Test plan

*Thomas Or*

June 23, 2020

Table of Contents

[1 Introduction 2](#_Toc529369248)

[1.1 Objectives 2](#_Toc529369249)

[1.2 Team members 2](#_Toc529369250)

[2 Features/units to be tested 2](#_Toc529369251)

[3 Approach 3](#_Toc529369252)

[4 Item pass/fail criteria 3](#_Toc529369253)

[5 Test cases 4](#_Toc529369254)

[5.1 Test Case ID: 001 4](#_Toc529369255)

[5.2 Test Case ID: 002 5](#_Toc529369256)

[5.3 Test Case ID: 003 6](#_Toc529369257)

[5.4 Test Case ID: 004 7](#_Toc529369258)

[5.5 Test Case ID: 005 9](#_Toc529369259)

[6 Schedule 9](#_Toc529369260)

[6.1 Test Schedule 9](#_Toc529369261)

[6.2 Deliverables 10](#_Toc529369262)

[7 Risks 10](#_Toc529369263)

# Introduction

This test plan outlines the testing requirements for the Android Phonebook Application. In particular, the SQLite/Database functionality of the application.

## Objectives

Objective is to test the SQLite/Local Database functionality of app.

## Team members

|  |  |
| --- | --- |
| **Name** | **Role** |
| *Qiao Li*  *Thomas Or* | Test Manager  Developer, Tester |

# Features/units to be tested

Features to be tested:

* Add a contact to the database
* Update a contact in the database
* Delete a contact from the database
* Get all contacts from the database
* Retrieve a Contact by ID

# Approach

The Testing Approach will use Black-Box testing. A user will run the application and test the features by following the Test cases outlined below.

# Test cases

## Test Case ID: 001

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID**: 001  **Executed by**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Execution Date**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Scenario**: Run the App and click the Add floating action button to get to the Add Page  **Test Case**: Add a contact to the Database | | | | |
| Prerequisite | | Able to run phonebook app on Android device or emulator | | |
| Test Data | | *Name: TestData*  *Phone: 0411111111*  *Email:* [*Test@test.com*](mailto:Test@test.com)  *Date: 23/06/2020* | | |
| Step | Action | | Expected Result | Success? |
| 1 | Perform swipe gesture on Welcome Page | | The list page loads successfully. | **or X** |
| 2 | On the List page, Click the ‘Add’ floating action button | | The add page loads successfully. |  |
| 3 | Enter data into the Text Fields | | Able to enter data: Name, Phone, Email and Date. Can add the Test Data listed above. |  |
| 4 | Click the ‘Add’ button | | The contact object is added to the database and the user is returned to the List Page. |  |
| 5 | In Android Studio, Open the Device File Explorer and save the application’s .db files and open with SQLiteDatabaseBrowser | | The contact has successfully been added to the database and can be verified with the Android Database Management tool, “SQLiteDatabaseBrowser”. |  |

## Test Case ID: 002

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID**: 002  **Executed by**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Execution Date**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Scenario**: Run the App and Update a Contact by Swiping a Contact and editing a Contact on the Edit Page  **Test Case**: Update a contact in the Database | | | | |
| Prerequisite | | Able to run phonebook app on Android device or emulator. | | |
| Test Data | | *Name: TestData*  *Phone: 0411111111*  *Email:* [*Test@test.com*](mailto:Test@test.com)  *Date: 23/06/2020* | | |
| Step | Action | | Expected Result | Success? |
| 1 | Perform swipe gesture on Welcome Page | | The list page loads successfully. | **or X** |
| 2 | On the List page, Swipe the first Contact to the Left and Click the Edit Button | | The Edit page loads successfully. The Page loads with the Contact’s data pre-filled in the Text Fields |  |
| 3 | Edit the data of the selected the Contact, change the data. | | Able to edit the data: Name, Phone, Email and Date |  |
| 4 | Click the ‘Update’ button | | The contact object is updated in the database and the user is returned to the List Page. |  |
|  | In Android Studio, Open the Device File Explorer and save the application’s .db files and open with SQLiteDatabaseBrowser | | The contact has successfully been updated in the database and can be verified with the Android Database Management tool, “SQLiteDatabaseBrowser”. |  |

## Test Case ID: 003

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID**: 003  **Executed by**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Execution Date**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Scenario**: Run the App and Delete a Contact by Long Click on a Contact and Dragging the Contact onto the Delete Button  **Test Case**: Update a contact in the Database | | | | |
| Prerequisite | | Able to run phonebook app on Android device or emulator. | | |
| Test Data | |  | | |
| Step | Action | | Expected Result | Success? |
| 1 | Perform swipe gesture on Welcome Page | | The list page loads successfully. | **or X** |
| 2 | On the List page, Long Click a Contact and drag the shadow onto the delete button | | After the contact has been dragged onto the delete button, a message will appear, confirming if the user wishes to delete the contact |  |
| 3 | Confirm the deletion of the Contact by select YES | | The Contact is successfully deleted, with a confirmation Toast Message saying the contact has been deleted. The deleted contact will also be removed from the Contact List. |  |
| 4. | In Android Studio, Open the Device File Explorer and save the application’s .db files and open with SQLiteDatabaseBrowser | | The contact has successfully been Deleted from the database and can be verified with the Android Database Management tool, “SQLiteDatabaseBrowser”. |  |

## Test Case ID: 004

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID**: 004  **Executed by**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Execution Date**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Scenario**: Run the App and View the List Page, Verify if the Contact in the List page matches the data in the Database  **Test Case**: Get all contacts from the Database | | | | |
| Prerequisite | | Able to run phonebook app on Android device or emulator. | | |
| Test Data | |  | | |
| Step | Action | | Expected Result | Success? |
| 1 | Perform swipe gesture on Welcome Page | | The list page loads successfully. | **or X** |
| 2 | Verify if the Contact List matches the data in the database | | By viewing the Contact List in the app and the Contact table in the database via SQLiteDatabaseBrowser, the data in the app and database are the same |  |
| 3. | Make a change to the data (e.g. Add another Contact) and then close the Application | | Able to Add a contact and the change can be verified in the database |  |
| 4. | Re-open the Application a verify if the data still matches the database | | The database data will still match the data seen on the List Page |  |

## Test Case ID: 005

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID**: 005  **Executed by**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Execution Date**: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Scenario**: Run the App and Update a Contact by Swiping a Contact and view the Contact on the Edit Page  **Test Case**: Retrieve a Contact by ID | | | | |
| Prerequisite | | Able to run phonebook app on Android device or emulator. | | |
| Test Data | |  | | |
| Step | Action | | Expected Result | Success? |
| 1 | Place a break point on getContactByID() function on the Edit Page and run the debugger. Attempt to Update a Contact | | The debugger will stop at the breakpoint. User can verify the value of the ID being passed in the getContactByID() function | **or X** |
| 2 | Continue the application and view the Contact retrieved by the function | | The Contact is displayed in the Edit Page |  |
| 3. | In Android Studio, Open the Device File Explorer and save the application’s .db files and open with SQLiteDatabaseBrowser.  Verify if the Contact retrieved was the correct contact i.e. the ID seen in Step1 matches the contact retrieved as seen in the database | | View the Contact data retrieved from the function and verify if the data retrieved matches with the ID passed through the function. The data retrieved by ID should match with the data in the database. |  |

# Schedule

## Test Schedule

|  |  |  |
| --- | --- | --- |
| **Task** | **Start** | **Finish** |
| Perform Testing with Test Plan | 23/6/2020 | 24/6/2020 |

## 

## Deliverables

|  |  |
| --- | --- |
| **Task** | **Date** |
| Test Plan | 23/6/2020 |
| Test Results |  |
| Test Report |  |