|  |
| --- |
| **Palmly Mobile Application:**  Requirements Specification (v 1.0) |

Project: Palmly iOS Application Development

Date: February 10, 2020

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

|  |
| --- |
| **5.1 Requirements Introduction** |

This document contains the system requirements for the building of what will ultimately be the Palmly Mobile Application. The entire project will be built as an Xcode project, allowing the members of this team to develop the Palmly palm reading application as an iOS application.

**5.1.1 Document Overview**

This document will be organized as follows. The following section will detail the requirements for each component of the system.

The following language shall be used to specify requirements:

**5.1.1.1 “Shall”**

This specifies a mandatory requirement that must be fulfilled by the project.

**5.1.1.2 “Should”**

This specifies a requirement that may or may not be satisfied by completion of the project.

**5.1.1.3 “Will”**

This declares a design goal describing the way in which other requirements accomplish a purpose. These goals may or may not be met by completion of this project.

|  |
| --- |
| **5.2 Functional Requirements** |

The completed system will be composed of an Xcode project that the members of the team will continue to develop in order to produce the iOS application. Additionally, the project will produce the graphical user interface that will allow the user to interact with the three trained networks from the Palmly Neural Network Training Project. In this regard, the user will be able to retrieve their individualized palm reading and accompanying lifestyle tip section via the GUI.

**5.2.1 Xcode Project**

The Xcode project will house the development of the Palmly application itself. The three trained neural networks will be incorporated into the backend functionality of the app in the Xcode project as well. The completed app will produce readings based on the hand line classifications that the neural nets return, and the user will be able to scroll through corresponding reading and lifestyle info based on these results. The app can be previewed via the iPhone simulator in the Xcode development environment.

5.2.1.1 The Xcode project shall maintain all code and external files relevant to the development of the Palmly application.

5.2.1.2 The Xcode project shall contain the storyboard mapping out all pages the user can click through in the application.

5.2.1.3 The Xcode project shall contain the three trained neural networks from the Palmly Neural Network Training Project.

5.2.1.4 The Xcode project shall contain all of the source images that will be featured in the application.

**5.2.2 Graphical User Interface**

The GUI will contain an initial upload screen where the user can select or take a palm image to be analyzed by the three neural networks. From there, the user can navigate to different pages via a tab selection bar located at the bottom of the screen. Each page will contain a breakdown of the reading in general or as it pertains to a specific hand line. Additionally, each specific hand line page will contain tabs to view the overall breakdown of the reading, a personality section, and a lifestyle component page.

5.2.2.1 The GUI shall contain an initial launch screen where the user can upload a palm image or take one themselves.

This page will also display instructions of the requirements the palm image must meet.

5.2.2.2 The GUI will contain four tabs located at the bottom of the screen that the user can click on to navigate around the app.

The GUI will contain a general reading page.

The GUI will contain a life line page.

The GUI will contain a head line page.

The GUI will contain a heart line page.

5.2.2.3 The GUI shall contain three tabs within each of the specific hand line pages.

The GUI will contain a breakdown page.

The GUI will contain a personality page.

The GUI will contain a lifestyle page.

|  |
| --- |
| **5.3 Performance Requirements** |

Since the target age are those in the Millennial Generation and Generation Z, the application needs to respond as quickly as other applications that people in this age group typically use, while still offering a simple and easy-to-use interface. The majority of the performance requirements will rely heavily on the speed of the MobileNet API as well as transitions between screens and different data displays.

**5.3.1 Reading Results**

5.3.1.1 The reading results shall return within 5 seconds of the user submitting their image.

Once the user has submitted the image, the result should appear on screen in at most 5 seconds. This time includes the instant the user submits the image to when the results appear on the page. This does not include situations for which the image is rejected by the system due to low quality.

5.3.1.2 The primary reading shall have an accuracy at or above 55%.

This accuracy will be output from the API. Any reading whose primary reading is below 50% accuracy shall provide two subsequent readings with accuracy at or above 25%. Any readings for which this is not possible, a disclaimer shall appear on the screen about the discrepancy in reading.

**5.3.2 Palm Line Pages**

5.3.2.1 Each palm line page shall appear within 1 second of its tab being clicked on.

Upon the user clicking the page’s respective tab, if the reading has already been received, the page along with all of its contents should appear within 1 second. This time limit does not include the time it takes a reading to be fetched from the API.

5.3.2.3 Each palm line page shall be populated prior to any user action, if the most recent reading is a cached reading.

If the user has not submitted an image for a new reading in their current session, each page shall be populated with the contents from their last reading. If the user has no cached readings, each page’s header and footer should be present with directions on how to submit an image for reading.

5.3.2.4 Each palm line page shall change its content within 5 seconds of a new reading being returned from the API.

Once the user has submitted an image, each palm line page shall populate within 5 seconds even if it is not the current page.

**5.3.3 Breakdown Page**

5.3.3.1 The breakdown page shall populate within 5 seconds of a new reading being returned from the API.

Once the user has submitted an image, the breakdown page shall populate within 5 seconds even if it is not the current page.

**5.3.4 Personality Page**

5.3.4.1 The personality page shall populate within 5 seconds of the initial reading being returned from the API.

Once the user has submitted an image, the personality page shall populate within 5 seconds even if it is not the current page.

**5.3.5 Lifestyle Page**

5.3.5.1 The lifestyle page shall rotate articles every time the user leaves the app then returns.

Upon the user leaving the application and returning, the contents of the lifestyle page should have shifted to reflect any new content that may have been added to the platform. If no new articles have been added, the contents may shift to a random ordering of articles.

**5.3.6 Pagination Tabs**

5.3.6.2 Each tab shall highlight according to what page the user is currently on.

This includes if the page changes due to built-in iOS pagination or clicking on the tabs within the application.

5.3.6.3 The tabs should be fixed in place as the user zooms, scrolls, or rotates within the interface.

When the user rotates their device, the icons may rotate to reflect the orientation of the device. Once the device’s orientation is settled, the icons should not rotate and the tabs should not transform. This includes responses like “swipe”, “zoom” and “scroll” that are built into iOS devices.

|  |
| --- |
| **5.4 Environment Requirements** |

The Palmly application will be created for iOS so that the target audience can easily use it when desired. It will be created using XCode so that once finished it can be deployed on the iOS app store so that any user who has an iOS device can download and use it.

**5.4.1 Development Environment Requirements**

The Palmly application shall be created using XCode version 11.3. It will be programmed in Swift.

**5.4.2 Execution Environment Requirements**

The Palmly application shall be able to execute using any iOS device.