab

ii) Should be redirected to new page

|  |
| --- |
| **8.4 Acceptance Test Plan** |

**8.4.1 Acceptance Tests Planned**

1. Uploading a picture - user should get a reading and be allowed to access all the tabs available
   1. User has not allowed application to access camera roll - message to allow access
   2. User selects a picture - redirected to reading interface and a reading should be displayed
   3. Reading should include all three lines
   4. Breakdown tab should include the breakdown of each of the three lines and what that means
   5. Personality tab should already have been populated and should include specifics of the user’s characteristics
   6. Lifestyle tab should already have been populated and should include specifics based on the user’s personality
2. Taking a picture - user should get a reading and should be allowed to access all the tabs which will all have information
   1. User has no allowed access to the camera - message to allow access
   2. User takes a picture - should be redirected to reading interface and a reading should be displayed
   3. Reading should include all three lines
   4. Breakdown tab should include the breakdown of each of the three lines and what that means
   5. Personality tab should already have been populated and should include specifics of the user’s characteristics
   6. Lifestyle tab should already have been populated and should include specifics based on the user’s personality

|  |
| --- |
| **8.5 Test Configuration Control** |

The tests for the Palmly iOS Application will reside within the Xcode project itself. Within the application directory, there will be a ‘palmly-appTest’ directory in addition to a ‘palmly-appUITest’ directory. The app tests directory will contain all of the tests that are relevant to the expected outcomes, including possible error messages, from the neural networks that are being called within the app. In this sense, this test suite will serve to ensure that the output of the networks is correct throughout various use cases. On the other hand, the UI test directory will contain tests that track a user’s experience throughout the app to ensure that all buttons, links, and the progression throughout the screens is correct, both from a developer and user’s perspective. These tests, all written in Swift, can be run directly from Xcode itself and will test for coverage as well as accuracy.

|  |
| --- |
| **8.6 Items Not Tested** |

At this point in the project, the correctness and accuracy of the trained neural networks will not be tested. As explained above, we will test the format of the output from the neural nets, but we will not be testing how closely the result matches the team’s perceived expected outcome. This is due to the fact that testing the neural networks at this point in the project will not change the user’s experience with the app. Therefore, throughout this phase of the Palmly project, the team will only be testing the output, connections, and functionality of the application itself, rather than the accuracy of the neural networks themselves. If time allows, the team can refine and retrain the neural networks to increase accuracy; however, this is not the main goal of this section of the project.

|  |
| --- |
| **8.7 Test Verification Matrix** |

|  |  |  |
| --- | --- | --- |
| **Requirement Number** | **Requirement Name** | **Test Name** |
| 1 | Neural Network Output | Label Image Script |
| 2 | Reading Results Page | App Test: Page Population with Results |
| 3 | Breakdown Pages | Test Breakdown |
| 4 | Personality Pages | Test Personality |
| 5 | Lifestyle Pages | Test Lifestyle |
| 6 | Pagination Tabs | UI Tests: Connections/Links Between Controllers |