***System Test Plan***

***For***

**Helping Hand**

**Team members:**

Elizabeth Jackson,

Gabrielle Stoney,

Gardar Benediktsson,

Harrison Dinius,

Jason Hansen.

|  |  |
| --- | --- |
| Version/Author | Date |
| V1.1; Harrison Dinius | 11/09/2020 |
| V1.2; Gabrielle Stoney, Gardar Benediktsson, Harrison Dinius, Jason Hansen, Elizabeth Jackson | 11/10/2020 |
| V2.1; Elizabeth Jackson | 11/17/2020 |
| V2.2; Gabrielle Stoney, Gardar Benediktsson, Jason Hansen, Harrison Dinius | 11/30/2020 |
| V3.1 Elizabeth Jackson, Garðar Benediktsson | 3/10/2021 |
| V3.2 Gabrielle Stoney | 3/11/2021 |
| V4.1 Harrison Dinius, Lizzy Jackson | 4/20/2021 |

**Table of Contents**

[1. Introduction 2](#_Toc57676266)

[1.1 Purpose 2](#_Toc57676267)

[1.2 Objectives 2](#_Toc57676268)

[2. Functional Scope 2](#_Toc57676269)

[3. Overall Strategy and Approach 2](#_Toc57676270)

[3.1 Testing Strategy 2](#_Toc57676271)

[3.2 System Testing Entrance Criteria 2](#_Toc57676272)

[3.3 Testing Types 2](#_Toc57676273)

[3.4 Suspension Criteria and Resumption Requirements 3](#_Toc57676274)

[4. Execution Plan 3](#_Toc57676275)

[4.1 Execution Plan 3](#_Toc57676276)

[5. Traceability Matrix & Defect Tracking 6](#_Toc57676277)

[5.1 Traceability Matrix 6](#_Toc57676278)

[5.2 Defect Severity Definitions 7](#_Toc57676279)

[6. Environment 7](#_Toc57676280)

[6.1 Environment 7](#_Toc57676281)

[7. Assumptions 7](#_Toc57676282)

[8. Risks and Contingencies 8](#_Toc57676283)

[9. Appendices 8](#_Toc57676284)

# Introduction

## Purpose

This document is a test plan for Helping Hand System Testing, produced by the Mental Health Application team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release. This document is also purposed to help any further team or work performed on it to guide improvements and building of the software.

## Objectives

* Supports the intended goal of notifying a user a contact has dropped in rank and achieves the required standards.
* Satisfies the Entrance Criteria for User Acceptance Testing defined in section 3.2.

# Functional Scope

The Modules in the scope of testing for the Helping Hand System Testing are outlined in Section 3 of the Mental Health Awareness Application SRS document.

# Overall Strategy and Approach

## Testing Strategy

Helping Hand System Testing will include testing of all functionalities that are in the scope (Refer Functional Scope Section) identified. System testing activities will include the testing of new functionalities, screen level validations, workflows, functionality access, testing of internal & external interfaces.

## System Testing Entrance Criteria

To start system testing, certain requirement must be met for testing readiness. The readiness can be classified into:

* 1. Reading from a JSON file.
  2. Writing to a comma separated value (CSV) file.
  3. Availability to contact information and social media from user’s cellular device.
  4. User’s social media JSON files downloaded from a social media network.
  5. Basic algorithm implementation.
  6. User interface to capture calculations.

## Testing Types

### Usability Testing

### The Usability Testing will cover the user interface attributes present in the Helping Hand mobile app. It will test both the accuracy and the usability of the application. The Usability Testing will ensure that the navigation through the app is clear, and the application will provide the user with an enjoyable and consistent experience.

### Functional Testing

Functional Testing will cover the how a user will move throughout the Helping Hand application. This section will ensure that the navigation through the application is easy to understand and that each part of the user interface is easy to understand.

## Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

### Suspension Criteria

Testing will be suspended if the incidents found will not allow further testing of the system/application under test. If testing is halted, and changes are made to the hardware, software, or database, it is up to the Testing Manager to determine whether the test plan will be re-executed, or part of the plan will be re-executed.

What would require suspension:

* 1. Algorithm cannot compute without proper information.
  2. Button failure
  3. Permissions notification failure

### Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been retested successfully.

|  |  |
| --- | --- |
| **Suspension Criteria** | **Resumption Requirement** |
| Algorithm cannot compute without proper information. | The required information was pulled successfully from contacts and saved properly to JSON file. |
| Button failure | The button is activated; it is now performing the intended function (either triggering the specified method or displaying its respective UI view). |
| Permissions notification failure | The permission notification is displayed. |

# Execution Plan

## Execution Plan

The execution plan below details all the test cases that must be performed in this logical order to ensure the project is meeting the requirements given in the SRS.

**4.1.1 Algorithm**

*4.1.1.1 Test Case 1.1*

After more than one message is sent, a communication experience survey will be created for the user that updates the contact's score. This survey will track the communication impact, positivity, and time.

Pass: Survey notification sent to user and response is recorded.  
Fail: Contact score is unchanged, or notification is never sent.

*4.1.1.2 Test Case 1.2*

When a contact is moved outside of their current circle via pseudo data or drastic drop in contact with someone, a notification will be sent to the user to remind them to communicate with that contact.

Pass: Notification sent to user indicating contact name.  
Fail: No notification sent, or incorrect contact name.

*4.1.1.3 Test Case 1.3*

A contact log will be read from a master contact list from a user's phone.

Pass: A contact log will appear for the user to see.  
Fail: A contact log will not appear for a user.

*4.1.1.4 Test Case 1.4*

Contact score and contact name is saved to JSON file named “Contact Score”.

Pass: CSV file named “Contact score” will be found in project folder.  
Fail: CSV file names “Contact Score” will not be found in project folder.

*4.1.1.5 Test Case 1.5*

Contact score can be calculated by information from JSON files.

Pass: CSV file named “Contact Score” will have recorded trust calculation.  
Fail: CSV will not have a new score recorded under “Contact Score”.

*4.1.1.6 Test Case 1.6*

Changes in a contact score will be reordered and saved to CSV file named “Contact Score”.

Pass: There will be a changed number in the CSV file “Contact Score”   
Fail: There will not be a changed number in the CSV file “Contact Score”

*4.1.1.7 Test Case 1.7*

Contact logs (Comments, Likes, Direct Messages (DM’s)) will be read from a JSON file.

Pass: Number of times a JSON file is read will be recorded.  
Fail: Number of times a JSON file was read is not recorded.

*4.1.1.8 Test Case 1.8*

Contact logs (Calls and texts) will be read from a CSV file.

Pass: Number of times a CSV file is read will be recorded.

Fail: Number of times a CSV is read is not recorded.

*4.1.1.9 Test Case 1.9*

See Circle button will change the view to circle view.

Pass: Screen will change from home page to “Algorithm” page.  
Fail: Screen will not change from home page.

*4.1.1.10 Test Case 1.10*

Circle View will display contact names.

Pass: Names from the user's contacts will display.   
Fail: There will be no names from the users' contacts that will display.

**4.1.2 Block Contact**

*4.1.2.1 Test Case 2.1*

A number is selected from the contacts list in the app. By clicking that number, that contact, along with all the data associated with that contact is taken out of the algorithm and not used in the Trust calculation.

Pass: When a number is selected from the block contact view, all data pertaining to it is removed from the JSON files.  
Fail: The number is selected from the block contact view, but all contact info and message logs associated with it are not removed from the JSON files.

*4.1.2.2 Test Case 2.2*

Contact names will be saved to a JSON file named “Contacts”.

Pass: Contact info is pulled from the user’s phone and saved to a JSON file named Contacts when the algorithm is run.  
Fail: Contact info on the user’s phone is not saved to a JSON file when the algorithm is run.

*4.1.2.3 Test Case 2.3*

Block Number button will change the view to contacts list view.

Pass: Screen will change from home screen to block number page.   
Fail: Screen will not change from home screen to block number page.

**4.1.3 Change weights**

*4.1.3.1 Test Case 3.1*

The communication weight values (Call, Text, Facetime) are adjustable within the change weight's view.

Pass: Weight values are set new values.  
Fail: Weight values are unchanged.

*4.1.3.2 Test Case 3.2*

Weight's button will change the view to weight view.

Pass: Screen will change from home screen to weights page.

Fail: Screen will not change from home screen to weights page.

*4.1.3.3 Test Case 3.3*

The sliders for the weights change shall have a range from 0 – 1, with a default value of 0.3.

Pass: Weights can be set values between 0 and 1.

Fail: Weight is set to value greater than 1 or less than 0.

*4.1.3.4 Test Case 3.4*

When the weight sliders are all set to 0, the algorithm will still run and no users are displayed in the circles.

Pass: Nobody displays in the circles when weights are set to 0.

Fail: Contact scores still have value calculated and display in the circles.

*4.1.3.5 Test Case 3.5*

When the weight sliders are all set to 1, the algorithm will still run, and all users will be considered equal. Only the frequency of communication will affect ranking.

Pass: Users in circles will only be based on total frequency of communication in any form.

Fail: Contact scores still show evidence of using weights.

**4.1.4 User Interface**

*4.1.4.1 Test Case 4.1*

When the application is launched for the very first time, a pop-up notification is displayed requesting permission to access iOS contacts that allows the contacts to be displayed.

Pass: Notification requesting contact access is displayed.  
Fail: Notification does not display.

*4.1.4.2 Test Case 4.2*

The “Show Circle” button is clicked. When that button is activated, the application displays a new view where the top contacts (up to 5) are displayed on the screen. The top 5 contacts are found with the algorithm.

Pass: The top 1 to 5 contacts are displayed on the screen.  
Fail: The are no contacts or more than 5 contacts displayed

*4.1.4.3 Test Case 4.3*

When the app is opened, the app fires up straight to a home screen which displays the app logo and the three main navigation buttons: See Circle, Block Contact, and Weights

Pass: The app logo and buttons for See Circle, Block Contact, and Weights are all present.

Fail: The app logo or any of the buttons do not appear.

*4.1.4.4 Test Case 4.4*

The iOS service permissions utilized by the app can be accessed and set in the iOS Settings app. Revoking access to one of the permissions from that menu will cause the app to request permission for data access again when the app is launched.

Pass: When the user launches the app with a permission denied, the app will request permission via a pop-up when the user attempts to use a feature of the app that requires permissions to be allowed.  
Fail: When the user launches the app with a permission denied, the app will not request permission via a pop-up when the user attempts to use a feature of the app that requires permissions to be allowed.

*4.1.4.5 Test Case 4.5*

A notification will display asking for permission to access user contacts.

Pass: A notification is displayed asking for permission to access user contacts.  
Fail: A notification asking for permission to access user contacts does not get displayed.

*4.1.4.6 Test Case 4.6*

The font for the circle shall be Default with a font size of 33.

Pass: The font is Default and the size of 33  
Fail: The font is not Default and is not size 33

*4.1.4.7 Test Case 4.7*

The labels for the circle shall be centered on the X-axis and each have a 30-point difference on the Y-axis, starting from the Y-center.

Pass: The labels are centered on the X-axis and 30 points apart on the Y-axis, starting from the Y-center.  
Fail: The labels are not centered on the X-axis and not 30 points apart on the Y-axis.

*4.1.4.8 Test Case 4.8*

Buttons for the home screen shall be centered on the X-axis and have a 45-point difference on the Y-axis, starting 110 points from the bottom.

Pass: The buttons are centered on the X-axis and have a 45-point difference on the Y-axis, starting 110 points from the bottom.  
Fail: The buttons are not centered on the X-axis and do not have a 45-point difference on the Y-axis.

*4.1.4.9 Test Case 4.9*

When the “Show Circles” button on the home screen is pressed the ShowCircle View is displayed.

Pass: The “Show Circles” button takes the user to the ShowCircle View

Fail: The button does not take the user to the ShowCircle View

*4.1.4.10 Test Case 4.10*

When the “Block Contact” button on the home screen is pressed the BlockContact View is displayed.

Pass: The “Block Contact” button takes the user to the BlockContact View

Fail: The button does not take the user to the BlockContact View

*4.1.4.11 Test Case 4.11*

When the “Weights” button on the home screen is pressed the Weights View is displayed.

Pass: The “Weights” button takes the user to the Weights View

Fail: The button does not take the user to the Weights View

# Traceability Matrix & Defect Tracking

## Traceability Matrix

List of requirements[[1]](#footnote-1) and the corresponding test cases

|  |  |
| --- | --- |
| **Requirement** | **Test Case** |
| 3.2.4, 3.2.5, 3.2.6, 3.2.7 | 1.1 |
| 3.2.3 | 1.2 |
| 3.2.6, 3.2.7 | 1.3 |
| 3.2.6, 3.2.7 | 1.4 |
| 3.2.6, 3.2.7, 3.2.10 | 1.5 |
| 3.2.6, 3.2.7 | 1.6 |
| 3.2.6, 3.2.7 | 1.7 |
| 3.2.6, 3.2.7 | 1.8 |
| 3.2.3 | 1.9 |
| 3.2.3 | 1.10 |
| 3.1.1 | 2.1 |
| 3.2.6, 3.2.7 | 2.2 |
| 3.2.9 | 2.3 |
| 3.2.8 | 3.1 |
| 3.2.8 | 3.2 |
| 3.2.8 | 3.3 |
| 3.2.8 | 3.4 |
| 3.2.8 | 3.5 |
| 3.2.11 | 4.1 |
| 3.2.1 | 4.2 |
| 3.2.2 | 4.3 |
| 3.8.3, 3.9.1 | 4.4 |
| 3.2.9 | 4.5 |
| 3.2.1 | 4.6 |
| 3.2.1 | 4.7 |
| 3.2.2 | 4.8 |
| 3.2.13 | 4.9 |
| 3.2.14 | 4.10 |
| 3.2.15 | 4.11 |

## Defect Severity Definitions

|  |  |
| --- | --- |
| **Critical** | The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect.   * 1.3 * 4.3 |
| **Medium** | The defect does not seriously impair system function can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows:   * 1.1 * 1.2 * 1.4 * 1.6 * 1.8 * 1.9 * 1.10 * 2.1 * 2.2 * 2.3 * 4.9 * 4.10 * 4.11 |
| **Low** | The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows:   * 1.5 * 1.7 * 3.1 * 3.2 * 3.3 * 3.4 * 3.5 * 4.1 * 4.2 * 4.4 * 4.5 * 4.6 * 4.7 * 4.8 |

# 

# Environment

## Environment

* The System Testing Environment will be the Mac application XCode.
  + Requirement 3.3.1[[2]](#footnote-2)

# Assumptions

* Tester has access to a MacOS Catalina 15.6 or above
* Tester has contacts and data regarding messages/FaceTime/call history on their phone.
* Tester has access to iTunes backup of their phone.

# Risks and Contingencies

Potential testing risks are outlined in Section 3.4. It is possible while testing the data pulling aspects of the code that something could cease to function correctly. As a contingency, sample JSON files will be used in the meantime to mimic pulling data from the device itself.

# Appendices

[Mental Health Awareness Application\_SRS\_Final.docx](https://github.com/stoneyg/Mental-Health-Application-Team-1/blob/master/Deliverables/Mental%20Health%20Awareness%20Application_SRS_Final.docx)

1. Reference to SRS Final [↑](#footnote-ref-1)
2. Reference to SRS Final [↑](#footnote-ref-2)