Health App

Test Case Specification

Version 1.3

December 3, 2017

Gayatri Patel

Melissa Heredia

Gopika Menon

Thanjila Uddin

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Authors** | **Comments** |
| 10/31/2017 | Version 1.0 | Gayatri Patel  Melissa Heredia  Gopika Menon  Thanjila Uddin | Initial Draft |
| 11/03/2017 | Version 1.1 | Gayatri Patel  Melissa Heredia  Gopika Menon  Thanjila Uddin | Second Draft |
| 11/04/2017 | Version 1.2 | Gayatri Patel  Melissa Heredia  Gopika Menon  Thanjila Uddin | Third Draft |
| 12/03/2017 | Version 1.3 | Gayatri Patel  Melissa Heredia  Gopika Menon  Thanjila Uddin | Final Draft |

# Table of Contents

[**Revision History**](#_ekeremplgwnf)1

[**Table of Contents**](#_w55cu8ae8jgv)2

[**Functional Testing**](#_dfynfp9xdlmg)3

[1.1 Offline Application](#_3wupotdlwpfy) 3

[1.1.1 User Registration](#_2lrt4pfndtxk) 3

[1.1.2 Medical Information](#_fusdjfrsilr9) 11

[1.1.3 Documents](#_h3qaudh01xqv) 24

[1.1.4 Reminders/Notifications](#_yqldhms4h8jh) 30

[1.1.5 Monthly Reminder](#_973bohw9vc8x) 35

[1.1.5 Print](#_kidlk8f6d25o) 37

[1.1.7 Menu](#_1wz7kx2pgw) 39

[1.1.8 User Logout](#_x4vt4m4y5fdl) 40

[1.2 Online Application](#_jwqeke3bmtxs) 41

[1.2.1 Server](#_emdcdv1g7xma) 41

[1.2.2 Share Data](#_t7kxxiwtod9v) 42

[1.2.3 Print](#_kwfzjl9ygwmp) 42

[**2. Non-Functional Testing**](#_93h9dhq2fwya)43

[2.1. Non-Functional Test Cases](#_ss06z8w63p4r) 43

[**2. Integration Testing**](#_akmcsua4ktt9)49

[2.1 Offline Application](#_1ph4coex4h3m) 49

[2.2 Online Application](#_ny4y1kbhix7) 50

[**3. System Testing**](#_5jhxj1foaxsn)51

[3.1 Offline Application](#_f7kbmpl2fiz0) 52

[3.2 Online Application](#_e3bt1yhfb4c) 56

[**Appendix**](#_x2edbai3r7pv)57

[Database Scripts](#_l6sdt4rmghfc) 57

[Requirements Traceability Matrix](#_phr6a77p7ckp) 62

[Key Terms](#_nfalunjcf3jh) 65

[References](#_qun0com7uzg) 66

# Functional Testing

## 1.1 Offline Application

### 1.1.1 User Registration

|  |  |
| --- | --- |
| **Test Case Name** | **User Registration - Successful** |
| **Test Case ID** | **TC - 01** |
| Priority | High |
| Preconditions | 1. User taps on App Store application on their iOS device (iPad or iPhone) and searches for ‘MyHealthKeeper’ 2. User taps on ‘Get’ then ‘Install’ button to download the application 3. User enters their Apple ID Password when prompted 4. User taps ‘Open’ button to open MyHealthKeeper 5. System executes DBS-00 6. System executes DBS-01 |
| Postconditions | 1. System executes DBS-02 |
| Test Steps | 1. User taps ‘Click to Register’ button 2. User enters ‘First Name’ in designated text field as ‘John’ 3. User enters ‘Last Name’ in designated text field as ‘Smith’ 4. User enters ‘Username’ containing the following format: 6-10 characters in length containing letters [A-Z,a-z] and optional integers [0-9] that does not already exist in the database in designated text field as ‘Smith1’ 5. User enters ‘Password’ of their choice in designated text field as ‘Smith1!’ 6. User performs one of the following:    1. User enters ‘Email’ containing the following format: letters, numbers and select special characters [A-Z,a-z,0-9,\_-.], an ‘@’ symbol, a ‘.’ symbol, and two or three letters [a-z] in designated text field as ‘john.smith@gmail.com’    2. User enters ‘Cellphone’ containing the following format: ten integers [0-9], without parentheses or dashes, in the designated text field as ‘’ 7. User taps ‘Next Page’ button |
| Expected Results | Smith1 is directed to the ‘Setup Security Questions’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **User Registration - Unsuccessful** |
| **Test Case ID** | **TC - 02** |
| Priority | High |
| Preconditions | 1. User taps on App Store application on their iOS device (iPad or iPhone) and searches for ‘MyHealthKeeper’ 2. User taps on ‘Get’ then ‘Install’ button to download the application 3. User enters their Apple ID Password when prompted 4. User taps ‘Open’ button to open MyHealthKeeper 5. System executes DBS-00 6. System executes DBS-01 |
| Postconditions | 1. User’s Registration information is not saved in the database’s ‘Registration’ table |
| Test Steps | 1. User taps ‘Click to Register’ button 2. User performs one of the following:    1. User enters ‘First Name’ in designated text field as ‘’    2. User enters ‘Last Name’ in designated text field as ‘’    3. User does not enter ‘Username’ according to the constraints in Step 4 of TC-1 in designated text field    4. User does not enter ‘Password’ according to the constraints in Step 5 of TC-1 in designated text field    5. User performs one of the following:       1. User does not enter ‘Email’ according to the constraints in Step 7 of TC-1 in designated text field       2. User does not enter ‘Cellphone’ according to the constraints in Step 8 of TC-1 in designated text field 3. User taps outside of the text field. |
| Expected Results | 1. If User performs Test Step 2a, User receives an ‘Error’ Alert on the iOS device screen stating “Please enter a value in the First Name field.” 2. If User performs Test Step 2b, User receives an ‘Error’ Alert on the iOS device screen stating “Please enter a value in the Last Name field.” 3. If User performs Test Step 2c, User receives one of the following ‘Error’ Alerts:    1. “Please enter a value in the Username field”    2. “Username already exists. Please enter a different username” 4. If User performs Test Step 2d, User receives an ‘Error’ Alert on the iOS device screen stating “Please enter a value in the Password field.” 5. If User performs Test Step 2e, User receives an ‘Error’ Alert on the iOS device screen stating “One or more fields may be empty. Please enter a value.” 6. User is not directed to the ‘Setup Security Questions’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Setup of Security Questions - Successful** |
| **Test Case ID** | **TC - 03** |
| Priority | Medium |
| Preconditions | 1. TC - 01 passes 2. System executes DBS-03 |
| Postconditions | 1. System executes DBS-04 |
| Test Steps | 1. Smith1 is directed to the ‘Setup Security Questions’ page 2. Smith1 selects first Security Question as “What was the name of your first pet?” from the picker. 3. Smith1 enters first Security Answer as “Leo” in designated text field 4. Smith1 selects second Security Question as “What company did you hold your first job?” from the picker. 5. Smith1 enters second Security Answer as “Starbucks” in designated text field 6. Smith1 selects third Security Question as “What was your first car?” from the picker. 7. Smith1 enters third Security Answer as “Honda Civic” in designated text field 8. Smith1 taps ‘Finish & Sign In’ button. |
| Expected Results | 1. Smith1 is directed to the Main page |

|  |  |
| --- | --- |
| **Test Case Name** | **Setup of Security Questions - Unsuccessful** |
| **Test Case ID** | **TC - 04** |
| Priority | Medium |
| Preconditions | 1. TC - 01 passes 2. System executes DBS-03 |
| Postconditions | 1. Smith1’s Security Questions and Answers are not saved in the database’s ‘Security’ table |
| Test Steps | 1. Smith1 performs one of the following:    1. Smith1 does not select first Security Question    2. Smith1 does not enter first Security Answer in designated text field    3. Smith1 does not select second Security Question    4. Smith1 does not enter second Security Answer in designated text field    5. Smith1 does not select third Security Question    6. Smith1 does not enter third Security Answer in designated text field 2. Smith1 taps ‘Finish & Sign In’ button |
| Expected Results | 1. If Smith1 performs Test Step 1a, User receives an ‘Error’ Alert on the iOS device screen stating “Please select a Security Question.” 2. If Smith1 performs Test Step 1b, User receives an ‘Error’ Alert on the iOS device screen stating “Please answer the Security Question.” 3. If Smith1 performs Test Step 1c, User receives an ‘Error’ Alert on the iOS device screen stating “Please select a Security Question.” 4. If Smith1 performs Test Step 1d, User receives an ‘Error’ Alert on the iOS device screen stating “Please answer the Security Question.” 5. If Smith1 performs Test Step 1e, User receives an ‘Error’ Alert on the iOS device screen stating “Please select a Security Question.” 6. If Smith1 performs Test Step 1f, User receives an ‘Error’ Alert on the iOS device screen stating “Please answer the Security Question.” 7. Smith1 remains on the Setup Security Questions page, and is not directed to the Main page |

|  |  |
| --- | --- |
| **Test Case Name** | **User Login - Successful** |
| **Test Case ID** | **TC - 05** |
| Priority | High |
| Preconditions | 1. TC - 01 and TC - 03 have passed 2. System executes DBS-01 3. System executes DBS-02 4. System executes DBS-03 5. System executes DBS-04 |
| Postconditions | 1. System executes DBS-05 2. Smith1’s username and password are compared and matched to the ‘username’ and ‘password’ columns of the database’s ‘Registration’ table |
| Test Steps | 1. Smith1 enters username as ‘Smith1’ according to the constraints in Step 4 of TC-1 in designated text field 2. Smith1 enters password as ‘Smith1!’ according to the constraints in Step 5 of TC-1 in designated text field 3. Smith1 taps ‘Sign In’ button. |
| Expected Results | 1. Smith1 is directed to the Main page |

|  |  |
| --- | --- |
| **Test Case Name** | **User Login - Unsuccessful** |
| **Test Case ID** | **TC - 06** |
| Priority | High |
| Preconditions | 1. TC - 01 and TC - 03 has passed 2. Execute DBS-01 3. Execute DBS-02 4. Execute DBS-03 5. Execute DBS-04 |
| Postconditions | 1. For Test Steps 1b, 1d, and 1f where Smith1 enters username as ‘Smith1’ in the designated text field, system executes DBS-05. 2. Smith1’s Login credentials are compared, but not matched in the database’s ‘Registration’ table |
| Test Steps | 1. Smith1 performs one of the following:    1. Smith1 leaves ‘Username’ text field empty    2. Smith1 leaves ‘Password’ text field empty    3. Smith1 does not enter ‘Username’ according to the constraints in Step 4 of TC-1 in designated text field    4. Smith1 does not enter ‘Password’ according to the constraints in Step 5 of TC-1 in designated text field    5. Smith1 enters ‘Username’ as ‘Name12’ which does not exist in the database’s ‘Registration’ table    6. Smith1 enters ‘Password’ as ‘Password123’ which does not exist or match to an existing username in the database’s ‘Registration’ table 2. Smith1 taps ‘Sign In’ button |
| Expected Results | 1. If Smith1 performs Test Step 1a and/or 1b, Smith1 receives an ‘Error’ Alert on the iOS device screen stating “Please enter a username and password” after tapping ‘Sign In’ button 2. If Smith1 performs Test Step 1c and/or 1d, Smith1 receives an ‘Error’ Alert on the iOS device screen stating “Username or password is not valid. If you are a first-time user, please Register.” after tapping ‘Sign In’ button 3. If Smith1 performs Test Step 1e and/or 1f, Smith1 receives an ‘Error’ Alert on the iOS device screen stating “Username or password is not valid. If you are a first-time user, please Register.” after tapping ‘Sign In’ button 4. Smith1 remains on the Sign In page, and is not directed to the Main page |

|  |  |
| --- | --- |
| **Test Case Name** | **Forgot Password - Successfully Answer Security Questions** |
| **Test Case ID** | **TC - 07** |
| Priority | Medium |
| Preconditions | 1. TC - 01 and TC - 03 have passed |
| Postconditions | 1. Smith1’s Security Answers are compared and matched to ‘pAnswer1’, ‘pAnswer2’, and ‘pAnswer3’ columns of the database’s ‘Security’ table |
| Test Steps | 1. Smith1 enters ‘Username’ according to the constraints in Step 4 of TC-1 in designated text field by entering ‘Smith1’ 2. Smith1 taps ‘Forgot Password’ button 3. Smith1’s username is compared and matched to ‘username’ column of the database’s ‘Registration’ table 4. Smith1’s username is used as a foreign key in the database’s ‘Security’ table to retrieve his Security Questions 5. Smith1 is directed to the ‘Answer Security Questions’ page 6. Smith1’s retrieved Security Questions are displayed on the ‘Answer Security Questions’ page 7. Smith1 enters first Security Answer in designated text field as ‘Leo’ 8. Smith1 enters second Security Answer in designated text field as ‘Starbucks’ 9. Smith1 enters third Security Answer in designated text field as ‘Honda Civic’ 10. Smith1 taps ‘Submit’ button |
| Expected Results | 1. Smith1 is directed to the ‘Reset Password’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Forgot Password - Unsuccessfully Retrieve Security Questions** |
| **Test Case ID** | **TC - 08** |
| Priority | Medium |
| Preconditions | 1. TC - 01 and TC - 03 have passed |
| Postconditions | Smith1 is not directed to the ‘Answer Security Questions’ page |
| Test Steps | 1. Smith1 performs one of the following:    1. Smith1 leaves ‘Username’ text field empty    2. Smith1 enters ‘Username’ that does not exist in the database’s ‘Registration’ table by entering ‘User12’ 2. Smith1 taps ‘Forgot Password’ button |
| Expected Results | 1. If Smith1 performs Test Step 1a or 1b, Smith1 receives an ‘Error’ Alert on the iOS device screen stating “Please enter a valid username.” after tapping ‘Forgot Password’ button 2. Smith1 remains on the Sign In page, and is not directed to the ‘Answer Security Questions’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Forgot Password - Unsuccessfully Answer Security Questions** |
| **Test Case ID** | **TC - 09** |
| Priority | Medium |
| Preconditions | 1. TC - 01 and TC - 03 have passed |
| Postconditions | 1. Smith1’s Security Answers are not matched in the database’s ‘Security’ table |
| Test Steps | 1. Smith1 enters ‘Username’ according to the constraints in Step 4 of TC-1 in designated text field by entering ‘Smith1’’ 2. Smith1 taps ‘Forgot Password’ button 3. Smith1’s username is compared and matched to ‘username’ column of the database’s ‘Registration’ table 4. Smith1’s username is used as a foreign key in the database’s ‘Security’ table to execute DBS-06 5. Smith1 is directed to the ‘Answer Security Questions’ page 6. Smith1’s retrieved Security Questions are displayed on the ‘Answer Security Questions’ page 7. Smith1 performs one or all of the following:    1. Smith1 does not enter first Security Answer or enters incorrect first Security Answer in designated text field    2. Smith1 does not enter second Security Answer or enters incorrect second Security Answer in designated text field    3. Smith1 does not enter third Security Answer or enters incorrect third Security Answer in designated text field 8. Smith1 taps outside the text field |
| Expected Results | 1. If Smith1 performs Test Step 7a, Smith1 receives an ‘Error’ Alert on iOS device screen stating “Incorrect Security Answers” after tapping ‘Submit’ button 2. If Smith1 performs Test Step 7b, Smith1 receives an ‘Error’ Alert on iOS device screen stating “Incorrect Security Answers” after tapping ‘Submit’ button 3. If Smith1 performs Test Step 7c, Smith1 receives an ‘Error’ Alert on iOS device screen stating “Incorrect Security Answers” after tapping ‘Submit’ button 4. Smith1 remains on the Answer Security Questions page, and is not directed to the ‘Reset Password’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Forgot Password - Successfully Reset Password** |
| **Test Case ID** | **TC - 10** |
| Priority | Medium |
| Preconditions | 1. TC - 01 and TC - 03 has passed 2. System executes DBS-06 3. Smith1 has successfully answered his Security Questions |
| Postconditions | 1. Smith1’s username is used as a foreign key in the database’s ‘Security’ table to retrieve the ‘password’ column 2. System executes DBS-07 |
| Test Steps | 1. Smith1 is directed to the ‘Reset Password’ page 2. Smith1 enters ‘New Password’ according to the constraints in Step 5 of TC-1 in designated text field by entering ‘Password12’ 3. Smith1 enters ‘Confirm New Password’ according to the constraints in Step 5 of TC-1 in designated text field by entering ‘Password12’ 4. Smith1 taps ‘Reset Password’ button |
| Expected Results | 1. Smith1 is directed to the ‘Sign In’ page 2. Smith1 can perform TC - 05 with the following login credentials:    1. Username: Smith1    2. Password: Password12 |

|  |  |
| --- | --- |
| **Test Case Name** | **Forgot Password - Unsuccessfully Reset Password** |
| **Test Case ID** | **TC - 11** |
| Priority | Medium |
| Preconditions | 1. TC - 01 and TC - 03 has passed 2. System executes DBS-06 3. Smith1 has successfully answered Security Questions |
| Postconditions | 1. Smith1’s ‘New Password’ is not updated in the ‘password’ column of the database’s ‘Security’ table |
| Test Steps | 1. Smith1 is directed to the ‘Reset Password’ page 2. Smith1 performs one of the following:    1. Smith1 leaves ‘New Password’ empty    2. Smith1 leaves ‘Confirm New Password’ empty    3. ‘New Password’ and ‘Confirm New Password’ do not match 3. Smith1 taps ‘Reset Password’ button |
| Expected Results | 1. If Smith 1 performs Test Step 2a and/or 2b, Smith1 receives an ‘Error’ Alert on iOS device screen stating “Field(s) cannot be empty. Please enter a value.” after tapping ‘Submit New Password’ button. 2. If Smith1 performs Test Step 2c, Smith1 receives an ‘Error’ Alert on iOS device screen stating “Passwords do not match. Please try again.” after tapping ‘Submit New Password’ button 3. Smith1 remains on the Reset Password page is not directed to the Sign In page |

### 1.1.2 Medical Information

|  |  |
| --- | --- |
| **Test Case Name** | **Enter Personal Information - Successful** |
| **Test Case ID** | **TC - 12** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Personal Information’ page. 3. System executes DBS-08 |
| Postconditions | 1. System executes DBS-09 |
| Test Steps | 1. Smith1 enters ‘Last Name’ in designated text field containing the following format: Between 3-30 characters in length containing letters [A-Z, a-z] as ‘Smith’ 2. Smith1 enters ‘First Name’ in designated text field containing the following format: Between 3-30 characters in length containing letters [A-Z, a-z] as ‘John’ 3. Smith1 enters ‘Date of Birth’ in designated text field in the following format: MM/DD/YYYY as ‘12/20/1998’ 4. Smith1 enters ‘Gender’ in designated text field containing the following format: characters containing letters [A-Z, a-z] as ‘Male’ 5. Smith1 enters ‘Street’ in designated text field as ‘123 Main’ 6. Smith1 enters ‘City’ in designated text field containing the following format: Between 2-60 characters in length containing letters [A-Z, a-z] as ‘Detroit’ 7. Smith1 selects ‘State’ from drop down list by scrolling and stopping on ‘Michigan’. 8. Smith1 enters ‘Zip code’ in designated text field that accepts up to 10 digits [0-9] as ‘48202’ 9. Smith1 taps ‘Save and Continue’ button |
| Expected Results | 1. Smith1 is directed to the ‘List of Doctor’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter List of Doctors** |
| **Test Case ID** | **TC - 13** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Doctors’ page 3. Execute DBS-10 |
| Postconditions | 1. Execute DBS-11 |
| Test Steps | 1. Smith1 enters ‘Name of Doctor’ in designated text field containing the following format: Between 3-30 characters in length containing letters [A-Z, a-z] and [.] as ‘Dr. Amrit’ 2. Smith1 taps on drop-down arrow next to ‘Name of Doctor’ text field    1. Additional text fields will be displayed below ‘Name of Doctor’ 3. Smith1 enters ‘Specialty’ in designated text field as ‘Pediatrician’ 4. Smith1 enters ‘Doctor Contact” in designated text field as ‘3131111111’ 5. Smith1 enters ‘Doctor Address’ in designated text view as ‘123 Baker, Detroit’ 6. Smith1 taps ‘Add’ button 7. Entered ‘Name’ will appear below Doctor Information text fields. 8. Doctor Information text fields will be emptied. 9. Smith1 taps on drop-down arrow next to ‘Name’ below Doctor Information text fields. 10. Entered Specialty, Contact, and Address will appear below Entered ‘Name’. 11. Smith1 taps “Save and Continue’. |
| Expected Results | 1. Smith1 is directed to the ‘List of Illnesses/Diseases’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter List of Illnesses/Diseases** |
| **Test Case ID** | **TC - 14** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Illnesses/Diseases’ page 3. System executes DBS-12 |
| Postconditions | 1. System executes DBS-13 |
| Test Steps | 1. Smith1 enters ‘Name of Illnesses/Diseases’ in designated text field containing up to 50 characters as ‘Heart Failure’ 2. Smith1 taps ‘Add’ button 3. Entered ‘Name of Illnesses/Disease’ appears below Illnesses/Disease text field. 4. ‘Name of Illnesses/Diseases’ text field will be emptied. 5. Smith1 taps “Save and Continue’. |
| Expected Results | 1. Smith1 is directed to the ‘List of Medications’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter List of Medications** |
| **Test Case ID** | **TC - 15** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Medications’ page 3. System executes DBS-14 |
| Postconditions | 1. System executes DBS-15 |
| Test Steps | 1. Smith1 enters ‘Name of Medication’ in designated text field containing up to 30 characters as ‘Amoxicilline’ 2. Smith1 enters ‘Dosage’ in the designated text field containing up to 30 characters as ‘500 mg’ 3. Smith1 enters ‘Status’ in the designated text field as ‘Past’ by scrolling the picker and stopping on ‘Past’ 4. Smith1 taps ‘Add’ button. 5. Entered ‘Name’ will appear below Medication Information text fields. 6. Medication Information text fields will be emptied. 7. Smith1 taps on drop-down arrow next to ‘Name’ below Medication Information text fields. 8. Entered Dosage and Status will appear below Entered ‘Name’. 9. Smith1 taps “Save and Continue’ |
| Expected Results | 1. Smith1 is directed to the ‘List of Surgeries’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter List of Surgeries** |
| **Test Case ID** | **TC - 16** |
| Priority | High |
| Preconditions | 1. Smith1 is logged in successfully. 2. Smith1 is on the ‘Enter List of Surgeries’ page 3. System executes DBS-16 |
| Postconditions | 1. System executes DBS-17 |
| Test Steps | 1. Smith1 enters ‘Surgery Name’ in designated text field as “Kidney transplant” 2. Smith1 taps on drop-down arrow next to ‘Surgery Name’ text field 3. Additional text fields will be displayed below ‘Surgery Name’ 4. Smith1 enters ‘Surgery Date’ in the designated text field as ‘12/12/2005’ 5. Smith1 enters ‘Surgery Description’ in the designated text field as ‘Doing well now’’ 6. Smith1 taps ‘Add’ button. 7. Entered ‘Name’ will appear below Surgery Information text fields. 8. Surgery Information text fields will be emptied. 9. Smith1 taps on drop-down arrow next to ‘Name’ below Surgery Information text fields. 10. Entered Date and Description will appear below Entered ‘Name’. 11. Smith1 taps “Save and Continue’ |
| Expected Results | 1. Smith1 is directed to the ‘List of Allergies’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter List of Allergies** |
| **Test Case ID** | **TC - 17** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Allergies’ page 3. System executes DBS-18 |
| Postconditions | 1. System executes DBS-19 |
| Test Steps | 1. Smith1 enters ‘Allergy Name’ in designated text field as ‘Peanuts’ 2. Smith1 taps on arrow next to ‘Allergy Name’ text field 3. Additional text fields will be displayed below ‘Allergy Name’’ 4. Smith1 enters ‘Allergy Medication’ in the designated text field as ‘None’ 5. Smith1 enters ‘Allergy Treatment’ in the designated text field as ‘None’ 6. Smith1 taps ‘Add’ button. 7. Entered ‘Name’ will appear below Medication Information text fields. 8. Medication Information text fields will be emptied. 9. Smith1 taps on drop-down arrow next to ‘Name’ below Medication Information text fields. 10. Entered Medication and Treatment will appear below Entered ‘Name’. 11. Smith1 taps “Save and Continue’ |
| Expected Results | 1. Smith1 is directed to the ‘List of Vaccines’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter List of Vaccines** |
| **Test Case ID** | **TC - 18** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Vaccines’ page 3. System executes DBS-20 |
| Postconditions | 1. System executes DBS-21 |
| Test Steps | 1. Smith1 enters ‘Vaccine Name’ in designated text field as ‘Hepatitis B’ 2. Smith1 taps on arrow next to ‘Vaccine Name’ text field 3. Additional text fields will be displayed below ‘Vaccine Name’ 4. Smith1 enters ‘Date’ in the designated text field in the following format: MM/DD/YYYY as ‘12/12/2014’ 5. Smith1 taps ‘Add’ button 6. Entered ‘Name’ will appear below Vaccine Information text fields. 7. Vaccine Information text fields will be emptied. 8. Smith1 taps on drop-down arrow next to ‘Name’ below Vaccine Information text fields. 9. Entered Date will appear below Entered ‘Name’. 10. Smith1 taps “Save and Continue’ button |
| Expected Results | 1. Smith1 is directed to the ‘Medical History’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter Medical History - Successful** |
| **Test Case ID** | **TC - 19** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Medical History’ page 3. System executes DBS-22 |
| Postconditions | 1. System executes DBS-23 |
| Test Steps | 1. Smith1 enters ‘Family History’ in designated text view as ‘No blood related Illnesses’ 2. Smith1 enters ‘Note’ in designated text field as ‘No additional notes’ 3. Smith1 taps ‘Save and Continue’ button |
| Expected Results | 1. Smith1 is directed to the ‘Insurance Information’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Enter Insurance Information - Successful** |
| **Test Case ID** | **TC - 20** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Insurance Information’ page 3. System executes DBS-24 |
| Postconditions | 1. System executes DBS-25 |
| Test Steps | 1. Smith1 enters ‘Type of Insurance’ in designated text field containing the following format: Between 2-30 characters in length containing letters [A-Z, a-z] as ‘Medicaid’ 2. Smith1 enters ‘Name of Insurance’ in designated text field containing the following format: Between 2-30 characters in length containing letters [A-Z, a-z] as ‘Blue Cross’ 3. Smith1 enters ‘Group ID’ in designated text field as ‘ABCXYZ123’ 4. Smith1 enters ‘Expiration Date’ in designated text field in the following format: MM/DD/YYYY as ‘12/30/2017’ 5. Smith1 taps ‘Save and Finish’ button |
| Expected Results | 1. An alert appears with the message “Upload Document? Would you like to Upload a Document?” |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete List of Doctors** |
| **Test Case ID** | **TC - 21** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Doctors’ page |
| Postconditions | 1. Smith1’s Doctor Information is deleted from the database’s ‘Doctor’ table |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Doctor Name row(s) 3. The selected Doctor Information shifts to the left of the screen, and Smith1 taps on the ‘Delete’ button that is visible to the right of the Doctor Information 4. Smith1 taps on ‘Done’ button on top-right corner of screen. 5. The red, circular icon that is visible to the left of the Doctor Name row(s) disappear(s) |
| Expected Results | 1. Smith1’s Doctor Information is deleted from the ‘List of Doctors’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete List of Illnesses/Diseases** |
| **Test Case ID** | **TC - 22** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Illnesses/Diseases’ page |
| Postconditions | 1. Smith1’s Illness/Disease Information is deleted from the database’s ‘Illnesses’ table |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Illness/Disease Name row(s) 3. The selected Illness/Disease Information shifts to the left of the screen, and Smith1 taps on the ‘Delete’ button that is visible to the right of the Illness/Disease Information 4. Smith1 taps on ‘Done’ button on top-right corner of screen. 5. The red, circular icon that is visible to the left of the Illness/Disease Name row(s) disappear(s) |
| Expected Results | 1. Smith1’s Illness/Disease Information is deleted from the ‘List of Illnesses/Diseases’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete List of Medications** |
| **Test Case ID** | **TC - 23** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Medications’ page |
| Postconditions | 1. Smith1’s Medication Information is deleted from the database’s ‘Medication’ table |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Medication Name row(s) 3. The selected Medication Information shifts to the left of the screen, and Smith1 taps on the ‘Delete’ button that is visible to the right of the Medication Information 4. Smith1 taps on ‘Done’ button on top-right corner of screen. 5. The red, circular icon that is visible to the left of the Medication Name row(s) disappear(s) |
| Expected Results | 1. Smith1’s Medication Information is deleted from the ‘List of Medications’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete List of Surgeries** |
| **Test Case ID** | **TC - 24** |
| Priority | Medium |
| Preconditions | 1. Smith1 is logged in successfully. 2. Smith1 is on the ‘Enter List of Surgeries’ page |
| Postconditions | 1. Smith1’s Surgery Information is deleted from the database’s ‘Surgery’ table |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Surgery Name row(s) 3. The selected Surgery Information shifts to the left of the screen, and Smith1 taps on the ‘Delete’ button that is visible to the right of the Surgery Information 4. Smith1 taps on ‘Done’ button on top-right corner of screen. 5. The red, circular icon that is visible to the left of the Surgery Name row(s) disappear(s) |
| Expected Results | 1. Smith1’s Surgery Information is deleted from the ‘List of Surgeries’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete List of Allergies** |
| **Test Case ID** | **TC - 25** |
| Priority | Medium |
| Preconditions | 1. Smith1 is logged in successfully. 2. Smith1 is on the ‘Enter List of Allergies’ page |
| Postconditions | 1. Smith1’s Allergy Information is deleted from the database’s ‘Allergy’ table |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Allergy Name row(s) 3. The selected Allergy Information shifts to the left of the screen, and Smith1 taps on the ‘Delete’ button that is visible to the right of the Allergy Information 4. Smith1 taps on ‘Done’ button on top-right corner of screen. 5. The red, circular icon that is visible to the left of the Allergy Name row(s) disappear(s) |
| Expected Results | 1. Smith1’s Allergy Information is deleted from the ‘List of Surgeries’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete List of Vaccines** |
| **Test Case ID** | **TC - 26** |
| Priority | Medium |
| Preconditions | 1. Smith1 is logged in successfully. 2. Smith1 is on the ‘Enter List of Vaccines’ page |
| Postconditions | 1. Smith1’s Vaccine Information is deleted from the database’s ‘Vaccine’ table |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Vaccine Name row(s) 3. The selected Vaccine Information shifts to the left of the screen, and Smith1 taps on the ‘Delete’ button that is visible to the right of the Vaccine Information 4. Smith1 taps on ‘Done’ button on top-right corner of screen. 5. The red, circular icon that is visible to the left of the Vaccine Name row(s) disappear(s) |
| Expected Results | 1. Smith1’s Vaccine Information is deleted from the ‘List of Vaccines’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit Doctor Information** |
| **Test Case ID** | **TC - 27** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Doctors’ page |
| Postconditions | 1. System executes DBS - 48 |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Doctor Name row(s) 3. The selected Doctor Information shifts to the left of the screen, and Smith1 taps on the ‘Edit’ button that is visible to the right of the Doctor Information 4. Smith1 is directed to the ‘Edit Doctor’ page 5. Smith1 changes Doctor Name from “Dr. Amrit” to “Dr. Misra” 6. Smith1 taps on ‘Update’ button 7. An Alert with the following message appears on the iOS device screen: “Edit Status Update was successful” |
| Expected Results | 1. Smith1’s Doctor Information is Updated/Edited in ‘List of Doctors’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit List of Illnesses/Diseases** |
| **Test Case ID** | **TC - 28** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Illnesses/Diseases’ page |
| Postconditions | 1. System executes DBS - 49 |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Illness/Disease Name row(s) 3. The selected Illness/Disease Information shifts to the left of the screen, and Smith1 taps on the ‘Edit’ button that is visible to the right of the Illness/Disease Information 4. Smith1 is directed to the ‘Edit Illnesses/Diseases’ page 5. Smith1 changes Illness Name from “Heart Failure” to “Heart Attack” 6. Smith1 taps on ‘Update’ button 7. An Alert with the following message appears on the iOS device screen: “Edit Status Update was successful” |
| Expected Results | 1. Smith1’s Illness/Disease Information is Updated/Edited in ‘List of Illnesses/Diseases’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit List of Medication** |
| **Test Case ID** | **TC - 29** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Medication’ page |
| Postconditions | 1. System executes DBS - 50 |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Medication Name row(s) 3. The selected Medication Information shifts to the left of the screen, and Smith1 taps on the ‘Edit’ button that is visible to the right of the Medication Information 4. Smith1 is directed to the ‘Edit Medication page 5. Smith1 changes Medication Name from “Amoxicilline” to “Plavix”’ 6. Smith1 taps on ‘Update’ button 7. An Alert with the following message appears on the iOS device screen: “Edit Status Update was successful” |
| Expected Results | 1. Smith1’s Medication Information is Updated/Edited in ‘List of Medication’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit List of Surgery** |
| **Test Case ID** | **TC - 30** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Surgeries’ page |
| Postconditions | 1. System executes DBS - 51 |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Surgery Name row(s) 3. The selected Surgery Information shifts to the left of the screen, and Smith1 taps on the ‘Edit’ button that is visible to the right of the Surgery Information 4. Smith1 is directed to the ‘Edit Surgery’ page 5. Smith1 changes Surgery Name from “Kidney transplant” to “Brain transplant” 6. Smith1 taps on ‘Update’ button 7. An Alert with the following message appears on the iOS device screen: “Edit Status Update was successful” |
| Expected Results | 1. Smith1’s Surgery Information is Updated/Edited in ‘List of Surgeries’’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit List of Allergies** |
| **Test Case ID** | **TC - 31** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Allergies’ page |
| Postconditions | 1. System executes DBS - 52 |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the allergy Name row(s) 3. The selected Allergy Information shifts to the left of the screen, and Smith1 taps on the ‘Edit’ button that is visible to the right of the Allergy name 4. Smith1 is directed to the ‘Edit Allergies’ page 5. Smith1 changes Allergy Name from “Peanuts” to 6. “Dust” 7. Smith1 taps on ‘Update’ button 8. An Alert with the following message appears on the iOS device screen: “Edit Status Update was successful” |
| Expected Results | 1. Smith1’s allergy Information is Updated/Edited in ‘List of Allergies’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit List of Vaccines** |
| **Test Case ID** | **TC - 32** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Enter List of Vaccines’ page |
| Postconditions | 1. System executes DBS - 53 |
| Test Steps | 1. Smith1 taps on ‘Edit’ button on top-right corner of screen. 2. Smith1 taps on the red, circular icon that is visible to the left of the Vaccine Name row(s) 3. The selected Vaccine Information shifts to the left of the screen, and Smith1 taps on the ‘Edit’ button that is visible to the right of the Vaccine name 4. Smith1 is directed to the ‘Edit Vaccines’ page 5. Smith1 changes Vaccine Name from “Hepatitis B” to “Hepatitis A” 6. Smith1 taps on ‘Update’ button 7. An Alert with the following message appears on the iOS device screen: “Edit Status Update was successful” |
| Expected Results | 1. Smith1’s Vaccine Information is Updated/Edited in ‘List of Vaccines’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **View Medical Summary** |
| **Test Case ID** | **TC - 33** |
| Priority | High |
| Preconditions | 1. TC -05 has passed. 2. System executes DBS-25 3. Smith1 is on the ‘Insurance Information’ page |
| Postconditions | 1. System executes DBS-37 to DBS-45 |
| Test Steps | 1. Smith1 taps “Save and Finish’ on Insurance Information page 2. An alert appears with the message “Would you like to Upload a Document?” 3. Smith1 taps on the ‘No’ button. |
| Expected Results | 1. The alert disappears from the screen 2. Smith1 is directed to Medical Summary page 3. Smith1’s Medical Information is displayed on the ‘Medical Summary’ page |

### 1.1.3 Documents

|  |  |
| --- | --- |
| **Test Case Name** | **Upload Document through Camera - Successful** |
| **Test Case ID** | **TC - 34** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Upload Document’ page 3. Smith1 provided application permission to access camera 4. System executes DBS-26 |
| Postconditions | 1. Alert with ‘Document Name ‘and ‘Document Description’ appears on iOS device |
| Test Steps | 1. Smith1 taps ‘Camera’ button 2. Camera view appears 3. Smith1 taps ‘Capture’ button 4. Smith1 taps ‘Use Photo’ button 5. Smith1 taps ‘Save’ button |
| Expected Results | 1. The image captured is uploaded to MyHealthKeeper |

|  |  |
| --- | --- |
| **Test Case Name** | **Upload Document through Camera - Unsuccessful** |
| **Test Case ID** | **TC - 35** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the ‘Upload Document’ page 3. Smith1 provided application permission to access camera 4. System executes DBS-26 5. Camera is not available on the device |
| Postconditions | 1. UIImagePickerController searches for Source Type as Camera and returns false |
| Test Steps | 1. Smith1 taps ‘Camera’ button on ‘Upload Document’ page |
| Expected Results | 1. Error message appears with the message “Camera is not available.” |

|  |  |
| --- | --- |
| **Test Case Name** | **Upload Document through Photo Library - Successful** |
| **Test Case ID** | **TC - 36** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Upload Document’ page 3. Smith1 provided application permission to access photo library 4. System executes DBS-26 5. Images are stored in Smith1’s iOS device’s photo library |
| Postconditions | 1. Alert with ‘Document Name‘ and ‘Document Description’ appears on iOS device |
| Test Steps | 1. Smith1 taps on ‘Photo Library’ button 2. Alert appears stating “MyHealthKeeper” would like to access your photos Allow app to store and save selected image. 3. Smith1 taps ‘OK’ button 4. Photo library appears on iOS device screen with images 5. Smith1 taps on an image displayed 6. Smith1 is directed to Upload Document page with image displayed on iOS device screen 7. Smith1 taps ‘Save’ button |
| Expected Results | 1. The selected image is uploaded to MyHealthKeeper |

|  |  |
| --- | --- |
| **Test Case Name** | **Upload Document through Photo Library - Unsuccessful** |
| **Test Case ID** | **TC - 37** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the ‘Upload Document’ page 3. Smith1 provided application permission to access photo library 4. System executes DBS-26 5. Photo Library is not available on the device. |
| Postconditions | 1. UIImagePickerController searches for Source Type as Photo Library and returns false |
| Test Steps | 1. Smith1 taps on ‘Photo Library’ button |
| Expected Results | 1. Error message appears with the message “Photo Library is not available.” |

|  |  |
| --- | --- |
| **Test Case Name** | **File Size Limit** |
| **Test Case ID** | **TC - 38** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Upload Document’ page 3. TC - 34 or TC - 36 passes |
| Postconditions | 1. File Size is calculated    1. File size is less than 4MB |
| Test Steps | 1. Smith1 taps on the ‘Save’ button on the bottom right corner of the page. |
| Expected Results | 1. An alert appears with a message which states “Enter Document Information”    1. Text fields are provided in the alert |

|  |  |
| --- | --- |
| **Test Case Name** | **File Size Limit Error** |
| **Test Case ID** | **TC - 39** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Upload Document’ page 3. TC - 34 or TC - 36 passes |
| Postconditions | 1. File Size is calculated    1. File size is greater than 4MB |
| Test Steps | 1. Smith1 taps ‘Save’ button |
| Expected Results | 1. An alert appears with a message stating “The image is too large. Please select another image. |

|  |  |
| --- | --- |
| **Test Case Name** | **Save Document Information** |
| **Test Case ID** | **TC - 40** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Upload Document’ page 3. TC - 34 or TC - 36 passes 4. TC - 38 has passed 5. Execute DBS - 26 |
| Postconditions | 1. Image is saved to the iOS device’s default directory 2. System executes DBS - 27 |
| Test Steps | 1. Smith1 enters “Test Document Name” in ‘Document Name’ text field 2. Smith1 enters “Test Document Description” in ‘Document Description’ text field. 3. Smith1 taps the ‘OK’ button on the alert |
| Expected Results | 1. Smith1 is directed to the Documents Summary Page 2. The image, name, and description entered are displayed on the Documents Summary page on the last row |

|  |  |
| --- | --- |
| **Test Case Name** | **Save Document Information - Unsuccessful** |
| **Test Case ID** | **TC - 41** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Upload Document’ page 3. TC - 34 or TC - 36 passes 4. TC - 38 has passed 5. Execute DBS - 26 |
| Postconditions | 1. System checks if ‘Document Name’ field is empty |
| Test Steps | 1. Smith1 leaves ‘Document Name’ text field empty. 2. Smith1 taps the ‘OK’ button on the Document Information alert. |
| Expected Results | 1. An alert appears with an error message stating “Document Name field cannot be empty. Please enter a value.” |

|  |  |
| --- | --- |
| **Test Case Name** | **View Uploaded Documents - Images Uploaded** |
| **Test Case ID** | **TC - 42** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Main’ page 3. TC - 34 or TC - 36 passes 4. TC - 38 has passed 5. TC - 40 has passed |
| Postconditions | System executes DBS - 46 |
| Test Steps | 1. Smith1 taps ‘Menu’ button on top-left corner of the Main page 2. Smith1 taps ‘View Uploaded Documents’ on Menu 3. Smith1 is directed to ‘Document Summary’ page |
| Expected Results | 1. Uploaded documents are displayed from oldest - newest 2. Each document’s corresponding name and description are displayed to the right of the image |

|  |  |
| --- | --- |
| **Test Case Name** | **View Uploaded Documents - Images Not Uploaded** |
| **Test Case ID** | **TC - 43** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Main’ page 3. System executes DBS - 26 |
| Postconditions | System executes DBS - 46 |
| Test Steps | 1. Smith1 taps ‘Menu’ on the top-left corner of the Main page 2. Smith1 taps ‘View Uploaded Documents’ on Menu 3. Smith1 is directed to Document Summary Page |
| Expected Results | 1. An error message appears indicating “You have not uploaded any images.” 2. Image, document name, and document description are not displayed |

|  |  |
| --- | --- |
| **Test Case Name** | **View Uploaded Documents - Enlarge image** |
| **Test Case ID** | **TC - 44** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Main’ page 3. TC - 34 or TC - 36 passes 4. TC - 38 has passed 5. TC - 40 has passed |
| Postconditions | 1. Viewcontroller, with an expanded imageview, is pushed to the view |
| Test Steps | 1. Smith1 taps ‘Menu’ on the top-left corner of the Main page 2. Smith1 taps ‘View Uploaded Documents’ 3. Smith1 is directed to ‘Document Summary’ page 4. Smith1 taps on an image |
| Expected Results | 1. Selected image is scaled fit the page    1. Image maintains aspect ratio, is on a black background, and only the selected image is displayed on the screen |

|  |  |
| --- | --- |
| **Test Case Name** | **View Uploaded Documents - Shrink Image to Original Displayed Size** |
| **Test Case ID** | **TC - 45** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Main’ page 3. TC - 34 or TC - 36 passes 4. TC - 38, 40, and 44 has passed |
| Postconditions | 1. Viewcontroller, with the Documents Summary Page, is pushed to the view |
| Test Steps | 1. Smith1 taps on the enlarged image |
| Expected Results | 1. Selected image resizes to the original displayed size 2. Smith1 is directed to Documents Summary Page 3. All Uploaded documents are displayed from oldest - newest 4. Each document’s corresponding name and description are displayed on the right of the image |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete Documents - Tap ‘Delete’ button** |
| **Test Case ID** | **TC - 46** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the ‘Document Summary’ page 3. TC - 34 or TC - 36 passes 4. TC - 38, 40, and 42 has passed |
| Postconditions | 1. DBS - 28 is executed 2. Image is deleted from application’s default directory |
| Test Steps | 1. Smith1 taps on ‘Delete’ button on the top right corner of the navigation bar 2. Smith1 taps on the red icon to the left of the first image 3. The selected cell slides left, and a red ‘Delete’ block appears to the right of the row 4. Smith1 taps on the red ‘Delete’ block |
| Expected Results | 1. Row with the image, name, and description are not displayed on the Documents Summary page    1. Remaining cells move up |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete Documents - Slide Selected Cell** |
| **Test Case ID** | **TC - 47** |
| Priority | Medium |
| Preconditions | 1. Smith1 is logged in successfully. 2. Smith1 is on the ‘Document Summary’ page 3. TC - 34 or TC - 36 passes 4. TC - 38, 40, and 42 has passed |
| Postconditions | 1. DBS - 28 is executed 2. Image is deleted from application’s default directory |
| Test Steps | 1. Smith1 slides the first cell, which contains an image and its corresponding name and description, from right to left on the iOS device’s screen 2. Smith1 taps the red ‘Delete’ block to the right of the row of the selected image |
| Expected Results | 1. The selected image, name, and description is not displayed on the ‘Documents Summary’ page    1. Remaining images, names, and descriptions move up on the page |

### 1.1.4 Reminders/Notifications

|  |  |
| --- | --- |
| **Test Case Name** | **Add Reminder - Successful** |
| **Test Case ID** | **TC - 48** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Reminders’ page 3. Smith1 has given application permission to send notifications to device 4. Execute DBS-00 and DBS-29 |
| Postconditions | 1. System executes DBS-30. |
| Test Steps | 1. Smith1 enters a reminder name in the ‘Reminder Name’ text field as ‘TestName’ 2. Smith1 enters a location in the ‘Reminder Location’ text field as” TestLocation” 3. Smith1 enters a reason in the ‘Reminder Reason’ text field as” TestReason” 4. Smith1 enters a test time in the ‘Reminder Time’ picker as the “12/12/2017 1:20pm” 5. Smith1 taps ‘Submit and Continue’ button |
| Expected Results | 1. Smith1’s iOS device receives an alert with the following message: “Insert was successful” 2. User will receive a notification on their device at 12/12/2017 at 1:20pm |

|  |  |
| --- | --- |
| **Test Case Name** | **Add Reminder - Unsuccessful** |
| **Test Case ID** | **TC - 49** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on ‘Reminder’ page 3. Execute DBS-00 and DBS-29 |
| Postconditions | 1. Reminder Information is not inserted into database |
| Test Steps | 1. Smith1 enters a location in the ‘Reminder Location’ text field as ”TestLocation” 2. Smith1 enters a reason in the ‘Reminder Reason’ text field as ”TestReason” 3. Smith1 enters a test time in the ‘Reminder Time’ picker as the “12/12/2017 1:20pm” 4. Smith1 taps ‘Submit and Continue’ button |
| Expected Results | 1. Smith1’s iOS device receives an alert with the following message: ‘Invalid Input: Reminder Name cannot be empty. Please enter a value.” |

|  |  |
| --- | --- |
| **Test Case Name** | **Add Reminder Without Giving Application Permission to Send Notifications to the iOS Device** |
| **Test Case ID** | **TC - 50** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the ‘Reminders’ page 3. Smith1 has not given the application permission to send notifications 4. Execute DBS-00 and DBS-29 |
| Postconditions | 1. System executes DBS-30. |
| Test Steps | 1. User enters a test reminder name in the ‘Reminder Name’ text field as “TestName” 2. Smith1 enters a test location in the ‘Reminder Location’ text field as “TestLocation” 3. Smith1 enters test reason in the ‘Reminder Reason’ text field as “TestReason” 4. Smith1 enters a test time in the ‘Reminder Time’ picker as the “12/12/2017 1:20pm” 5. Smith1 taps ‘Submit and Continue’ button |
| Expected Results | 1. Smith1’s iOS device receives an alert with the following message: “Insert was successful” 2. Smith1’s iOS device receives an alert with the following message: “Please enable notifications for this app in order to receive notifications. Please give this application permission to send notifications by adjusting the settings of the application” |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit Reminder - Successful** |
| **Test Case ID** | **TC - 51** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 has given application permission to send notifications to device 3. Smith1 is on the ‘View Reminder’ page 4. Execute DBS-00 and DBS-29 and DBS-30 |
| Postconditions | 1. System executes DBS-35 2. The reminder notification sent to Smith1’s iOS device will be updated in the notification center with the information Smith1 enters |
| Test Steps | 1. Smith1 taps on the Reminder cell which has the name “TestName 12/12/2017 1:20pm’’ 2. Smith1 taps ‘Yes’ on the alert with the following message: “Would you like to edit this reminder?” 3. Smith1 is redirected to the Edit Reminder page 4. Smith1 updates Reminder Location in the Reminder Location text field as “TestLocaton2” 5. Smith1 updates Reminder Reason in the Reminder Reason text field as “TestReason2” 6. Smith1 taps ‘Update’ button |
| Expected Results | 1. Smith1’s iOS device receives an alert with the following message: “Reminder Update was successful” |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit Reminder - Unsuccessful** |
| **Test Case ID** | **TC - 52** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on ‘View Reminder’ page 3. Execute DBS-00 and DBS-29 and DBS-30 |
| Postconditions | 1. The reminder that Smith1 wishes to update (“TestName”) will not get updated in the database |
| Test Steps | 1. Smith1 taps on the Reminder cell which has the name “TestName 12/12/2017 1:20pm’’ 2. Smith1 taps ‘Yes’ on the alert with the following message: “Would you like to edit this reminder?” 3. Smith1 is directed to the ‘Edit Reminder’ page 4. Smith1 deletes all characters from the ‘Reminder Name’ text field 5. Smith1 taps ‘Update’ button |
| Expected Results | 1. Smith1’s iOS device receives an alert with the following message: “Invalid Input: Reminder Name cannot be empty. Please enter a Reminder Name” |

|  |  |
| --- | --- |
| **Test Case Name** | **Edit Reminder Without Giving Application Permission to Send Notifications to the iOS Device** |
| **Test Case ID** | **TC - 53** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘View Reminders’ page 3. Execute DBS-00 and DBS-29 and DBS-30 |
| Postconditions | 1. System executes DBS-35 |
| Test Steps | 1. Smith1 taps on the Reminder cell which has the name “TestName 12/12/2017 1:20pm’’ 2. Smith1 taps ‘Yes’ on the alert with the following message: “Would you like to edit this reminder?” 3. Smith1 is directed to ‘Edit Reminder’ page 4. Smith1 updates the Reminder Location in the ‘Reminder Location’ text field as “TestLocation2” 5. Smith1 updates the Reminder Reason in the ‘Reminder Reason’ text field as “TestReason2” 6. Smith1 taps ‘Update’ button |
| Expected Results | 1. Smith1’s iOS device receives an alert with the following message: ‘Reminder Update was successful’ 2. Smith1’s iOS device receives an alert with the following message: “Please enable notifications for this app in order to receive notifications. Please give this application permission to send notifications by adjusting the settings of the application” |

|  |  |
| --- | --- |
| **Test Case Name** | **Delete Reminder - Successful** |
| **Test Case ID** | **TC - 54** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘View Reminders’ page 3. Execute DBS-00 and DBS-29 and DBS-30 |
| Postconditions | 1. System executes DBS-36 2. Smith1’s iOS device will not receive the notification for the deleted reminder at 12/12/2017 at 1:20pm |
| Test Steps | 1. Smith1 taps ‘Delete’ on the navigation bar on the ‘View Reminders’ page 2. Smith1 taps on the red, circular icon that appears on the left side of the reminder cell with the name “TestName 12/12/2017 1:20pm”    1. The selected Reminder Information slides to the left of the screen    2. A red Delete button appears to the right side of the selected Reminder Information 3. Smith1 taps ‘Delete’ button 4. Smith1 taps ‘Done’ button on the top-corner of the page |
| Expected Results | 1. The “TestName 12/12/2017 1:20pm” Reminder Cell is deleted from table |

|  |  |
| --- | --- |
| **Test Case Name** | **No Available Reminders on Reminder Summary** |
| **Test Case ID** | **TC - 55** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Main’ page 3. Execute DBS-00 and DBS-29 |
| Postconditions | 1. The ‘Reminder Summary’ page does not display any reminders |
| Test Steps | 1. Smith1 taps ‘Menu’ 2. Smith1 taps ‘View Reminders’ 3. Smith1 is taken to the ‘ViewReminders’ page |
| Expected Results | 1. An alert appears with following message: “You have not set any appointment reminders“ |

|  |  |
| --- | --- |
| **Test Case Name** | **View Reminder Summary** |
| **Test Case ID** | **TC - 56** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Main’ page 3. System executes DBS-00 and DBS - 29 and DBS-30 |
| Postconditions | 1. System executes DBS-47 2. All of Smith1’s reminders are displayed on the Reminder Summary page. |
| Test Steps | 1. Smith1 taps the ‘Menu’ button 2. Smith1 taps ‘Reminder Summary’ on the Menu 3. Smith1 is redirected to the Reminder Summary page |
| Expected Results | 1. The reminder cell with the name “TestName 12/12/2017 1:20pm” is displayed on ‘View Reminders’ page |

### 1.1.5 Monthly Reminder

|  |  |
| --- | --- |
| **Test Case Name** | **Turning Off Monthly Reminders** |
| **Test Case ID** | **TC - 57** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘View Reminders’ page 3. System executes DBS-00, DBS-34, DBS-32 |
| Postconditions | 1. System executes DBS-33. 2. Smith1’s iOS device will not receive a reminder message prompting them to update their Medical History at 12pm on the first date of every month. |
| Test Steps | 1. Smith1 taps on Monthly Reminder switch to ‘Off’ position |
| Expected Results | 1. Smith1’s iOS device receives the message: “You will not receive monthly notifications to remind you to update your medical history.” |

|  |  |
| --- | --- |
| **Test Case Name** | **Turning on Monthly Reminders** |
| **Test Case ID** | **TC - 58** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the ‘View Reminders’ page 3. Smith1 has given application permission to send notifications to the iOS device 4. System executes DBS-00, DBS-34, DBS-33 |
| Postconditions | 1. System executes DBS-32. 2. Smith1 will receive a reminder notification with message: ”Please update your medical information on myHealthKeeper” prompting them to update their medical history at 12pm on the first of every month. |
| Test Steps | 1. Smith1 taps on monthly reminder switch to ‘On’ position |
| Expected Results | 1. Smith1’s iOS device receives the alert with the following message: “Monthly notifications to reminder you to update your medical history have been turned on. You will receive a notification on the first day of every month at 12pm reminding to update your medical information.” |

|  |  |
| --- | --- |
| **Test Case Name** | **Turning On Monthly Reminders without Giving the Application Permission to send Notifications to the iOS device** |
| **Test Case ID** | **TC - 59** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on View Reminder page 3. Smith1 has not given application permission to send notifications to the iOS device 4. System executes DBS-00, DBS-34, DBS-33 |
| Postconditions | 1. System executes DBS-32. 2. Smith1’s iOS device will not receive a reminder message prompting them to update their medical history at 12pm on the first of every month. |
| Test Steps | 1. Smith1 taps on monthly reminder switch to ‘On’ position |
| Expected Results | 1. Smith1’s iOS device receives an alert with the following message: ”Monthly notifications to remind you to update your medical history have been turned on. You will receive a notification on the first day of every month at 12pm reminding to update your medical information. Warning: you have not given the application permission to send notifications. Please give this application permission to send notifications by adjusting the settings of the application.” |

### 1.1.5 Print

|  |  |
| --- | --- |
| **Test Case Name** | **Print Report without an Image - Successful** |
| **Test Case ID** | **TC - 60** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the Main page |
| Postconditions | 1. System connects to the selected AirPrint enabled printer and performs the print job. |
| Test Steps | 1. Smith1 taps ‘Menu’ button on the top-left corner of the screen 2. Menu appears on the screen 3. Smith1 selects ‘Print’ from the Menu 4. Smith1 is directed to the ‘Print Report’ page 5. An alert appears with the following message: “Select an image to Print? Would you like to print an image?” 6. Smith1 selects ‘No’. 7. Smith1 performs one of the following:    1. Smith1 taps ‘Print Standard Report’ button       1. Smith1’s username is used as a foreign key in the database to retrieve that user’s Medical Information       2. Smith1 is directed to the ‘Printer Options’ page       3. Smith1’s retrieved Medical Information is displayed on the ‘Printer Options’ page    2. Smith1 taps ‘Print Detailed Report’ button       1. Smith1’s username is used as a foreign key in the database to retrieve that user’s Medical Information       2. Smith1 is directed to the ‘Printer Options’ page       3. Smith1’s retrieved Medical Information is displayed on the ‘Printer Options’ page 8. Smith1 selects a Printer 9. Smith1 selects number of copies 10. Smith1 selects Printer options (if available) 11. Smith1 taps ‘Print’ button on top-right corner of the screen |
| Expected Results | 1. Smith1’s Medical Information is printed on an 8x11 paper |

|  |  |
| --- | --- |
| **Test Case Name** | **Print Report with an Image - Successful** |
| **Test Case ID** | **TC - 61** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the Main page 3. Smith1 has already uploaded at least 1 image to MyHealthKeeper |
| Postconditions | 1. System connects to the selected AirPrint enabled printer and performs the print job. |
| Test Steps | 1. Smith1 taps ‘Menu’ button on the top-left corner of the screen 2. Menu appears on the screen 3. Smith1 selects ‘Print’ from the Menu 4. Smith1 is directed to the ‘Print Report’ page 5. An alert appears with the following message: “Select an image to Print? Would you like to print an image?” 6. Smith1 selects ‘Yes’ on the alert. 7. Smith1 is directed to the Uploaded Documents page. 8. Smith1 selects an image by tapping on the image row. 9. An alert with the following message appears: “Image Status Would you like to select and print this image in your report?” 10. Smith1 selects ‘Yes’ on the alert. 11. Smith1 is directed back to the Print Report page with the selected image displayed on the iOS device’s screen. 12. Smith1 performs one of the following:     1. Smith1 taps ‘Print Standard Report’ button        1. Smith1’s username is used as a foreign key in the database to retrieve that user’s Medical Information        2. Smith1 is directed to the ‘Printer Options’ page        3. Smith1’s retrieved Medical Information and selected image is displayed on the ‘Printer Options’ page     2. Smith1 taps ‘Print Detailed Report’ button        1. Smith1’s username is used as a foreign key in the database to retrieve that user’s Medical Information        2. Smith1 is directed to the ‘Printer Options’ page        3. Smith1’s retrieved Medical Information and selected image is displayed on the ‘Printer Options’ page 13. Smith1 selects a Printer 14. Smith1 selects number of copies 15. Smith1 selects Printer options (if available) 16. Smith1 taps ‘Print’ button on top-right corner of the screen |
| Expected Results | 1. Smith1’s Medical Information is printed on an 8x11 paper |

### 1.1.7 Menu

|  |  |
| --- | --- |
| **Test Case Name** | **Navigate to Another Page through Menu** |
| **Test Case ID** | **TC - 62** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed 2. A menu is available on the left corner of the navigation bar |
| Postconditions | 1. Print Report viewcontroller is pushed to the view |
| Test Steps | 1. User taps on ‘Menu’ 2. User taps on desired page by tapping on ‘Print’ |
| Expected Results | 1. User is directed to desired page Print Report |

|  |  |
| --- | --- |
| **Test Case Name** | **Menu - View Page Options for Edit Medical Information** |
| **Test Case ID** | **TC - 63** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed 2. A Menu button is displayed on the top- left corner of the Navigation bar |
| Postconditions | 1. Additional page options are displayed on the Menu |
| Test Steps | 1. User taps on ‘Menu’ 2. User taps on ‘Edit Medical Information’ |
| Expected Results | 1. Additional cells are displayed under ‘Edit Medical Information’    1. Cells are displayed in a lighter color with smaller text |

### 1.1.8 User Logout

|  |  |
| --- | --- |
| **Test Case Name** | **User Logout** |
| **Test Case ID** | **TC - 64** |
| Priority | High |
| Preconditions | 1. TC - 05 has passed 2. Smith1 is on the Main page |
| Postconditions | 1. Smith1 is signed out of their account 2. Viewcontroller with the Sign In page is pushed to the view |
| Test Steps | 1. Smith1 taps ‘Menu’ button on the top-left corner of the Main page 2. Menu appears on the left side of the screen 3. Smith1 taps ‘Logout’ on the Menu |
| Expected Results | 1. User is directed to the ‘Sign In’ page |

## 

## 1.2 Online Application

### 1.2.1 Server

|  |  |
| --- | --- |
| **Test Case Name** | **Sync Data to Server** |
| **Test Case ID** | **TC - 65** |
| Priority | Medium |
| Preconditions | 1. Smith1 has two iOS devices with MyHealthKeeper downloaded 2. Smith1 is logged in to the application from device number 1 3. Smith1 is on the Main page from device number 1 |
| Postconditions | System executes DBS - 06, DBS-37 to DBS-47 |
| Test Steps | 1. Smith1 taps on ‘Menu’ 2. Smith1 taps on ‘Sync’    1. Data is retrieved from the local database and uses POST method to send data to the AWS server    2. Data is stored in MySQL database on the AWS server    3. Alert appears with a message stating “Sync Successful” 3. Smith1 taps on “Logout” 4. User takes device number 2 and opens up application by tapping on MyHealthKeeper 5. Smith1 logs in by entering “Smith1” into ‘Username’ text field and “Smith1!” into ‘Password’ text field 6. Smith1 taps on ‘Menu’ 7. Smith1 taps on ‘Sync’    1. Data is retrieved from the server using the GET method and is stored in the local database    2. Alert appears with a message stating “Sync Successful” 8. Smith1 taps on ‘View Medical’ on the Menu |
| Expected Results | 1. Smith1 is directed to the Medical Summary page 2. Data sent from device number 1 is displayed on the page |

### 1.2.2 Share Data

|  |  |
| --- | --- |
| **Test Case Name** | **Share Data to Another Device** |
| **Test Case ID** | **TC - 66** |
| Priority | Medium |
| Preconditions | 1. Smith1 and User number 2 have an iOS device. 2. Smith1 has MyHealthKeeper downloaded on his iOS device. 3. Smith1 is logged in to the application 4. Smith1 is on the ‘Print Report’ page 5. Smith1 has tapped ‘Print Standard Report’ or ‘Print Detailed Report’ |
| Postconditions | 1. User number 2 will receive a PDF of the selected report |
| Test Steps | 1. Smith1 presses and holds onto the page of the ‘Print Preview’ until it becomes full-screen 2. Smith1 taps on ‘Share’ button on the bottom-left corner of the application 3. Smith1 taps on a Device Name listed |
| Expected Results | 1. User Two will be prompted with an alert to Accept or Decline the PDF    1. User Two will select ‘Accept’ 2. The PDF of the ‘Print Preview’ will be sent to the Device listed |

### 1.2.3 Print

|  |  |
| --- | --- |
| **Test Case Name** | **Save as PDF** |
| **Test Case ID** | **TC - 67** |
| Priority | Medium |
| Preconditions | 1. Smith1 is logged in to the application 2. Smith1 is on Print Preview Page |
| Postconditions | 1. A PDF of the selected report is saved in the iBook application |
| Test Steps | 1. Smith1 presses and holds onto the page in the Printer Options page    1. Page becomes full-screen 2. Smith1 taps on the ‘Share’ icon on the bottom-left of the screen 3. Smith1 taps ‘Import with iBook’ |
| Expected Results | 1. Smith1 is navigated to the iBook application 2. A PDF of the selected report opens up |

# 2. Non-Functional Testing

## 2.1. Non-Functional Test Cases

|  |  |
| --- | --- |
| **Test Case Name** | **Database Performance** |
| **Test Case ID** | **TC - 68** |
| Priority | High |
| Preconditions | 1. Smith1 is logged into the application |
| Postconditions | 1. Data will be stored/retrieved securely in the Database |
| Test Steps | 1. Execute the following test cases:    * TC - 12 to TC - 27    * TC - 42    * TC - 44    * TC - 46    * TC - 48    * TC - 51    * TC - 54    * TC - 56    * TC - 60 2. Record the time it takes to store/retrieve data from the database for each test case |
| Expected Results | 1. The recorded time for accessing and executing an action in the database for each test case will complete by 3 seconds |

|  |  |
| --- | --- |
| **Test Case Name** | **Time to Upload File** |
| **Test Case ID** | **TC - 69** |
| Priority | High |
| Preconditions | 1. Smith1 is logged into the application |
| Postconditions | 1. Uploaded Image is displayed on the screen |
| Test Steps | 1. Execute test cases TC - 34 and TC - 36 2. Record the time it takes for the image to display on the screen |
| Expected Results | 1. Image is uploaded and displayed on the screen within 1 minute for each test case |

|  |  |
| --- | --- |
| **Test Case Name** | **Error Handling for Database** |
| **Test Case ID** | **TC - 70** |
| Priority | High |
| Preconditions | 1. Smith1 is logged into the application |
| Postconditions | 1. An alert will appear with an error message |
| Test Steps | 1. Execute test cases    * TC - 6    * TC - 8    * TC - 11 |
| Expected Results | 1. An alert will appear with an error message indicating the information entered into the text field is not valid and does not match the text stored in the database. |

|  |  |
| --- | --- |
| **Test Case Name** | **Offline Application Use Availability** |
| **Test Case ID** | **TC - 71** |
| Priority | High |
| Preconditions | N/A |
| Postconditions | 1. The application is available on the user’s iOS device. |
| Test Steps | 1. Smith1 downloads MyHealthKeeper from the App Store. 2. Smith1 opens the application. 3. Smith1 has successfully registered and set up Security Questions |
| Expected Results | 1. Smth1 can use the application on the iOS device that it is installed on. |

|  |  |
| --- | --- |
| **Test Case Name** | **Online Application Use Availability** |
| **Test Case ID** | **TC - 72** |
| Priority | Medium |
| Preconditions | 1. Smith1 has downloaded the application. 2. Smith1 is logged in. 3. The application is connected to the Amazon AWS server. |
| Postconditions | 1. Smith1 can access the application. |
| Test Steps | 1. Smith1 has successfully registered and set up Security Questions |
| Expected Results | 1. Smith1 can access MyHealthKeeper connected to the server. |

|  |  |
| --- | --- |
| **Test Case Name** | **Local SQLite Database for Offline Application** |
| **Test Case ID** | **TC - 73** |
| Priority | Medium |
| Preconditions | N/A |
| Postconditions | 1. Data will be stored in the SQLite Database 2. Data in the database can be viewed from SQLite Manager |
| Test Steps | 1. Smith1 is logged into the application 2. System executes database scripts    * DBS - 02    * DBS - 04    * DBS - 09    * DBS - 11    * DBS - 27    * DBS - 30 |
| Expected Results | 1. Data will be stored in the SQLite Database |

|  |  |
| --- | --- |
| **Test Case Name** | **SQLite Database Encryption** |
| **Test Case ID** | **TC - 74** |
| Priority | Medium |
| Preconditions | 1. The application on the iOS device is storing data on the SQLite Database |
| Postconditions | N/A |
| Test Steps | 1. Smith1 enters data into text fields 2. Smith1 taps on ‘Save’ or ‘Continue’ buttons |
| Expected Results | 1. Data will be sent to database through SQLCipher    1. SQLCipher encrypts the data stored in the database |

|  |  |
| --- | --- |
| **Test Case Name** | **HTTPS for Online Application** |
| **Test Case ID** | **TC - 75** |
| Priority | High |
| Preconditions | 1. Smith1 is logged in to MyHealthKeeper |
| Postconditions | 1. Smith1’s data is sent from the application to the DBS database hosted by AWS using the HTTPS protocol |
| Test Steps | 1. The Smith1 taps ‘Sync’ on the Menu |
| Expected Results | 1. Smith1’s data is sent from the application to the MySQL database hosted by AWS using the HTTPS protocol 2. HTTPS encrypts the Smith1’s data |

|  |  |
| --- | --- |
| **Test Case Name** | **User Authentication** |
| **Test Case ID** | **TC - 76** |
| Priority | High |
| Preconditions | 1. Smith1 has downloaded the application on their iOS device 2. Smith1 has successfully registered and set up Security Questions |
| Postconditions | 1. Smith1 can access their account in MyHealthKeeper |
| Test Steps | 1. Smith1 logs into MyHealthKeeper |
| Expected Result | 1. Smith1 is able to log in to MyHealthKeeper with their username and password 2. Smith1 is able to Reset their Password by answering the Security Questions they set up during Registration |

|  |  |
| --- | --- |
| **Test Case Name** | **Server Application Maintenance Time** |
| **Test Case ID** | **TC - 77** |
| Priority | High |
| Preconditions | 1. MyHealthKeeper is connected to the AWS server |
| Postconditions | N/A |
| Test Steps | 1. Smith1 opens the application when the server is under maintenance. 2. Smith1 attempts to Login to MyHealthKeeper |
| Expected Result | 1. Smith1 will not be able to Login to the application. |

|  |  |
| --- | --- |
| **Test Case Name** | **Addition/Updates to Application Features** |
| **Test Case ID** | **TC - 78** |
| Priority | High |
| Preconditions | 1. Application is downloaded on Smith1’s iOS device |
| Postconditions | 1. Smith1 will receive the Application’s latest features and updates |
| Test Steps | 1. Smith1 taps on ‘Update’ in the App Store 2. Update will start Downloading latest version of the Application |
| Expected Result | 1. After Update is completed, Smith1 will be able to use updated version of application 2. Smith1 will receive the Application’s latest features and updates |

|  |  |
| --- | --- |
| **Test Case Name** | **iOS Compatibility** |
| **Test Case ID** | **TC - 79** |
| Priority | High |
| Preconditions | 1. iOS devices such as iPhone 5s and above, iPad 5th Generation and above, and have a mobile operating system of version 10.0 or higher |
| Postconditions | 1. Application size is compatible with the iOS device’s operating system |
| Test Steps | 1. Open the application on an iOS device with version 10.0 or higher |
| Expected Result | 1. Application size is compatible with the iOS device’s operating system |

|  |  |
| --- | --- |
| **Test Case Name** | **Screen Resolution** |
| **Test Case ID** | **TC - 80** |
| Priority | High |
| Preconditions | 1. iOS devices such as iPhone 5s and above with screen resolutions of 1136 x 640 pixels, and iPad 5th Generation and above with screen resolutions of 2048 x 1536 pixels [1]. |
| Postconditions | 1. Application’s auto-layout constraints are compatible with the iOS device’s screen resolution |
| Test Steps | 1. Open the application on an iOS device with iPhone 5s and above with screen resolutions of 1136 x 640 pixels, and iPad 5th Generation and above with screen resolutions of 2048 x 1536 pixels |
| Expected Result | 1. Application’s auto-layout constraints are compatible with the iOS device’s screen resolution |

# 

# 

# 2. Integration Testing

## 2.1 Offline Application

|  |  |
| --- | --- |
| **Test Case Name** | **Verify SQLCipher Decrypts Data** |
| **Test Case ID** | **TC - 81** |
| Priority | High |
| Preconditions | 1. Encrypted SQLite database has been created. 2. TC – 05, TC-16, TC-17, TC-18, TC19 has passed 3. Smith1 is logged in to MyHealthKeeper 4. Smith1 is on the Main page |
| Postconditions | 1. SQLCipher will decrypt the data stored in the database. |
| Test Steps | 1. Smith1 taps on Menu button on top-left corner of the Main page 2. Smith1 taps on View Medical 3. Smith1 is able to access their medical data |
| Expected results | 1. Smith1’s data is decrypted in the SQLite database. |

|  |  |
| --- | --- |
| **Test Case Name** | **AirPrint Integration** |
| **Test Case ID** | **TC - 82** |
| Priority | High |
| Preconditions | 1. Smith1 is logged in. 2. Smith1 is on the Main page 3. Smith1’s data is inserted into the encrypted SQLite database. 4. Smith1’s iOS device has an available WIFI connection |
| Postconditions | 1. UIPrintInteractionController is called 2. Printer Options page is displayed on Smith1’s iOS device |
| Test Steps | 1. Smith1 taps on ‘Menu’ on top-left corner of the Main page 2. Smith1 taps on ‘Print’ on the Menu 3. Smith1 is directed to ‘Print Report’ page 4. An alert appears with the following message: “Select an image to Print? Would you like to print an image?” 5. Smith1 selects ‘No’. 6. Smith1 performs one of the following on the Print Report page:    1. Smith1 taps on ‘Print Standard Report’ button    2. Smith1 taps on ‘Print Detailed Report’’ button |
| Expected Results | 1. Smith1 is directed to the Printer Options page with their selected report displayed in the view. |

# 

## 2.2 Online Application

|  |  |
| --- | --- |
| **Test Case Name** | **Amazon Web Services - Server Connection** |
| **Test Case ID** | **TC - 83** |
| Priority | High |
| Preconditions | 1. EC2 instance on AWS Server and RDS is setup and configured 2. Smith1’s iOS device has internet connection 3. TC - 01 has passed |
| Postconditions | 1. As long as the application is connected to the server, Smith1 should be able to login to the application without experiencing any connection errors/issues. |
| Test Steps | 1. Smith1 opens MyHealthKeeper on their iOS device 2. Smith1 attempts to log in to the application |
| Expected Results | 1. Smith1 is able to login to MyHealthKeeper |

|  |  |
| --- | --- |
| **Test Case Name** | **Amazon Web Services - Sync Data** |
| **Test Case ID** | **TC - 84** |
| Priority | High |
| Preconditions | 1. Smith1 has two iOS devices with MyHealthKeeper installed 2. Smith1 is logged in to MyHealthKeeper on iOS device 1 3. Smith1’s iOS devices have an internet connection |
| Postconditions | 1. As long as the application is connected to the server, Smith1 should be able to sync their data without experiencing any connection errors/issues |
| Test Steps | 1. Smith1 taps on ‘Menu’ 2. Smith1 taps on ‘Sync’    1. Data is retrieved from the local database and POSTs data to the AWS server    2. Data is stored in MySQL database on the AWS server    3. Alert appears with a message stating “Sync Successful” 3. Smith1 taps on “Logout” 4. An alert appears with the following message: ”Unsaved Changes Would you like to sync your data before logging out?” 5. Smith1 selects ‘No’ on the alert. 6. Smith1 is logged out of iOS device 1. 7. Smith1 takes iOS device 2 and opens up application by tapping on MyHealthKeeper 8. Smith1 logs in by entering “Smith1” into ‘Username’ text field and “Smith1!” into ‘Password’ text field 9. Smith1 taps on ‘Menu’ 10. Smith1 taps on ‘Sync’     1. MyHealthKeeper GETs data from the server and is stored in the local database     2. Alert appears with a message stating “Sync Successful” |
| Expected Results | 1. Smith1 should be able to sync the data without any errors. |

# 

# 3. System Testing

## 3.1 Offline Application

|  |  |
| --- | --- |
| **Test Case Name** | **User Registration - Successful** |
| **Test Case ID** | **TC - 85** |
| Priority | High |
| Preconditions | 1. User downloaded application and opened it on their iOS device |
| Postconditions | 1. Smith1 is directed to the ‘Setup Security Questions’ page |
| Parameters | [First Name]: New User’s First Name  [Last Name]: New User’s Last Name  [Username]: New User’s Username  [Password]: New User’s Password  [Email: New]: User’s Email address  [Cellphone]: New User’s Cellphone number |
| Test Steps | 1. User taps ‘Click to Register’ button 2. User enters [First Name] in designated text field 3. User enters [Last Name] in designated text field 4. User enters [Username] containing the following format: 6-10 characters in length containing letters [A-Z,a-z] and numbers [0-9] in designated text field 5. User enters [Password] 6. User performs one of the following:    1. User enters [Email] containing the following format: letters, numbers and select special characters [A-Z,a-z,0-9,\_-.], an ‘@’ symbol, a ‘.’ symbol, and 2 or 3 letters [a-z] in designated text field    2. User enters [Cellphone] containing the following format:10 numbers [0-9], without parentheses or dashes, in the designated text field 7. User taps ‘Next Page’ button |
| Expected Results | 1. User’s Registration information is saved in the database’s ‘Registration’ table 2. Smith1 is directed to the ‘Setup Security Questions’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **User Login - Successful** |
| **Test Case ID** | **TC - 86** |
| Priority | High |
| Preconditions | 1. User has successfully registered and set up Security Questions |
| Postconditions | 1. Smith1 is directed to the Main page |
| Parameters | [Username]: Smith1’s Username  [Password] Smith1’s Password |
| Test Steps | 1. Smith1 enters [Username] according to the constraints in Step 4 of TC-1 in designated text field 2. Smith1 enters [Password]according to the constraints in Step 5 of TC-1 in designated text field 3. Smith1 taps ‘Sign In’ button |
| Expected Results | 1. Smith1’s username and password are compared and matched to the ‘username’ and ‘password’ columns of the database’s ‘Registration’ table 2. Smith1 is directed to the Main page |

|  |  |
| --- | --- |
| **Test Case Name** | **View Medical Summary** |
| **Test Case ID** | **TC - 87** |
| Priority | High |
| Preconditions | 1. Smith1 is logged in successfully. |
| Postconditions | 1. Smith1 is on the ‘Medical Summary’ page 2. Smith1’s Medical Information is displayed on the ‘Medical Summary’ page |
| Parameters | [Last Name]: Smith1’s Last Name  [First Name]: Smith1’s First Name  [Date of Birth]: Smith1’s Date of Birth  [Gender]: Smith1’s Gender  [Street]: Smith1’s Address - Street  [City]: Smith1’s Address - City  [State]: Smith1’s Address - State  [Zip Code]: Smith1’s Address - Zip Code  [Doctor’s Name]: Smith1’s Doctor’s Name  [Doctor’s Specialty]: Smith1’s Doctor’s Specialty  [Doctor’s Address]: Smith1’s Doctor’s Office Address  [Illness/Disease Name]: Smith1’s Illness/Disease Name  [Medication Name]: Smith1’s Medication Name  [Dosage]: Smith1’s Medication Dosage  [Status]: Smith1’s Medication Status  [Surgery Name]: Smith1’s Surgery Name  [Surgery Date]: Smith1’s Surgery Date  [Surgery Description]: Smith1’s Surgery Description  [Allergy Name]: Smith1’s Allergy Name  [Allergy Treatment]: Smith1’s Allergy Treatment  [Allergy Medication]: Smith1’s Allergy Medication  [Vaccine Name]: Smith1’s Vaccine Name  [Vaccine Date]: Smith1’s Vaccine Administered Date  [Family History]: Smith1’s Family History  [Note]: Smith1’s Additional Notes  [insurance Type]: Smith1’s Insurance Type  [Insurance Name]: Smith1’s Insurance Name  [Group ID]: Smith1’s Insurance Group ID  [Expiration Date]: Smith1’s Insurance Expiration Date |
| Test Steps | 1. Smith1 taps “Save and Finish’ on Insurance Information page 2. An alert appears on the screen asking Smith1 to Upload a Document 3. Smith1 taps ‘No’ 4. Smith1 Medical Summary will include [Last Name],[First Name],[DOB],[Gender],[Street],[City],[Zipcode],[State],[Doctor’s Name] [Doctor’s Specialty], [Doctor’s Address],[ Illness/Disease Name],[Medication Name],[Dosage],[Status],[Surgery Name],[Surgery Description],[Surgery Date],[Allergy Name],[Allergy Treatment],[Allergy Medication],[vaccine Name],[Vaccine Date],[Family History],[Note],[insurance Type],[Insurance Name], [Group ID],[Expiration Date] |
| Expected Results | 1. The alert disappears from the screen 2. Smith1 is directed to Medical Summary page 3. Smith1’s Medical Information is displayed on the ‘Medical Summary’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **View Uploaded Documents Summary** |
| **Test Case ID** | **TC - 88** |
| Priority | High |
| Preconditions | 1. Follow steps for TC - 34 2. Smith1 is on the ‘Upload Documents’ page |
| Postconditions | 1. Images and its corresponding name are displayed |
| Parameters | [Image]: Uploaded Image  [Name]: Text entered for document name  [Description]: Text entered for document description |
| Test Steps | 1. Smith1 taps ‘OK’ button 2. Smith1 is directed to ‘Document Summary’ page |
| Expected Results | 1. Uploaded documents are displayed from oldest - newest 2. Summary page includes [Image] with its corresponding [Name] and [Description] to the right of the [Image] |

|  |  |
| --- | --- |
| **Test Case Name** | **View Reminder Summary** |
| **Test Case ID** | **TC - 89** |
| Priority | Medium |
| Preconditions | 1. TC - 05 has passed. 2. Smith1 is on the ‘Main’ page 3. TC-48 is has passed |
| Postconditions | 1. Reminder Name and Reminder Time for each of the Smith1’s reminders are displayed on the ‘View Reminders’ page |
| Parameters | [ReminderName]: name of the reminder  [ReminderTime]: time the reminder is set to go off |
| Test Steps | 1. Smith1 taps the Menu button 2. Smith1 taps ‘Reminder Summary’ cell on Menu 3. Smith1 is redirected to the ‘Reminder Summary’ page 4. Smith1’s Reminder Summary will include: [ReminderName] and [ReminderTime] for each reminder. |
| Expected Results | 1. Reminder Name and Reminder Time for each of the Smith1’s reminders are displayed on the ‘View Reminders’ page |

|  |  |
| --- | --- |
| **Test Case Name** | **Monthly Reminder** |
| **Test Case ID** | **TC - 90** |
| Priority | Medium |
| Preconditions | 1. Smith1 has enabled the application to send notifications to their iOS device. 2. The time is 12:00 pm on the first day of the month. |
| Postconditions | 1. The Smith1 will receive a reminder notification with message: “Please update your medical information on myHealthKeeper”, prompting them to update their medical information. |
| Parameters | [Time]: First of month at 12:00 pm |
| Test Steps | 1. The notification center sends out the monthly notification at [Time]. |
| Expected Results | 1. Smith1 will receive a reminder notification with message: ”Please update your medical information on myHealthKeeper”, prompting them to update their medical information. |

## 3.2 Online Application

|  |  |
| --- | --- |
| **Test Case Name** | **Retrieve Data from Server** |
| **Test Case ID** | **TC - 91** |
| Priority | Medium |
| Preconditions | 1. Smith1’s iOS device has MyHealthKeeper installed 2. Smith1 is logged in the MyHealthKeeper application |
| Postconditions | 1. Data synced to Amazon AWS is available on the Smith1’s iOS device |
| Parameter | [Summary]: Medical Summary Page, Uploaded Documents Summary Page, Reminders Summary Page |
| Test Steps | 1. Smith1 taps on ‘Menu’ 2. Smith1 taps on a [Summary] page |
| Expected Results | 1. Smith1 is redirected to the [Summary] 2. All data for the specific [Summary] page is displayed |

# 

# 

# Appendix

## Database Scripts

|  |  |
| --- | --- |
| **Script ID** | **Data Description** |
| DBS - 00 | Empty entire database |
| DBS - 01 | Create database to store Registration Information  "Create Table Registration ('pFirstName' text not null, 'pLastName' text not null, 'pUsername' text not null, 'pPassword' text not null, 'pEmail' text null, 'pCellPhone') text null);" |
| DBS - 02 | Insert User’s Registration Information into the Registration table  "Insert into Registration ('pFirstName', 'pLastName', 'pUsername', 'pPassword','pEmail','pCellPhone') values (“John”, “Smith”, “‘Smith1’”, “‘Smith1!”, “‘john.smith@gmail.com”, “”); |
| DBS - 03 | Create database to store Smith1’s Security Questions and Answers  "Create Table Security ‘rowID’ integer primary key not null, ('pQuestion1' text not null, 'pQuestion2' text not null, 'pQuestion3' text not null, 'pAnswer1' text not null, 'pAnswer2' text not null, 'pAnswer3' text not null, 'pUser' text not null);" |
| DBS - 04 | Insert Smith1’s Security Questions and Answers into the Security table  "Insert into Security (‘rowID’, 'pQuestion1', 'pQuestion2', 'pQuestion3', 'pAnswer1','pAnswer2','pAnswer3', 'pUser') values (1, “What was the name of your first pet?”,”What company did you hold your first job?”,”What was your first car?”,”Leo”,”Starbucks”,”Honda Civic”,”Smith1”); |
| DBS - 05 | Retrieve Smith1’s password from the Registration table for successful login  "select ('pUser') from Registration where ('pUser')=? " |
| DBS - 06 | Retrieve Smith1’s Security Questions from the Security table  "select ('pAnswer1') from Security where ('pUser')=? "  "select ('pAnswer2') from Security where ('pUser')=? "  "select ('pAnswer3') from Security where ('pUser')=? " |
| DBS - 07 | Updates Smith1’s password in the Registration table  “update Registration set 'pPassword'=? where ‘pUsername’=? ;" |
| DBS - 08 | Create database to store Smith1’s Personal Information  “Create table PersonalInformation ('LastName' text not null, 'FirstName' text not null, 'DateOfBirth' text not null, 'Gender' text not null, 'Street' text not null, 'City' text not null, 'ZipCode' text not null, 'State' text not null, 'SameUser' text)” |
| DBS - 09 | Insert Smith1’s Personal Information into the ‘PersonalInformation’ table  “insert into PersonalInformation ('LastName', 'FirstName', 'DateOfBirth', 'Gender','Street','City','ZipCode','State','SameUser') values (“Smith”,”John”,”12/20/1998”,”Male”,”123 Main”,”Detroit”,”48202”,”Michigan”,”Smith1”)” |
| DBS - 10 | Create database to store Smith1’s Doctor Information  “Create table Doctor('rowID' integer primary key not null, 'DoctorName' text not null ,’DoctorSpeciality’ text, ‘Doctor’sAddress’ text, 'Doctorcontact' text, 'SameUser' text)" |
| DBS - 11 | Insert Smith1’s doctor information into ‘Doctor’ table  “insert into Doctor('rowID', 'DoctorName','DoctorSpeciallity','DoctorAddress','Doctorcontact','SameUser') values  (1, “Dr. Amrit”, ”Pediatrician”, “123 Baker, Detroit, 3131111111”, ”Smith1")” |
| DBS - 12 | Create database to store Smith1’s Illness/Disease Information  “Create table Illnesses(‘rowID’ integer primary key not null, 'illnesseName' text not null, 'SameUser' text)” |
| DBS - 13 | Insert Smith1’s illness/disease information into ‘Illnesses’ table  “insert into Illnesses(‘rowID’, 'illnessName', 'SameUser') values (“Heart Failure”,”Smith1”)” |
| DBS - 14 | Create database to store Smith1’s Medication Information  “Create table Medicinelist (‘medId’ integer primary key not null, ‘MedName’ text not null,’dosage’ text not null,’status’ text not null, ‘username’ text )” |
| DBS - 15 | Insert Smith1’s medication information into ‘Medicinelist’ table  “insert into Medicinelist('MedName','dose','status','SameUser') values (“Amoxicilline”,”500 mg”, “Past”, “Smith1”)” |
| DBS - 16 | Create database to store Smith1’s Surgery Information  “Create table Surgery ('rowID' integer primary key not null, 'SurgeryName' text not null, ’SurgeryDate’ text, ‘SurgeryDescription’, 'sameuser' text)” |
| DBS - 17 | Insert Smith1’s surgery information into ‘Surgery’ table  “insert into Surgery('rowID', 'SurgeryName', 'Surgerydate','SurgeryDescription', 'sameuser') values  (1, “Kidney transplant”, “12/12/2005”, “Doing well now”, “Smith1”)” |
| DBS - 18 | Create database to store Smith1’s allergy information  “Create table Allergies('rowID' integer primary key not null, 'allergiesName' text not null, ’treatment’ text medication’ text , 'sameuser' text)” |
| DBS - 19 | Insert Smith1’s allergy information into ‘Allergies’ table  “insert into Allergies('rowID', 'allergiesName',’treatment’, ‘medication’, 'sameuser') values (1, “Peanuts”, “None”, “None”,”Smith1”)” |
| DBS - 20 | Create database to store Smith1’s vaccine information  “Create table Vaccines('rowID' integer primary key not null, 'vaccinesName', text not null,’vaccineDate’ text, 'sameuser' text)” |
| DBS - 21 | Insert Smith1’s vaccine information into ‘Vaccines’ table  “insert into Vaccines('rowID', 'vaccinesName',’vaccineDate’, 'sameuser') values (1,”Hepatitis B”,”12/12/2014”, “Smith1”)” |
| DBS - 22 | Create database to store Smith1’s additional information such as family medical history and Note  “Create table MedicalInformation ('Family\_History' text not null,'Note' text not null, 'SameUser' text)” |
| DBS - 23 | Insert Smith1’s additional information into ‘MedicalInformation’ table  “insert into MedicalInformation('Family\_History','Note','SameUser') values (“No blood related Illnesses”, “No additional notes”,”Smith1”)” |
| DBS - 24 | Create database to store Smith1’s medical insurance information  “Create table InsuranceInformation ('Insurance\_Type' text not null,'Insurance\_Name' text not null, 'Member\_ID' text not null, ’Expiration\_Date', 'SameUser' text)” |
| DBS - 25 | Insert Smith1’s medical insurance information into ‘InsuranceInformation’ table  “insert into “InsuranceInformation('Insurance\_Type','Insurance\_Name','Member\_ID',’Expiration\_Date','SameUser') values (“Medicaid”, “Blue Cross”, “ABCXYZ123”,”12/30/2017”,”Smith1”)” |
| DBS - 26 | Create database to store Smith1’s document information  “Create table Document (‘rowID’ integer primary key not null, ‘docName’ text not null, ‘docDescription’ text not null, ‘docImage’ text not null, ‘sameuser’ text)” |
| DBS - 27 | Insert Smith1’s Document Name and Document Description into the ‘Document’ Table  “Insert into Document(‘rowID’, ‘docName’, ‘docDescription, ‘‘sameuser’) values (1, “Test Document Name”, “Test Document Description”, “username1”)” |
| DBS - 28 | Deletes the first uploaded document  “Delete from Document WHERE rowID = 1” |
| DBS - 29 | Create database to store Smith1’s reminder information  “Create table reminder('reminderID' integer primary key not null, 'reminderName' text not null, 'reminderLocation' text not null, 'reminderReason text not null,'reminderDate text not null,’reminderUser’ test)” |
| DBS - 30 | Insert Smith1’s reminder information into ‘Reminder’ table  “Insert into reminder ('reminderID', 'reminderName', 'reminderLocation', 'reminderReason','reminderDate, ‘reminderUser’) values (1,'Test Reminder','TestLocation','ReminderReason','12/12/2017 1:20 pm’, ‘Smith1’’)” |
| DBS - 31 | Creates a table to store Smith1’s monthly Reminder information  “create table reminderMonthly (‘reminderUser’ text primary key not null, 'reminderStatus' bool not null )” |
| DBS - 32 | Turns on monthly reminder in database:  “update reminderMonthly set 'reminderStatus'=1 where ‘reminderUser’=”Smith1” |
| DBS - 33 | Turns off monthly reminder in database:  “update reminderMonthly set 'reminderStatus'=0 where ‘reminderUser’=”Smith1”” |
| DBS - 34 | Insert Smith1’s monthly reminder information into ‘Monthly Reminder’ table  insert into reminderMonthly ('reminderUser','reminderStatus') values (Smith1)',’1') |
| DBS -35 | Updates Reminder Information  Update reminder set ‘reminderName’=TestName1 ‘reminderLocation’=TestLocation2 ‘reminderReason’=TestReason2 ‘reminderDate’=12/12/20171:20pm where `reminderUser` =’Smith1’ |
| DBS-36 | Deletes Reminder information  “DELETE FROM reminder where reminderID = 1” |
| DBS-37 | "select \* from PersonalInformation where SameUser = Smith1" |
| DBS-38 | "select \* from Doctor where SameUser = Smith1" |
| DBS-39 | "select \* from Illnesses where SameUser = Smith1" |
| DBS-40 | "select \* from Medicinelist where SameUser = Smith1" |
| DBS-41 | "select \* from Surgery where sameuser = Smith1" |
| DBS-42 | "select \* from Allergies where sameuser = Smith1" |
| DBS-43 | "select \* from Vaccines where sameuser = Smith1" |
| DBS-44 | "select \* from MedicalInformation where SameUser = Smith1" |
| DBS-45 | "select \* from InsuranceInformation where SameUser = Smith1" |
| DBS-46 | "select \* from Document where docUser = Smith1 |
| DBS-47 | "select \* from reminder where reminderUser = Smith1 |
| DBS-48 | update Doctor set ‘DoctorName’=Dr.Misra , ‘DoctorSpeciallity’=Pediatrician, ‘DoctorAddress’=123 Baker, Detroit 48202, ‘Doctorcontact’  =3131111111 where rowID=1 ;" |
| DBS-49 | update Illnesses set ‘illnesseName’= Heart Attack where rowID=1 ; |
| DBS-50 | update Medicinelist set ‘MedName’= plavix , ‘dose’= 500mg, ‘status’  = past where rowID= 1;" |
| DBS-51 | "update Surgery set ‘SurgeryName’ = Brain Transplant , ‘Surgerydate’  = 12/12/2005, ‘SurgeryDescription’= doing well now where rowID=1 |
| DBS-52 | update Allergies set ‘allergiesName’ =Dust , ‘allergiesmedi’= none, ‘allergiestreatment’ =None where rowID=1 ;" |
| DBS-53 | "update Vaccines set ‘vaccinesName’ =Hepatites A, ‘vaccinesdate  ‘= 12/12/2014 where rowID=1 ;" |

## Traceability Matrix

The following traceability matrix will be used to trace each requirement throughout the mobile application development. Each requirement will have use cases and test cases associated with it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID** | **Name** | **Priority** | **Use Case(s)** | **Test Case(s)** |
| FR1 | Registration | High | UC-1 | TC-01, TC-02, TC-85 |
| FR2 | Setup Security Questions | Medium | UC-2 | TC-03, TC-04 |
| FR3 | Login to Application | High | UC-3 | TC-05, TC-06, TC-86 |
| FR4 | Forgot Password | Medium | UC-4 | TC-07, TC-08, TC-09, TC-10, TC-11 |
| FR5 | Enter Medical Information | High | UC-5, UC-6, UC-7, UC-8, UC-9, UC-10, UC-11, UC-12, UC-13 | TC-12, TC-13, TC-14, TC-15, TC-16 , TC-17, TC-18, TC-19, TC-20 |
| FR6 | Update Medical Information | Low | UC-21 | TC-27, TC-28, TC-29, TC-30, TC-31, TC-32 |
| FR7 | Delete Medical Information | High | UC-15, UC-16, UC-17, UC-18, UC-19, UC-20 | TC-21, TC-22, TC-23, TC-24, TC-25, TC-26 |
| FR8 | Upload Photo through Camera | High | UC-22 | TC-34, TC-35 |
| FR9 | Upload Photo through Photo Library/Gallery | High | UC-23 | TC-36, TC-37 |
| FR10 | Save Document | High | UC-24 | TC-38, TC-39, TC-40, TC-41 |
| FR11 | View Uploaded Documents | High | UC-25 | TC-42, TC-43, TC-44, TC-45, TC-88 |
| FR12 | Delete Documents | High | UC-26 | TC-46, TC-47 |
| FR13 | Add Reminder | Medium | UC-30 | TC-48, TC-49, TC-50 |
| FR14 | Monthly Reminder | Medium | UC-27,UC-28, UC-29 | TC-57, TC-58  TC-59, TC-90 |
| FR15 | View Reminder(s) | Medium | UC-31 | TC-55, TC-56 TC-89 |
| FR16 | Edit Reminder(s) | Medium | UC-32 | TC-51, TC-52, TC-53 |
| FR17 | Delete Reminder(s) | Medium | UC-33 | TC-54 |
| FR18 | View Full Medical History | High | UC-14 | TC-33, TC-87 |
| FR19 | Export Data to PDF | Low | UC-38 | TC-67 |
| FR20 | Printing Report | Medium | UC-34 | TC-60, TC-61 TC-82 |
| FR21 | Connecting iOS Devices | Medium | UC-37 | TC-66 |
| FR22 | Sync Data to Server | Medium | UC-36 | TC-65, TC-83, TC-84, TC-91 |
| FR23 | Menu | High | UC-39 | TC-62, TC-63 |
| FR24 | Logout | High | UC-35 | TC-64 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Non-Functional Requirements** | **Name** | **Priority** | **Test Case(s)** |
| NFR3.3.1.1 | Database Performance | Medium | TC-68 |
| NFR3.3.1.2 | File Upload Time | Medium | TC-69 |
| NFR3.3.2.1 | Database Error Handling | High | TC-70 |
| NFR3.3.3.1 | Offline Application Use Availability | High | TC-71 |
| NFR3.3.3.2 | Online Application Use Availability | Medium | TC-72 |
| NFR3.3.4.1 | Local SQLite Database for Offline Application | High | TC-73 |
| NFR3.3.4.2 | SQLite Data Encryption | High | TC-74, TC-81 |
| NFR3.3.4.3 | HTTPS for Online Application | Medium | TC-75 |
| NFR3.3.4.4 | User Authentication | High | TC-76 |
| NFR3.3.5.1 | Server Application Maintenance Time | Medium | TC-77 |
| NFR3.3.5.2 | Addition/Updates to Application Features | Medium | TC-78 |
| NFR3.3.6.1 | iOS Compatibility | High | TC-79 |
| NFR3.3.6.2 | Screen Resolution | High | TC-80 |

## Key Terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | An individual interacting with the mobile application |
| HIPAA | Stands for Health Insurance Portability and Accountability Act. Provides data privacy and security for patients’ medical records. |
| Physician | Healthcare professional trained and licensed to practice medicine |
| Pediatrician | Doctor who specializes in the medical care of children |
| Patient | An individual who is receiving medical treatment or is under the care of a physician. |
| Transition Period | Time period between ages 16 - 25 where patients move from a pediatrician to an adult physician |
| iOS | Mobile operating system developed by Apple |
| EKG | Stands for Electrocardiogram. Records electrical activity of the heart. |
| Chronic Medical Disease | Disease that lasts three months or more. Mostly cannot be cured through medication or prevented through vaccination. |
| GUI | Graphical User Interface allows a user to interact with an electronic device through features like windows, icons, and buttons. |
| Server | To process requests and deliver data to other computers (clients) over a local network or the internet. |
| Bluetooth | A wireless technology to exchange data over short distances. |
| FMDB | Flying Meat Database |
| HTTPS (Hyper Text Transfer Protocol Secure) | Encrypted communication between application and server. |
| Objective-C | Objective Oriented Programming Language used by Apple for the OS X and iOS operating systems. |

## References

[1] Apple, “Official Apple Support” support.apple.com/kb/SP751?locale=en\_US.