**Final Project Report**

**INFSCI 2545 Software Quality Assurance**

**Instructor- Prof. Richard Lomoty**

**School of Computing & Information, University of Pittsburgh**

**Test of imhere Mobile app**

**Team Members:**

**Yixiao Gao**

**Huiyin Liu**

**Ben Ying**

**Bao Lei**

**Table of Contents**

SECTION 1: Introduction

SECTION 2: Test Scope

SECTION 3: Test Strategy

SECTION 4: Test Walkthrough

SECTION 5: Defects List

SECTION 6: Suggestions

APPENDIX I

**Section 1 Introduction**

iMHere is designed to connect patient and clinician through care-plan, and simultaneously keep them in continuous contact.iMHere forms a continuous, secure, Internet-based communication bridge, between patient and clinician platforms, that spans any distance. On the patient side, there are disease state specific smartphone apps, and on the clinician side there is a web-based portal. Patients receive new instructions moments after the clinician enters them, and patient responses are immediately available for clinician review. Patients may report problems to their clinician, and the clinician can immediately make adjustments.

**1.1 Purpose:**

In order to improve the efficacy and accuracy of testing, this document proposes an outline of the whole test plan, which is trying to make sure the software could be released or not. For the executable functional requirements test, we did a detailed work through, and list the defects we found so far.

**1.2 Background:**

As a medical software, iMHere seeks to overcome these limitations by providing clinicians with a system to remotely deliver and efficiently monitor customized self-care plans to patients, and providing patients with reminders to follow those plans. We are going to pay more attention for the reliability test and how the software protects the user privacy.

**Section 2 Test Scope**

**Functional Requirements**

|  |  |  |
| --- | --- | --- |
| Main Function | Sub menu | Unit Function |
| Dashboard | Coins | Using Coins to represent progress the user made. |
| Schedules | Includes all the schedule that the user need to follow |
| MyMeds | It includes all the medications the user used. |
| TeleCath | User can add schedule to remind his meeting with remote Doctor. |
| BMQs | User can use this function to set reminder for Bone marrow suppression problems. |
| Mood | User using this function to report mental issue. |
| Skincare | User using this to report wound or set schedule. |
| Education | N/A |
| PHR | User using this function to collect, track and share past and current information about your health. |
| Goals | User using this function to set health goal, challenges, archives. |
| Wheelchair | Manual handbook of Wheelchair. |
| Exercise | User using this function to track exercise status. |
| Nutrition | User using this function to track nutrition intake. |
| Supplies | This function includes the medication supplier’s information. |
| Messages |  | User using this function to send message to his therapist. |
| Contacts |  | User can add contact information to make quick contact with others. |
| Settings |  | The main setting of the application. |
| Connected |  | This function prompt user if there is now network connection. |

**Section 3 Test Strategy**

This chapter demonstrate all the test we plan to do and briefly introduce different test contents.

**3.1 Function test**

1. The functional test is mainly based on the written functional test cases to traverse the software functions.

2. The tests involved mainly include basic dashboard medical function, as well as the installation, uninstallation, running tests, and Test exception handling (including the sudden disconnection of the network or slow operation of the network, insufficient processing of the machine memory, etc.).

**3.2 Interrupt test**

1. The software should still be able to run normally after receiving calls, receiving SMS, locking screen, alarm rings, etc.

2. It should be still be able to run normally when the system switches from background to foreground.

**3.3 Compatibility test**

1. Hardware adaptation: different mobile phone, hardware performance and adaptation of different screen sizes

2. OS version compatibility: IOS 9.0 or later, compatible with iPhone, iPad, and iPod touch. Requires Android 6.0 or later. If some new APIs are used in older systems, it probably causes crash.

3. Adaptation of different resolution screens. The resolution of mobile devices is various. If the app is not properly processed, it may not display well or even affect the operation of the function

4. The compatibility test could be performed on the Android simulator and we can select several typical IOS machines.

**3.4 Property test**

1. The client performance test focuses on installation and uninstallation time, startup time, page load time, and the CPU, memory, power consumption that occupied by the main functions.

2. As for the performance of the server, focusing on response time, information concurrency, throughput and transaction rate, etc.

**3.5 Stability test**

1. Use random event stream to simulate human operation, which is helpful to check the memory overflow and null pointer problem.

2. Mainly detect the problem such as ANR (application not responding) and crash

**3.6 Security test**

1. Verify the user information protect mechanism.

**3.7 User acceptance test**

1. Select typical representative users to use it practically and report defects.

**Section 4 Test walkthrough**

Based on the test strategies, considering some of these need the support from developers, so we chose the requirements that could be tested on our devices, which involved the function test, interrupt test, compatibility test, and property test.

**4.1 Functional Requirements Test Walkthrough**

This test plan walkthrough works for all the inputs which are classified by the following input requirements: Dashboard, Model-Input-Letter, Model-Input-PhoneNumber, Model-Input-Zipcode, Model-Input-String, Model-Input-IntergerNumber, Model-Input-FloatingNumber. Those input requirements are classified in the Appendix I.

**Chapter 1: REQUIREMENTS**

**Dashboard:**

REQ-DASHBOARD: Upon open this app-iMHere, this system should display welcome interface. There is no input parameter in the Dashboard interface. But should accept click action and response to the action. “Dashboard” at the top of the interface, TIMELINE icon at the right top of the interface and SETTING icon at the left top of the interface. User profile photo and name should display at the interface. Below the profile section, there are REWARD and SCHEDULE sections list at the interface. Twelve sub-models are list at the interface with model’s names and their icon: MyMeds, TaleCath, BMQs, Mood, Skincare, Education, PHR, Goals, Wheelchair, Exercise, Nutrition, Supplies.

REQ-DASHBOARD-TIMELINE: Upon click the TIMELINE icon, the iMHere system should display Timeline of the user.

REQ-DASHBOARD-SETTING: Upon click the SETTING icon, the iMHere system should display the version of this app, “iMHere 2.2” for example. And shows five icons with the name of the sub-setting items: Dashboard, Messages, Contacts, Settings, Connected.

REQ-DASHBOARD-PROFILE: Upon click the Profile icon, the iMHere system should display “My Profile” activity interface.

REQ-DASHBOARD-REWARD: The coins that the users earned is displayed at the REWARD section. Upon click the REWARD section, the iMHere system should display “Reward” activity interface.

REQ-DASHBOARD-SCHEDULE: Upon click the SCHEDULE section, the iMHere system should display “Schedule” activity interface.

REQ-DASHBOARD-MYMEDS: Upon click the MYMEDS section, the iMHere system should display “MyMeds” activity interface.

REQ-DASHBOARD-TALECATH: Upon click the TALECATH section, the iMHere system should display “TaleCath” activity interface.

REQ-DASHBOARD-BMQS: Upon click the BMQS section, the iMHere system should display “BMQs” activity interface.

REQ-DASHBOARD-MOOD: Upon click the MOOD section, the iMHere system should display “Mood” activity interface.

REQ-DASHBOARD-SKINCARE: Upon click the SKINCARE section, the iMHere system should display “Skincare” activity interface.

REQ-DASHBOARD-EDUCATION: Upon click the EDUCATION section, the iMHere system should display “Education” activity interface.

REQ-DASHBOARD-PHR: Upon click the PHR section, the iMHere system should display “PHR” activity interface.

REQ-DASHBOARD-GOALS: Upon click the GOALS section, the iMHere system should display “Goals” activity interface.

REQ-DASHBOARD-WHEELCHAIR: Upon click the WHEELCHAIR section, the iMHere system should display “Wheelchair” activity interface.

REQ-DASHBOARD-EXERCISE: Upon click the EXERCISE section, the iMHere system should display “Exercise” activity interface.

REQ-DASHBOARD-NUTRITION: Upon click the NUTRITION section, the iMHere system should display “Nutrition” activity interface.

REQ-DASHBOARD-SUPPLIES: Upon click the SUPPLIES section, the iMHere system should display “Supplies” activity interface.

REQ-DASHBOARD-NF: The system shall active and display and execute the appropriate activity sections.

**Model-Input-Letter:**

REQ-INPUT-LETTER-PARAMETER: The system shall accept the letters, which can only be letters value. If the parameter is not letters value, the system should not accept the click of the “Submit” and display a message “Please enter valid parameter”.

REQ-INPUT-LETTER-SETUP: Upon get into this activity interface, the system shall display appropriate message, such as “Prescriber”.

REQ-INPUT-LETTER-NF: The system shall display the appropriate message within two seconds of the program being executed.

**Model-Input-String**

REQ-INPUT-STRING-PARAMETER: The system shall accept the string as the description of the textbox.

REQ-INPUT-STRING-SETUP: Upon get into this activity interface, the system shall display appropriate message, such as “Street, Notes, Brand name”.

REQ-INPUT-STRING-NF: The system shall display the appropriate message within two seconds of the program being executed.

**Model-Input-PhoneNumber**

REQ-INPUT-PHONENUMBER-PARAMETER: The system shall accept the ten numbers string as the value of PhoneNumber. If the parameter is not ten numbers string value, the system should not accept the click of the “Submit” and display a message “Please enter valid parameter”.

REQ-INPUT-PHONENUMBER-SETUP: Upon get into this activity interface, the system shall display appropriate message, such as “PhoneNumber”.

REQ-INPUT-PHONENUMBER-NF: The system shall display the appropriate message within two seconds of the program being executed.

**Model-Input-Zipcode**

REQ-INPUT-ZIPCODE-PARAMETER: The system shall accept the five numbers string as the value of ZIPCODE. If the parameter is not five numbers string value, the system should not accept the click of the “Submit” and display a message “Please enter valid parameter”.

REQ-INPUT-ZIPCODE-SETUP: Upon get into this activity interface, the system shall display appropriate message, such as “Zipcode”.

REQ-INPUT-ZIPCODE-NF: The system shall display the appropriate message within two seconds of the program being executed.

**Model-Input-INTERGERNUMBER**

REQ-INPUT-INTERGERNUMBER-PARAMETER: The system shall accept the interger as the value of Age. If the parameter is not ten numbers string value, the system should not accept the click of the “Submit” and display a message “Please enter valid parameter”.

REQ-INPUT-INTERGERNUMBER-SETUP: Upon get into this activity interface, the system shall display appropriate message, such as “Oneset(age), When(age)”.

REQ-INPUT-PHONENUMBER-NF: The system shall display the appropriate message within two seconds of the program being executed.

**Model-Input-FLOATINGNUMBER**

REQ-INPUT-FLOATINGNUMBER-PARAMETER: The system shall accept the interger as the value of Age. If the parameter is not ten numbers string value, the system should not accept the click of the “Submit” and display a message “Please enter valid parameter”.

REQ-INPUT-FLOATINGNUMBER-SETUP: Upon get into this activity interface, the system shall display appropriate message, such as “Oneset(age), When(age)”.

REQ-INPUT-FLOATINGNUMBER-NF: The system shall display the appropriate message within two seconds of the program being executed.

**Click-Input**

REQ-INPUT-CLICK-PARAMETER: The system shall accept the click action from the device. Upon get into this activity interface, the system shall display appropriate message, such as some message or next appropriate activity interface.

REQ-INPUT-PHONENUMBER-NF: The system shall display the appropriate message within two seconds of the program being executed.

**Chapter 2: PLOTTING OUT THE TEST PLAN**

**Dashboard:**

Part1: Input

N/A

Part2: Output

1. REQ-DASHBOARD
2. REQ-DASHBOARD-SETTING
3. REQ-DASHBOARD-REWARD

Part3: Performance

1. REQ-DASHBOARD-NF

**Model-Input-Letter:**

Part 1: Input (Accepting the parameter)

1. REQ-INPUT-LETTER-PARAMETER

Part 2: Output (Displaying messages and results)

Sub 1: Setup Message

1. REQ-INPUT-LETTER-SETUP

Sub 2: Results

Part 3: Performance

1. REQ-INPUT-LETTER-NF

**Model-Input-String:**

Part 1: Input (Accepting the parameter)

(a) REQ-INPUT-STRING-PARAMETER

Part 2: Output (Displaying messages and results)

Sub 1: Setup Message

(a) REQ-INPUT-STRING-SETUP

Sub 2: Results

Part 3: Performance

(a) REQ-INPUT-STRING-NF

**Model-Input-PhoneNumber:**

Part 1: Input (Accepting the parameter)

(a) REQ-INPUT-PHONENUMBER-PARAMETER

Part 2: Output (Displaying messages and results)

Sub 1: Setup Message

(a) REQ-INPUT-PHONENUMBER-SETUP

Sub 2: Results

Part 3: Performance

(a) REQ-INPUT-PHONENUMBER-NF

**Model-Input-Zipcode:**

Part 1: Input (Accepting the parameter)

(a) REQ-INPUT-ZIPCODE-PARAMETER

Part 2: Output (Displaying messages and results)

Sub 1: Setup Message

(a) REQ-INPUT-ZIPCODE-SETUP

Sub 2: Results

Part 3: Performance

(a) REQ-INPUT-ZIPCODE-NF

**Model-Input-INTERGERNUMBER:**

Part 1: Input (Accepting the parameter)

(a) REQ-INPUT-INTERGERNUMBER-PARAMETER

Part 2: Output (Displaying messages and results)

Sub 1: Setup Message

(a) REQ-INPUT-INTERGERNUMBER-SETUP

Sub 2: Results

Part 3: Performance

(a) REQ-INPUT-INTERGERNUMBER-NF

**Model-Input-FLOATINGNUMBER:**

Part 1: Input (Accepting the parameter)

(a) REQ-INPUT-FLOATINGNUMBER-PARAMETER

Part 2: Output (Displaying messages and results)

Sub 1: Setup Message

(a) REQ-INPUT-FLOATINGNUMBER-SETUP

Sub 2: Results

Part 3: Performance

(a) REQ-INPUT-FLOATINGNUMBER-NF

**Click-Input:**

Part 1: Input (Accepting the parameter)

Part 2: Output (Displaying messages and results)

Part 3: Performance

(a) REQ-INPUT-ZIPCODE-NF

**Chapter 3: FILLING OUT TEST PLAN**

**Dashboard:**

Since there is no input parameter, the possible use cases is the section that the user clicks:

1. The user clicks TIMELINE icon
2. The user clicks REWARD icon
3. The user clicks SCHEDULE icon
4. The user clicks SETTING icon
5. The user clicks MYMEDS icon
6. The user clicks TALECATH icon
7. The user clicks BMQs icon
8. The user clicks MOOD icon
9. The user clicks SKINCARE icon
10. The user clicks EDUCATION icon
11. The user clicks PHR icon
12. The user clicks GOALS icon
13. The user clicks WHEELCHAIR icon
14. The user clicks EXERCISE icon
15. The user clicks NUTRITION icon
16. The user clicks SPPLIES icon

User Test Case:

IDENTIFIER: DASHBOARD-TEST-TIMELINE

TEST CASE: Run the program with clicking TIMELINE icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click TIMELINE icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the TIMELINE icon, the iMHere system should display Timeline of the user. The program does not execute any other activities.

IDENTIFIER: DASHBOARD-TEST-REWARD

TEST CASE: Run the program with clicking TIMELINE icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click REWARD icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the REWARD icon, the iMHere system should display Reward of the user. The program does not execute any other activities.

IDENTIFIER: DASHBOARD-TEST- SCHEDULE

TEST CASE: Run the program with clicking SCHEDULE icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click SCHEDULE icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the SCHEDULE icon, the iMHere system should display Schedule of the user. The program does not execute any other activities.

IDENTIFIER: DASHBOARD-TEST- SETTING

TEST CASE: Run the program with clicking SETTING icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click SETTING icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the SETTING icon, the iMHere system should display the version of this app, “iMHere 2.2” for example. And shows five icons with the name of the sub-setting items: Dashboard, Messages, Contacts, Settings, Connected.

IDENTIFIER: DASHBOARD-TEST- MyMeds

TEST CASE: Run the program with clicking MyMeds icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click MyMeds icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the MYMEDS section, the iMHere system should display “MyMeds” activity interface.

IDENTIFIER: DASHBOARD-TEST- TALECATH

TEST CASE: Run the program with clicking TALECATH icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click TALECATH icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the TALECATH section, the iMHere system should display “TaleCath” activity interface.

IDENTIFIER: DASHBOARD-TEST- BMQs

TEST CASE: Run the program with clicking BMQs icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click BMQs icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the BMQS section, the iMHere system should display “BMQs” activity interface.

IDENTIFIER: DASHBOARD-TEST- MOOD

TEST CASE: Run the program with clicking MOOD icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click MOOD icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the MOOD section, the iMHere system should display “Mood” activity interface.

IDENTIFIER: DASHBOARD-TEST- SKINCARE

TEST CASE: Run the program with clicking SKINCARE icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click SKINCARE icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the SKINCARE section, the iMHere system should display “Skincare” activity interface.

IDENTIFIER: DASHBOARD-TEST- EDUCATION

TEST CASE: Run the program with clicking EDUCATION icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click EDUCATION icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the EDUCATION section, the iMHere system should display “Education” activity interface.

IDENTIFIER: DASHBOARD-TEST- PHR

TEST CASE: Run the program with clicking PHR icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click PHR icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the PHR section, the iMHere system should display “PHR” activity interface.

IDENTIFIER: DASHBOARD-TEST- GOALS

TEST CASE: Run the program with clicking GOALS icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click GOALS icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the GOALS section, the iMHere system should display “Goals” activity interface.

IDENTIFIER: DASHBOARD-TEST- WHEELCHAIR

TEST CASE: Run the program with clicking WHEELCHAIR icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click WHEELCHAIR icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the WHEELCHAIR section, the iMHere system should display “Wheelchair” activity interface.

IDENTIFIER: DASHBOARD-TEST- EXERCISE

TEST CASE: Run the program with clicking EXERCISE icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click EXERCISE icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the EXERCISE section, the iMHere system should display “Exercise” activity interface.

IDENTIFIER: DASHBOARD-TEST- NUTRITION

TEST CASE: Run the program with clicking NUTRITION icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click NUTRITION icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the NUTRITION section, the iMHere system should display “Nutrition” activity interface.

IDENTIFIER: DASHBOARD-TEST- SPPLIES

TEST CASE: Run the program with clicking SPPLIES icon

PRECONDITIONS: None

INPUT VALUES: N/A

EXECUTION STEPS: Click SPPLIES icon

OUTPUT VALUES: N/A

POSTCONDITIONS: Upon click the SPPLIES section, the iMHere system should display “Supplies” activity interface.

**Model-Input-Letter:**

Starting with the inputs. The possible use cases:

i. The user enters one valid parameter

ii. The user enters no parameter

iii. The user enters number which is invalid

iv. The user enters symbol which is invalid

i. Valid Parameter Test

IDENTIFIER: VALID-INPUT-LETTER-PARAMETER

TEST CASE: Run the program with a valid parameter

PRECONDITIONS: None

INPUT VALUES: Susan R

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: With other parameters input correctly, the system should display the next appropriate activity interface. The program does not display “Please enter valid parameter”

OBSERVED BEHAVOIR: With other parameters input correctly, the system displays the next appropriate activity interface. The program does not display “Please enter valid parameter”.

ii. Invalid Parameter Test

IDENTIFIER: INVALID-INPUT-LETTER-NONE-PARAMETER

TEST CASE: Run the program without entering a parameter

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-LETTER-PARAMETER

TEST CASE: Run the program with enter the number

PRECONDITIONS: None

INPUT VALUES: 50

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-LETTER-PARAMETER

TEST CASE: Run the program with entering the symbol

PRECONDITIONS: None

INPUT VALUES: %

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

Testing Part 2: Startup Message

IDENTIFIER: INVALID-INPUT-LETTER-PARAMETER

TEST CASE: Run the program to check whether the startup message displays

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: the system shall display appropriate message, such as “Name ”.

POSTCONDITIONS: The program should be ready to accept a parameter.

**Model-Input-String:**

Starting with the inputs. The possible use cases:

i. The user enters a valid parameter

ii. The user enters no parameter

i. Valid Parameter Test

IDENTIFIER: VALID-INPUT-STRING-PARAMETER

TEST CASE: Run the program with a valid parameter

PRECONDITIONS: None

INPUT VALUES: 111 Fifth ave, APT123

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: With other parameters input correctly, the system should display the next appropriate activity interface.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

ii. Invalid Parameter Test

IDENTIFIER: INVALID-INPUT-STRING-NONE-PARAMETER

TEST CASE: Run the program with a valid parameter

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface, without displaying “Please enter valid parameter”.

**Model-Input-PhoneNumber:**

Starting with the inputs. The possible use cases:

i. The user enters a valid parameter

ii. The user enters no parameter

iii. The user enters less than ten or more than ten numbers string

iv. The user enters letters which is invalid

v. The user enters symbol which is invalid

i. Valid Parameter Test

IDENTIFIER: VALID-INPUT-PHONENUMBER-PARAMETER

TEST CASE: Run the program with a valid parameter

PRECONDITIONS: None

INPUT VALUES: 1234567890

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: With other parameters input correctly, the system should display the next appropriate activity interface.

OBSERVED BEHAVOIR: With other parameters input correctly, the system displays the next appropriate activity interface. The program does not display “Please enter valid parameter”.

ii. Invalid Parameter Test

IDENTIFIER: INVALID-INPUT-PHONENUMBER-NONE-PARAMETER

TEST CASE: Run the program without parameter input

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-PHONENUMBER-MORE-PARAMETER

TEST CASE: Run the program with less than ten or more than ten numbers parameter

PRECONDITIONS: None

INPUT VALUES: 112353

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-PHONENUMBER-LETTER-PARAMETER

TEST CASE: Run the program with letter which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: Susan R

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-PHONENUMBER-SYMBOL-PARAMETER

TEST CASE: Run the program with symbol which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: %%\*\*@

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

**Model-Input-Zipcode:**

Starting with the inputs. The possible use cases:

i. The user enters a valid parameter

ii. The user enters no parameter

iii. The user enters less than five or more than five numbers string

iv. The user enters letters which is invalid

v. The user enters symbol which is invalid

i. Valid Parameter Test

IDENTIFIER: VALID-INPUT-ZIPCODE-PARAMETER

TEST CASE: Run the program with a valid parameter

PRECONDITIONS: None

INPUT VALUES: 12345

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: With other parameters input correctly, the system should display the next appropriate activity interface.

OBSERVED BEHAVOIR: With other parameters input correctly, the system displays the next appropriate activity interface. The program does not display “Please enter valid parameter”.

ii. Invalid Parameter Test

IDENTIFIER: INVALID-INPUT-ZIPCODE-NONE-PARAMETER

TEST CASE: Run the program without parameter input

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-ZIPCODE-MORE-PARAMETER

TEST CASE: Run the program with less than five or more than five numbers parameter

PRECONDITIONS: None

INPUT VALUES: 12345678

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-ZIPCODE-LETTER-PARAMETER

TEST CASE: Run the program with letter which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: Susan R

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-ZIPCODE-SYMBOL-PARAMETER

TEST CASE: Run the program with symbol which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: %%\*\*@

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

**Model-Input-INTERGERNUMBER:**

Starting with the inputs. The possible use cases:

i. The user enters a valid parameter

ii. The user enters no parameter

iii. The user enters more than one parameters

iv. The user enters letters which is invalid

v. The user enters symbol which is invalid

i. Valid Parameter Test

IDENTIFIER: VALID-INPUT-INTERGERNUMBER-PARAMETER

TEST CASE: Run the program with a valid parameter

PRECONDITIONS: None

INPUT VALUES: 35

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: With other parameters input correctly, the system should display the next appropriate activity interface.

OBSERVED BEHAVOIR: With other parameters input correctly, the system displays the next appropriate activity interface. The program does not display “Please enter valid parameter”.

ii. Invalid Parameter Test

IDENTIFIER: INVALID-INPUT-INTERGERNUMBER-NONE-PARAMETER

TEST CASE: Run the program without parameter input

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-INTERGERNUMBER-MORE-PARAMETER

TEST CASE: Run the program with more than one parameters input

PRECONDITIONS: None

INPUT VALUES: 30 50

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-INTERGERNUMBER-LETTER-PARAMETER

TEST CASE: Run the program with letter which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: Susan R

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-INTERGERNUMBER-SYMBOL-PARAMETER

TEST CASE: Run the program with symbol which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: %%\*\*@

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

**Model-Input-FLOATINGNUMBER:**

Starting with the inputs. The possible use cases:

i. The user enters a valid parameter

ii. The user enters no parameter

iii. The user enters more than one parameters

iv. The user enters letters which is invalid

v. The user enters symbol which is invalid

i. Valid Parameter Test

IDENTIFIER: VALID-INPUT-FLOATINGNUMBER-PARAMETER

TEST CASE: Run the program with a valid parameter

PRECONDITIONS: None

INPUT VALUES: 3.5

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: With other parameters input correctly, the system should display the next appropriate activity interface.

OBSERVED BEHAVOIR: With other parameters input correctly, the system displays the next appropriate activity interface. The program does not display “Please enter valid parameter”.

ii. Invalid Parameter Test

IDENTIFIER: INVALID-INPUT-FLOATINGNUMBER-NONE-PARAMETER

TEST CASE: Run the program without parameter input

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-FLOATINGNUMBER-MORE-PARAMETER

TEST CASE: Run the program with more than one parameters input

PRECONDITIONS: None

INPUT VALUES: 3.0 3.5

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-FLOATINGNUMBER-LETTER-PARAMETER

TEST CASE: Run the program with letter which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: Susan R

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

IDENTIFIER: INVALID-INPUT-FLOATINGNUMBER-SYMBOL-PARAMETER

TEST CASE: Run the program with symbol which is invalid parameter

PRECONDITIONS: None

INPUT VALUES: %%\*\*@

EXECUTION STEPS: Click the “Submission” button

OUTPUT VALUES: N/A

POSTCONDITIONS: The program displays “Please enter valid parameter”.

OBSERVED BEHAVOIR: The system displays the next appropriate activity interface without displaying “Please enter valid parameter”.

**Click-Input:**

The user clicks the button of the app.

IDENTIFIER: VALID-INPUT-CLICK-PARAMETER

TEST CASE: Run the program with click the button

PRECONDITIONS: None

INPUT VALUES: None

EXECUTION STEPS: Run the program with click the button

OUTPUT VALUES: N/A

POSTCONDITIONS: With other parameters input correctly, the system should display the next appropriate activity interface.

**Chapter 4: DETERMINING FOCUS**

**Dashboard:**

Create your equivalence classes

1. click TIMELINE icon 🡪 “Timeline” activity interface
2. click REWARD icon 🡪 “Reward” activity interface
3. click SCHEDULE icon 🡪 “Schedule” activity interface
4. click SETTING icon 🡪 “Setting” activity interface
5. click MYMEDS icon 🡪 “MyMeds” activity interface
6. click TALECATH icon 🡪 “TaleCath” activity interface
7. click BMQs icon 🡪 “BMQs” activity interface
8. click MOOD icon 🡪 “Mood” activity interface
9. click SKINCARE icon 🡪 “Skincare” activity interface
10. click EDUCATION icon 🡪 “Education” activity interface
11. click PHR icon 🡪 “PHR” activity interface
12. click GOALS icon 🡪 “Goals” activity interface
13. click WHEELCHAIR icon 🡪 “Wheelchair” activity interface
14. click EXERCISE icon 🡪 “Exercise” activity interface
15. click NUTRITION icon 🡪 “Nutrition” activity interface
16. click SPPLIES icon 🡪 “Spplies” activity interface

Consider the explicit boundary values such as:

There are no boundary values.

**Model-Input-Letter:**

Create your equivalence classes:

1234567890

**Model-Input-String:**

Create your equivalence classes:

111 Fifth ave, APT123

**Model-Input-PhoneNumber:**

Create your equivalence classes:

1234567890

**Model-Input-Zipcode:**

Create your equivalence classes:

12345

**Model-Input-INTERGERNUMBER:**

Create your equivalence classes:

35

**Model-Input-FLOATINGNUMBER:**

Create your equivalence classes:

3.5

**Click-Input:**

No equivalence classes.

**4.2 Non-Functional Requirements Test Walkthrough**

**Chapter 1: REQUIREMENTS**

**Interrupt:** Interrupt-test take the the dashboard interface as the preconditions.

REQ-Interrupt-Phone-call: iMHere should display the dashboard interface after users finished a phone call.

REQ-Interrupt-SMS: iMHere should display the dashboard interface after users close the systems’ SMS notifications.

REQ-Interrupt-Lock: iMHere should display the dashboard interface after users unlock the system, wake it up again.

REQ-Interrupt-Alarm: iMHere should display the dashboard interface after a system alarm rings.

REQ-Interrupt-Charge: iMHere should display the dashboard interface when users start to charge or end of charging.

REQ-Interrupt-Notification: iMHere should display the dashboard interface after users close the notifications from other mobile applications.

REQ-Interrupt-Switch: iMHere should display the dashboard interface when users go back to the home screen.

**Compatibility**: Compatibility test take the IPhone 6s, Android as the test environment

REQ-Compatibility-OS-Android: iMHere can be download from the app store, installed correctly in the Android system, users can open the application without crash, and click exit key could return to the last interface.

REQ-Compatibility-OS-IOS: iMHere can be download from the app store, installed correctly in the IOS system, users can open the application without crash, and click home key could return to the home screen.

REQ-Compatibility-Resolution-720P: iMHere should display correctly without information lost(all the icons display the name of it. ) in the screen with 720P resolution.

REQ-Compatibility-Resolution-1080P: iMHere should display correctly without information lost(all the icons display the name of it. ) in the screen with 1080P resolution.

REQ-Compatibility-Resolution-2K: iMHere should display correctly without information lost(all the icons display the name of it. ) in the screen with 2K resolution.

**Property:** test take the IPhone 6s, Android, as the test environment

REQ-Property-Installation-time: Users could install iMhere after download all the content with in 30s.

REQ-Property-Uninstallation-time: Users could uninstall iMhere within 10s.

REQ-Property-Startup-time: The system could open the application and display the home screen - dashboard interface within 3s after click the iMHere icon.

REQ-Property-Response-time: After users operate, iMHere should have corresponding reaction for the operation within 1s, this requirement is applied to all the functional requirements.

**Chapter 2: PLOTTING OUT THE TEST PLAN**

**Part1: Interrupt test**

**Part2: Compatibility test**

**part3: Property test**

**Chapter 3: FILLING OUT TEST PLAN**

part1-1 IDENTIFIER: INTERRUPT-TEST-Phone-Call

TEST CASE: Give a phone call to the test device

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: answer the phone call and hang up

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere display the dashboard interface correctly without crash

part1-2 IDENTIFIER: INTERRUPT-TEST-SMS

TEST CASE: send a message to the test device

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: click the SMS notification and close it

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere display the dashboard interface correctly without crash

part1-3 IDENTIFIER: INTERRUPT-TEST-Lock

TEST CASE: lock test device screen

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: unlock test device

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere display the dashboard interface correctly without crash

part1-4 IDENTIFIER: INTERRUPT-TEST-Alarm

TEST CASE: set an alarm in the test device system

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: wait the alarm to ring then close it

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere display the dashboard interface correctly without crash

part1-5 IDENTIFIER: INTERRUPT-TEST-Phone-Charge

TEST CASE: link to the charger

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: end the charging

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere display the dashboard interface correctly without crash

part1-6 IDENTIFIER: INTERRUPT-TEST-Phone-Notification

TEST CASE: news application push a notification

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: click the notification then close it

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere display the dashboard interface correctly without crash

part2-1. IDENTIFIER: Compatibility-TEST-Phone-Android

TEST CASE: open the application in the Android 6.8 system

PRECONDITIONS: system home screen

INPUT VALUES: N/A

EXECUTION STEPS: click iMHere

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere open correctly without crash

part2-2. IDENTIFIER: Compatibility-TEST-Phone-IOS

TEST CASE: open the application in the IOS 12.1 system

PRECONDITIONS: system home screen

INPUT VALUES: N/A

EXECUTION STEPS: click iMHere

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere open correctly without crash

part2-3. IDENTIFIER: Compatibility-TEST-Resolution-720P

TEST CASE: open the application in the 720P screen

PRECONDITIONS: system home screen

INPUT VALUES: N/A

EXECUTION STEPS: click iMHere

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere open without crash, all the icons display correctly

part2-4. IDENTIFIER: Compatibility-TEST-Resolution-1080P

TEST CASE: open the application in the 1080P screen

PRECONDITIONS: system home screen

INPUT VALUES: N/A

EXECUTION STEPS: click iMHere

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere open without crash, all the icons display correctly

part2-5. IDENTIFIER: Compatibility-TEST-Resolution-2K

TEST CASE: open the application in the 2K screen

PRECONDITIONS: system home screen

INPUT VALUES: N/A

EXECUTION STEPS: click iMHere

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere open without crash, all the icons display correctly

part3-1 IDENTIFIER: Property-TEST-Installation

TEST CASE: download the application and install it

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: N/A

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere could be installed within 30s

part3-2 IDENTIFIER: Property-TEST-Uninstallation

TEST CASE: uninstall iMHere

PRECONDITIONS: Dashboard interface

INPUT VALUES: N/A

EXECUTION STEPS: N/A

OUTPUT VALUES: N/A

POSTCONDITIONS: iMHere could be installed within 10s

**Section 5 Defects list**

|  |  |
| --- | --- |
| Identifier | details |
| INVALID-INPUT-LETTER-NONE-PARAMETER | With other parameters input correctly, the system displays the next appropriate activity interface. The program does not display “Please enter valid parameter”. |
| INVALID-INPUT-LETTER-NUMBER-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-LETTER-SYMBOL-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-STRING-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-PHONENUMBER-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-PHONENUMBER-MORE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-PHONENUMBER-LETTER-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-PHONENUMBER-SYMBOL-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-ZIPCODE-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-ZIPCODE-MORE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-ZIPCODE-MORE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-INTERGERNUMBER-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-INTERGERNUMBER-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-INTERGERNUMBER-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-INTERGERNUMBER-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-FLOATINGNUMBER-NONE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-FLOATINGNUMBER-MORE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-FLOATINGNUMBER-MORE-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |
| INVALID-INPUT-FLOATINGNUMBER-SYMBOL-PARAMETER | The system displays the next appropriate activity interface without displaying “Please enter valid parameter”. |

**Chapter 6 Suggestions**

Overall, we provide some suggestions based on the defects we detected through the test process.

The app should display appropriate message when user input an invalid input.

The system should allow users to use the return key of their smartphones.

**APPENDIX I**

**String**

MyMeds - Find Medication – ‘Type med name’

MyMeds – Find Medication – Manual Entry – ‘Brand name’

MyMeds – Find Medication – Manual Entry – ‘Generic Name’

MyMeds – Find Medication – Manual Entry – ‘Dosage unit

MyMeds – Find Medication – Manual Entry – ‘Labeler’

TeleCath – Report Issue –‘Notes’

TeleCath– Add new schedule – ‘Notes’

BMQs – Report Issue – ‘Notes’

BMQs – Add New Schedule – ‘Notes’

Mood – Add new Schedule – ‘Notes’

Skincare – Wound report – ‘Description of location’

Skincare – Wound report – Notes’

Skincare – add New Schedule – ‘Notes’

PHR – medical history – ‘Diagnosis’

PHR- medical history – ‘Onset’

PHR -medical history – ‘notes’

PHR – surgical history – ‘procedure’

PHR – surgical history – ‘when’

PHR – surgical history – ‘Notes’

PHR- Medications – modify – ‘Strength’

PHR – Medications – modify – ‘notes’

PHR – Medications – modify – ‘notes’

PHR – Allergy – ‘Allergen’

PHR – PHR – Allergy – ‘onset’

PHR – Allergy – ‘type of reaction’

PHR – Allergy – ‘action plan’

PHR – Immunization – ‘vaccine’

PHR – Immunization – ‘when’

PHR – Immunization – ‘notes’

PHR – Family Medical History – ‘relationship’

PHR – Family Medical History – ‘Diagnosis’

PHR – Family Medical History – ‘Onset’

PHR – Family Medical History – ‘notes’

PHR – Tools & Devices – ‘tools/device’

PHR – Tools & Devices – ‘onset’

PHR – Tool & Devices – ‘Notes’

PHR – Social History – Edit – ‘Live with’

PHR – Social History – Edit – ‘Social work concern’

PHR – Notes – ‘notes’

Goal – ‘type your goal’

Goal – ‘SubGoal’

Supplies – Vendors – ‘Vendor name’

Supplies – Vendor – ‘Street’

Supplies – Vendor – ‘City’

Supplies – Vendor – ‘Email’

Supplies – Vendor – ‘Notes’

Supplies – add Supply – ‘Name’

Supplies – add Supply – ‘category’

Supplies – add Supply – ‘Brand’

Supplies – add Supply – ‘Notes’

**FLOATINGNUMBER:**

PHR – Medication- modify – ‘Dosage’

Supplies – Add Button – Vendor – Add New Schedule – ‘Quantity’

**INTEGERNUMBER:**

PHR – Medical History – Add – ‘’Onset(age)”

PHR – Surgical History – Add – ‘When(age)’

PHR – Allergy – ‘Onset(age)’

PHR – Immunization – ‘When(age)’

PHR – Family Medical History – ‘Onset(age)’

PHR – Tools & Device – ‘Onset(age)’

**ZIPCODENUMBER:**

Supplies – Add Button – Vendor – Add Button – ‘Zip Code’

**PHONENUMBER:**

Supplies – Add Button – Vendor – Add Button – ‘Phone Number’

**LETTER**

MyMed – Medication – ‘Prescriber’

**CLICK:**

MyMeds – ‘Add Button’: Upon click the Add Button, the icon should rotate and a ‘Find Medication’ page should pop up.

MyMeds – Add Button - ‘Manual Entry’: Upon click the Manual Entry, a new form page would display.

MyMed – Add Button – ‘CheckList’: Upon click any item in Checklist, system will display the information page of that medication.

MyMed – Add Button – CheckList – ‘Back’: Upon click the Back button, system will display the previous checklist page.

MyMed – Add Button – CheckList – ‘Add’: Upon click the Add button, a window will pop out with attention ‘Add \*\* to your list?’

MyMed– Add Button – CheckList – ‘Cancel’: Upon click the Back button, system will display the ‘My Meds’ page.

MyMed – Add Button - Manual Entry – ‘Dosage Unit’: Upon click the ‘Dosage Unit’ a check list will pop out.

MyMed – Add Button - Manual Entry – ‘Back’: Upon click the Back button, the system will display the find my Medication page.

MyMed – Add Button - Manual Eentry – ‘Add’: Upon click the Add button, a window will pop out with attention ‘Add \*\* to your list?’

MyMed – Add Button - Manual Entry – ‘Cancel’: Upon click the Cancel button, system will display the main page of My meds.

TeleCath – ‘Add Button’: Upon click the add button, the icon will rotate, and three new icon will pop out.

TeleCath – Add Button - ‘Report Issue’: Upon click the report issue button, the system will

display the ‘TeleCath Self Report’ page.

TeleCath – Add Button -‘Add New Schedule’: Upon click the Add New Schedule button, system will display a ‘schedule’ page.

TeleCath – Add Button – Report Issue – ‘Experience Problem’: Upon click any of the items in the Experienced problem, the item will be activated.

TeleCath – Add Button - Report Issue – ‘submit’: Upon click the ‘Submit’ button, an alert will pop out saying ‘Attention, are you sure to submit this report?’

TeleCath – Add Button – Report Issue – ‘Cancel’: Upon click the ‘Cancel’ button, the system will display the TeleCath main page.

TeleCath – Add Button - Add New Schedule – ‘Repeat’: Upon click the Repeat button, any item you choose will be activated.

TeleCath – Add Button - Add New Schedule – ‘Time’: Upon click the ‘Time’ button, A calendar will pop out.

TeleCath – Add Button - Add New Schedule – ‘End Date’: Upon click the ‘End Date’ button, A calendar will pop out.

TeleCath – Add Button - Add New Schedule – ‘SAVE’: Upon click the ‘SAVE’ button, the system will add the schedule to the TeleCath main page, and a pop out a prompt ‘A schedule was added’.

TeleCath – Add Button - Add New Schedule – ‘Cancel’:

TeleCath – Add Button - Add New Schedule – ‘Every hour’: Upon click the ‘Every Hour’ button, a check list will pop out.

TeleCath – Add Button - Add New Schedule – ‘Every Week’: Upon click the ‘Every week’ button, a check list will pop out.

BMQs – ‘Add Button’: Upon click the Add Button, the icon will rotate, and two new icon will show up (Report issue, Add New Schedule)

BMQs – Add Button – ‘Report Issue’: Upon Click the ‘Report Issue’, a ‘BMQs Report page will Display.

BMQs – Add Button – ‘Add New Schedule’: Upon click the ‘Add New Schedule’ button, a Schedule page will display

BMQs – Add Button – Report Issue – ‘Experience problem’: Upon click any item in the check list, the item will be activated.

BMQs – Add Button – Report Issue – ‘Submit’: Upon click the submit button, an attention

window will pop to ensure you want to add this report.

BMQs – Add Button – Report Issue – ‘Cancel’: Upon click the the Cancel button the system will display the BMQs’s main page.

BMQs – Add Button - Add New Schedule – ‘Repeat’: Upon click the Repeat button, any item you choose will be activated.

BMQs – Add Button - Add New Schedule – ‘Time’: Upon click the ‘Time’ button, A calendar will pop out.

BMQs – Add Button - Add New Schedule – ‘End Date’: Upon click the ‘Time’ button, A calendar will pop out.

BMQs – Add Button - Add New Schedule – ‘SAVE’: Upon click the ‘SAVE’ button, the system will add the schedule to the BMQs main page, and a pop out a prompt ‘A schedule was added’.

BMQs – Add Button - Add New Schedule – ‘Cancel’: Upon click the ‘Cancel’ button, the

system will display the BMQs Main Page.

BMQs – Add Button - Add New Schedule – ‘Every hour’: Upon click the ‘Every Hour’ button, a check list will pop out.

BMQs – Add Button - Add New Schedule – ‘Every Week’: Upon click the ‘Every Week’ button, a check list will pop out.

BMQs – Add Button - Add New Schedule – ‘Reset’: Upon click the Reset, the whole schedule will be reset(Except the start time)

Mood – ‘Add Button’: Upon Click the Add Button, the icon will rotate, and two new icon will show up(Report Mood, Add New Schedule)

Mood – Add Button – ‘Report Mood’: Upon click the Add Button, the system will display the Mood report page.

Mood – Add Button – ‘Add New Schedule’ : Upon click the Add New Schedule button, the

system will display the Schedule page.

Mood – Add Button – Report Mood – ‘Cancel’: Upon click the Cancel button, the system will display the Mood Schedule’s main page.

Mood – Add Button - Add New Schedule – ‘Time’: Upon click the Time, a calendar will pop out.

Mood – Add Button - Add New Schedule – ‘Start Date’: Upon click the Start date, a Calendar will pop out.

Mood – Add Button - Add New Schedule – ‘End Date’: Upon click the End Date, a calendar will pop out.

Mood – Add Button - Add New Schedule – ‘SAVE’: Upon click the ‘SAVE’ button, the system will add the schedule to the Mood’s main page, and a pop out a prompt ‘A schedule was added’.

Mood – Add Button - Add New Schedule – ‘On’: Upon click any item in the On check list, the item will be activated.

Mood – Add Button - Add New Schedule – ‘Cancel’: Upon click on the Cancel button, the system will display the main page of the Mood Schedule page.

Mood – Add Button - Add New Schedule – ‘Every Week’: Upon click the Every Week, a check list will pop out.

SkinCare – ‘Add Button’: Upon click the Add Button, the icon will rotate, and two new icon will show up(Wound Reports, Add New Schedule).

SkinCare – Add Button – ‘Wound Reports’: Upon click the Wound Reports, the System will

display the wound report page.

SkinCare – Add Button – ‘Add New Schedule’: Upon click the Add New Schedule Button, the

system will display schedule page.

SkinCare – Add Button – Wound Reports – ‘Add Button’: Upon click the Wound report button, the system will display ‘New Wound Report’ page.

SkinCare – Add Button – Wound Reports – Add Button – ‘View Mode’: Upon click the view mode, the content will switch between graphic and form view.

SkinCare – Add Button – Wound Reports – Add Button – ‘Turn Over’: Upon click the Turn over button, the graphic will turn over to show the hidden part in the graphic.

SkinCare – Add Button – Wound Reports – Add Button – ‘Body parts’: Upon click any part of the graphic, the system will display a new page that can add detail information about it.

SkinCare – Add Button – Wound Reports – Add Button – Body Parts – ‘Take Picture’: Upon click on the ‘Take picture’ button, the system will using the system camera to take pictures.

SkinCare – Add Button – Wound Reports – Add Button – Body Parts – ‘Next’: Upon click the ‘Next’ button, system will display the next page till the last one.

SkinCare – Add Button – Wound Reports – Add Button – Body Parts – ‘Cancel’: Upon click the Cancel button, the system will display the ‘Wound report’ page.

SkinCare – Add Button - Add New Schedule – ‘Repeat’: Upon click any item of the repeat, the item will be activated.

SkinCare – Add Button - Add New Schedule – ‘Time’: Upon click the Time, a calendar will pop out.

SkinCare – Add Button - Add New Schedule – ‘End Date’: Upon click the End Date, a calendar will pop out.

SkinCare – Add Button - Add New Schedule – ‘SAVE’: Upon click the ‘SAVE’ button, the system will add the schedule to the Skincare Schedule’s main page, and a pop out a prompt ‘A schedule was added’.

SkinCare – Add Button - Add New Schedule – ‘Cancel’: Upon click on the Cancel button, the system will display the main page of the Skincare Schedule page.

SkinCare – Add Button - Add New Schedule – ‘Every hour’: Upon click the ‘Every Hour’ button, a check list will pop out.

SkinCare – Add Button - Add New Schedule – ‘Every Week’: Upon click the ‘Every Week’ button, a check list will pop out.

Education – ‘Retry’: Upon click the ‘Retry’ button, an alert window will show the information.

PHR – ‘Medical History’: Upon click the Medical History icon, the system will display its main page.

PHR – Medical History – ‘Remove All’: Upon click the ‘remove all’ button, the system will pop out a window with alert message to ensure you want to remove all the history.

PHR – Medical History – ‘Add’: Upon click on the Add button, the system will display a page providing a blank form.

PHR – Medical History – Add – ‘Diagnosis’: Upon click the Diagnosis, a check list will pop put, or you can type in it directly.

PHR – Medical History – Add – ‘Onset’: Upon click the Onset, it was activated, and you can type in value.

PHR – Medical History – Add – ‘SAVE’: Upon click the SAVE button, system will add the medical history to the medical history page, and prompt ‘Medical history added’

PHR – Medical History – Add – ‘Cancel’: Upon click the Cancel button, system will display the Medical History page.

PHR – ‘Surgical History’: Upon click the Surgical History icon, the system will display its main page.

PHR – Surgical History– ‘Remove All’: Upon click the ‘remove all’ button, the system will

pop out a window with alert message to ensure you want to remove all the history.

PHR – Surgical History – ‘Add’: Upon click on the Add button, the system will display a page

providing a blank form.

PHR – Surgical History – Add – ‘Procedure’: Upon click the Procedure, a check list will pop put, or you can type in it directly.

PHR – Surgical History – Add – ‘When’: it was activated once click on it, and you can type in value.

PHR – Surgical History – Add – ‘SAVE’: Upon click the SAVE button, system will add the Surgical history to the Surgical history page, and prompt ‘Surgical history added’

PHR – Surgical History – Add – ‘Cancel’: Upon click the Cancel button, system will display the Surgical History page.

PHR – ‘Medications’: Upon click the Medications icon, the system will display its main page.

PHR – Medications – ‘Modify’

PHR – Medications – Modify – ‘Strength’: Upon click on it, you can change its value.

PHR – Medications – Modify – ‘Dosage’: Upon click on it, you can change its value.

PHR – Medications – Modify – ‘SAVE’: Upon click on it, the system will prompt ‘Medication updated’

PHR – Medications – Modify – ‘Cancel’: Upon click on it, the system will display the Medications’ main page.

PHR – ‘Allergy’: Upon click the Allergy icon, the system will display its main page.

PHR – Allergy – ‘Remove All’: Upon click the ‘remove all’ button, the system will pop out a window with alert message to ensure you want to remove all the history.

PHR – Allergy – ‘Add’: Upon click on the Add button, the system will display a page providing a blank form.

PHR – Allergy – Add – ‘Life Threatening’: Upon click on it, the item you click will be activated.

PHR – Allergy – Add – ‘Allergen’: Upon click on it, it’s been activated, and you can change it’s value.

PHR – Allergy – Add – ‘Onset’: Upon click the Onset, it was activated, and you can type in value.

PHR – Allergy – Add - ‘Type of Reaction’: Upon click on it, it’s been activated, and you can change it’s value.

PHR – Allergy – Add – ‘Action Plan’: Upon click on it, it’s been activated, and you can change it’s value.

PHR – Allergy – Add – ‘SAVE’: Upon click on it, the system will add it to the Allergy page and prompt ‘Allergy Added.’

PHR – Allergy – Add – ‘Cancel’: Upon click on it, the system will display the allergy main page.

PHR – ‘Immunization History’: Upon click the Immunization History icon, the system will display its main page.

PHR – Immunization History – ‘Remove all’: Upon click the ‘remove all’ button, the system will pop out a window with alert message to ensure you want to remove all the history.

PHR – Immunization History – ‘Add’: Upon click on the Add button, the system will display a page providing a blank form.

PHR – Immunization History – Add - ‘Vaccine’: Upon click on it, it’s been activated, and you can change its value.

PHR – Immunization History – Add - ‘When’: Upon click When, it was activated, and you can type in value.

PHR – Immunization History – Add - ‘Notes’: Upon click on it, it’s been activated, and you can change its value.

PHR – Immunization History – Add - ‘SAVE’: Upon click on it, the system will add it to the Immunization page and prompt ‘Immune Added.’

PHR – Immunization History – Add - ‘Cancel’: Upon click on it, the system will display the Immunization main page.

PHR – ‘Family Medical History’: Upon click the Family Medical History icon, the

System will display its main page.

PHR – Family Medical History – ‘Remove All’: Upon click the ‘remove all’ button,

the system will pop out a window with alert message to ensure you want to remove all the history.

PHR – Family Medical History – ‘Add’: Upon click on the Add button, the system will

display a page providing a blank form.

PHR – Family Medical History – Add – ‘Relationship’: Upon click on it, it’s been activated, and you can change its value.

PHR – Family Medical History – Add – ‘Diagnosis’: Upon click on it, it’s been activated, and you can change its value.

PHR – Family Medical History – Add – ‘Onset’: Upon click the Onset, it was activated, and you can type in value.

PHR – Family Medical History – Add – ‘Living Status’: Upon click on the Family Medical History, any item clicked will be activated.

PHR – Family Medical History – Add – ‘Notes’: Upon click on it, it’s been activated,

and you can change its value.

PHR – Family Medical History – Add – ‘SAVE’: Upon click on it, the system will add it to the

Family Medical History page and prompt ‘FamilyHistroy History Added.’

PHR – Family Medical History – Add – ‘Cancel’: Upon click on it, the system will display the

Family Medical History main page.

PHR – ‘Tools & Devices’: Upon click the Tools & Devices icon, the system will display its main page.

PHR – Tools & Device – Add Button – ‘Remove All’: Upon click the ‘remove all’ button, the system will pop out a window with alert message to ensure you want to remove all the history.

PHR – Tools & Device – Add Button – ‘Add’: Upon click on the Add button,

the system will display a page providing a blank form.

PHR – Tools & Device – Add Button – Add – ‘Tool/Device’: Upon click on it,

it’s been activated, and you can change its value.

PHR – Tools & Device – Add Button – Add – ‘Active’: Any item you clicked will be activated.

PHR – Tools & Device – Add Button – Add – ‘SAVE’: Upon click on it, the

system will add it to the Tool/Device page and prompt ‘Tool Device Added.’

PHR – Tools & Device – Add Button – Add – ‘Cancel’: Upon click on it,

the system will display the Tools/Devices main page.

PHR – ‘Social History’: Upon click the Social History icon, the system will display its main page.

PHR – Social History – ‘Edit’: Upon click on the Add button, the system will display a

page providing a blank form.

PHR – Social History – Edit – ‘Housing Environment’: Upon click one it, it was activated, and you can change its value.

PHR – Social History – Edit – ‘Employment Status’: Upon click on it, Any item you clicked will be activated.

PHR – Social History – Edit – ‘Martial Status’: Upon click on it, Any item you clicked will be activated.

PHR – Social History – Edit – ‘Education’: Upon click on it, Any item you clicked will be activated.

PHR – Social History – Edit – ‘Active Student’: Upon click on it, Any item you clicked will be activated.

PHR – Social History – Edit – ‘Alcohol’: Upon click on it, any item you clicked will be activated.

PHR – Social History – Edit - ‘Tobacco’: Upon click on it, any item you clicked will be activated.

PHR – Social History – Edit – ‘Social Work Concern’: Upon click one it, it was activated, and you can change its value.

PHR – Social History – Edit – ‘SAVE’: Upon click on it, the system will add it to the

Social History and prompt ‘Social History Added.’

PHR – Social History – Edit – ‘Cancel’: Upon click on it, the system will display

the Social History main page.

PHR – ‘Notes’: Upon click the Notes icon, the system will display its main page.

PHR – Note – ‘Remove All’: Upon click the ‘remove all’ button, the system will pop out a window with alert message to ensure you want to remove all the history.

PHR – Note – ‘Add’: Upon click on the Add button, the system will display a page providing a blank form.

PHR – Note – Add – ‘Note’: Upon click on the Add button, the system will display

a page providing a blank form.

PHR – Note – Add – Note – ‘SAVE’: Upon click on the SAVE button, the system will add it to the Notes page without any prompt.

PHR – Note – Add – Note – ‘Cancel’: Upon click on the Cancel button, the system will display the Notes’ page.

Goal – ‘Goal’: Upon click on it, the Goal tab will display.

Goal – ‘Challenge’: Upon click on it, the Challenge tab will display.

Goal – ‘Archive’: Upon click on it, the Archive tab will display.

Goal – Goal - ‘Add Button’: Upon click on the Add button, the system will display

a page providing a blank form.

Goal – Goal – Add Button – ‘SubGoal’: Upon click on the SubGoal Button, a new textArea will display.

Goal – Goal – Add Button – ‘Delete’: Upon click on the delete button, the present

textArea will be deleted.

Goal – Goal – Add Button – ‘SAVE’: Upon click on the SAVE button, the system will add it

to the My Goal List page.

Goal – Goal – Add Button – ‘Cancel’: Upon click on the Cancel button, the system will display the My Goal List page.

Wheelchair – ‘Guide Book’: Upon click on the Guide book, the system will display the saved

content.

Exercise – ‘Add Button’: Upon click on the Add button, the system will display

a page of icons representing different exercises.

Exercise – Add Button – ‘Exercise icon’: Any one or multiply icons that are clicked will be activated.

Exercise – Add Button – ‘SAVE’: Upon click on the SAVE button, all the activated icons

will added to the Exercise page.

Exercise – Add Button – ‘Cancel’: Upon click on the Cancel button, the system will display the Exercise page.

Exercise – Add Button – ‘add or reduce’: Upon click on the add or reduce, the number in its icon will add in 5 steps.

Exercise – ‘Review’: Upon click on it, the system will display the graph summary page.

Nutrition – ‘Review’: Upon click on it, the system will display the graph summary page.

Nutrition – List One – ‘Check Boxes’: Upon click on the check box, in proper value, a

green check will display, and red check will display if over the proper value.

Nutrition – List One – ‘icon info’: Upon click on the icon info button, a window with

present information will pop out.

Supplies – ‘Add Button’: Upon click on the Add Button, the icon will rotate, and two new icon will show up(Vendors, Add Supply)

Supplies – Add Button – ‘Vendors’: Upon click on it, system will display the Vendor page.

Supplies – Add Button – Vendors – ‘Add Button’: Upon click on the Add button,

the system will display a page providing a blank form.

Supplies – Add Button – Vendors – Add Button – ‘Name’: Upon click on it, it was activated and you can change its value.

Supplies – Add Button – Vendors – Add Button – ‘Address’: Upon click on it, it was activated and you can change its value.

Supplies – Add Button – Vendors – Add Button – ‘phone number’: Upon click on it, it was

activated and you can change its value.

Supplies – Add Button – Vendors – Add Button – ‘email address’: Upon click on it, it was

activated and you can change its value.

Supplies – Add Button – Vendors – Add Button – ‘Notes’: Upon click on it, it was

activated and you can change its value.

Supplies – Add Button – Vendors – Add Button – ‘SAVE’: Upon click on it, system will add it to the Vendors’ page and prompted ‘a new vendor was added’.

Supplies – Add Button – Vendors – Add Button – ‘Cancel’: Upon click on it, system will add it to the Vendors’ page.

Supplies – Add Button – ‘Add Supply’: Upon click on the Add button, the system will display a page providing a blank form.

Supplies – Add Button – Add Supply - ‘Product’: Upon click on it, it was activated, and you can change its value.

Supplies – Add Button– Add Supply – ‘Detail’: Upon click on it, system will display a Select Properties page.

Supplies – Add Button– Add Supply – Detail – ‘Select Properties’: Upon click on it, any item you clicked will be activated.

Supplies – Add Button – Add Supply – Detail – Select Properties – ‘Cancel’: Once click on it, system will display Supply page. without any action.

Supplies – Add Button – Add Supply – Detail – Select Properties – ‘OK’: Upon click on it, any item has been activated will be add to the Supply page.

Supplies – Add Button – Add Supply – ‘Quantity’: Upon click on it, a number pad will pop up, and you can input its value.

Supplies – Add Button– Add Supply – Maintenance – ‘Last Ordered On’: Once click on it, a calendar will pop up and you can set its value.

Supplies – Add Button– Add Supply – Maintenance – ‘Last Refilled On’: Once click on it, a calendar will pop up and you can set its value.

Supplies – Add Button– Add Supply – Maintenance – ‘Last Updated On’: Once click on it, a calendar will pop up and you can set its value.

Supplies – Add Button– Add Supply – Maintenance – ‘Due on’: Once click on it, a calendar will pop up and you can set its value.

Supplies – Add Button– Add Supply – ‘Reminder at’: Once click on it, a calendar will pop up and you can set its value.

Supplies – Add Button – Add Supply – Reminder – ‘Reset’: Once click on it, any value set in the Reminder at will be reset.

Supplies – Add Button– Add Supply – ‘Vendor’: Once click on it, system will display a vendor list that you can choose to activate.

Supplies – Add Button– Add Supply – ‘SAVE’: Once click on it, system will add present information to the Supplies page and prompt ‘A new supply was added’.

Supplies – Add Button– Add Supply – ‘Cancel’: Upon click on it, system will display the Supplies page without action.

MyMed – Medication – Modify - ‘Add Schedule’

TeleCath – Add Button – Add New Schedule - Daily – ‘Time’

TeleCath – Add Button – Add New Schedule - Daily – ‘Start Date’

TeleCath – Add Button – Add New Schedule - Daily – ‘End Date’

TeleCath – Add Button – Add New Schedule – Hourly – ‘Start-End Time’

TeleCath – Add Button – Add New Schedule – Hourly – ‘Start Date’

BMQs – Add Button – Add New Schedule – Daily – ‘Time’

BMQs – Add Button – Add New Schedule – Daily – ‘Start Date

BMQs – Add Button – Add New Schedule – Daily – ‘End Date’

BMQs – Add Button – Add New Schedule – Hourly – ‘Start-End Time’

BMQs – Add Button – Add New Schedule – Hourly – ‘End Date’

Mood – Add Button – Add New Schedule – ‘Time’

Mood – Add Button – Add New Schedule – ‘Start Date

Mood – Add Button – Add New Schedule – ‘End Date’

Skincare – Add Button – Add New Schedule - Daily – ‘Time’

Skincare – Add Button – Add New Schedule - Daily – ‘Start Date’

Skincare – Add Button – Add New Schedule - Daily – ‘End Date’

Skincare – Add Button – Add New Schedule – Hourly – ‘Start-End Time’

Skincare – Add Button – Add New Schedule – Hourly – ‘Start Date’

Supplier – Add Button – Add Supply – Maintenance – ‘Last Ordered on’

Supplier – Add Button – Add Supply – Maintenance – ‘Last Refilled On’

Supplier – Add Button – Add Supply – Maintenance – ‘Last Update On’

Supplier – Add Button – Add Supply – Maintenance – ‘Due On’

Supplier – Add Button – Add Supply – Reminder – ‘Reminder at’