Aleph

FireAlertScanner
Software Testing Plan (TP)

Version <1.0>

Revision History

Date	Version	Description	Author
12/3/2013	1.0	Did everything	Nick

Table of Contents

Software Testing Plan (TP)

1. Introduction

This document outlines the procedure to be taken to test the FireAlertScanner system and all of its sub-components. Documented below you will find an overview of the system, features and sub-systems/sub-components that will be tested, and an overview of individual test cases.

2. Relationship to other documents

This document is meant to plan the testing of the implementation of the requirements detailed in the SRS and the design of the system detailed in the SDS. The individual test cases cover all of functional and non-functional requirements in the SRS and compare the actual implementation and the planned implementation found in the SDS.

3. System overview

The FireAlertScanner has been implemented in such a way that many of the functional and non-functional requirements are compacted into a small screen space to suit the device interface. In actuality the FireAlertScanner system is divided up into three main “activities” or “screens” that provide all the system functionality.

The testing documented below will take place in one of the three screens, focusing on the part of the screen that provides the functionality to be tested. The screens are connected sequentially; to reach the last screen, you must first traverse the first two. Testing that takes place on a specific screen will assume that the previous screens have been traversed, and do not affect the testing at hand. The only exception to this is if the testing to be done depends on the information passed between the two screens, then the testing will be affected by the previous screens traversed.

4. Features to be tested/not to be tested

Only system testing will be tested and documented in this document and all referenced documents as per the Lab-8Testing Doucments.pdf.

Both functional and non-functional requirements will be tested and documented. These functional and non-functional requirements are derived from the individual use-cases defined in the SRS. In most cases, the use-cases have been altered slightly to suit the testing situation and the actual implementation to provide accurate testing. In some cases, additional requirements are tested to cover requirements that became apparently necessary during implementation.

Changes to the requirements defined in the use-cases noted above will appear in the individual testing specifications.

5. Pass/Fail criteria

For a test to pass, the FireAlertScanner system must satisfy the requirements defined in the testing specification that are derived from the corresponding use-case. This means the input must be performed exactly as stated and the output must be exactly as the expected output.

For a test to fail, the FireAlertScanner must not provide the expected output for a test’s input.

6. Approach

Each use-case defined in the SRS will have a corresponding test. Each test will attempt to satisfy the requirements outlined in each use-case. Input for each test will be derived from the flow of events for its corresponding use-case as will excepted output.

This strategy was chosen for simplicity: use-cases define the functional cases for use and therefore describe ideal situations of functionality for the system. Deriving tests from the use-cases provide accurate determinants for the overall system success.

Some tests may deviate from this strategy, requirements that became apparently necessary will not be described in the use-cases. In these cases, test cases will be written to suit the specific situation as per direction of the implementation.

7. Suspension and resumption

If, in the case that changes to the system are made during testing activates, and they affect the testing at hand. The test will be suspended and repeated to reflect the most current system.

8. Testing materials (hardware/software requirements)

Testing will take place on the Winmate device. The Winmate device will be running a release build and will be running under normal circumstances. Any other special circumstances related to the functionality of the Winmate device, i.e. file structure etc. will be stated within the testing specification.

9. Test cases

Below you will find a list of tests to be performed and you will find the corresponding test case specifications either in its corresponding file or if you're reading this in the combined pdf, below.

Test – Authenticate User

Test - Display Client Locations

Test – Display List of Clients

Test – Get Code Manually

Test – Inspect Equipment

Test – Manage User Account

Test – Record Results

Test – Scan Equipment

Test – Send Results

Test – XML File Does not Exist

10. Testing schedule

Testing will be carried out by the developers on a schedule that fits into the overall project schedule.

SE 3350y - SOFTWARE ENGINEERING DESIGN I

Fire Alert Scanner
Test Case Specification (TCS)

Version <1.0>

Revision History

Date	Version	Description	Author
<01/12/13>	<1.0>	Entered all required info	Zach Hoggard

Table of Contents

1. Test case specification identifier

This test case shall be referred to as Get Code Manually, indicating that it covers the functionality of the Get Code Manually use case.

2. Test items

- XML file is not contained within the external storage (SD Card)

3. Input specifications

1. Log into Fire Alert Scanner.

4. Output specifications

1.1 If XML file is not in SD card, display prompt to user to inform of results.

1.2 If XML file is in SD card, Fire Alert Scanner will continue to load without displaying prompt.

5. Environmental needs

The device must have a file called Inspection_Data.xml in external storage with equipment data (Only for part of the test).

6. Special procedural requirements

N/A

7. Intercase dependencies

N/A

SE 3350y - SOFTWARE ENGINEERING DESIGN I

Fire Alert Scanner
Test Case Specification (TCS)

Version <1.0>

Revision History

Date	Version	Description	Author
<01/12/13>	<1.0>	Entered all required info	Zach Hoggard

Table of Contents

1. Test case specification identifier

This test case shall be referred to as Scan Equipment, indicating that it covers the functionality of the Scan Equipment use case.

2. Test items

- Scan equipment while not on right page in app.
- Scan equipment with invalid barcode
- Scan equipment with correct barcode

3. Input specifications

1. Attempt to scan equipment with when not on correct page in app.
2. Attempt to scan equipment not contained within Winmate device.
3. Attempt to scan equipment with correct barcode.

4. Output specifications

1.1 Nothing should happen.

2.1 Prompt is displayed indicating that barcode is incorrect.

3.1 Correct equipment list is expanded where the user can enter test results pertaining to the scanned equipment.

5. Environmental needs

The device must have a file called Inspection_Data.xml in the external storage which will contain all equipment data.

6. Special procedural requirements

N/A

7. Intercase dependencies

N/A

SE 3350y - SOFTWARE ENGINEERING DESIGN I

FireAlertScanner
Test Case Specification (TCS) – Record Results

Version 1.0

Revision History

Date	Version	Description	Author
02/Dec/2013	1.0	Document created	Benjamin Schubert

Table of Contents

Test Case Specification (TCS)

1. Test case specification identifier

Record Results

2. Test items

Make a change and switch rooms/floors
Make a change and press the save button
Make a change and press the back button

3. Input specifications

- Navigate to ScanActivity
- Expand any equipment item
- Select *Pass* or *Fail* for any inspection element
- Drop down the Room spinner and select a different room
- Press the *Yes* button of the displayed dialog
- Make another change to an inspection element
- Press the *Save* button
- Make another change to an inspection element
- Press the device's back button

4. Output specifications

1. Switching rooms/floors after a change
   1.1. A dialog is displayed, prompting the user to save changes, or finish the current room.
   1.2. The *InspectionData.xml* on the SD card reflects the changes made.
2. Pressing the save button
   2.1. The save button greys out and says "Saving…".
   2.2. The *InspectionData.xml* on the SD card reflects the changes made.
3. Pressing the back button after a change
   3.1. A dialog is displayed, prompting the user to save changes.
   3.2. The *InspectionData.xml* on the SD card reflects the changes made.
   3.3. The UI is navigated to the previous Activity.

5. Environmental needs

InspectionData.xml must be located in the external data storage to load the menus correctly.

6. Special procedural requirements

N/A

7. Intercase dependencies

N/A

SE 3350y - SOFTWARE ENGINEERING DESIGN I

Fire Alert Scanner
Test Case Specification (TCS)

Version <1.0>

Revision History

Date	Version	Description	Author
<01/12/13>	<1.0>	Updated all required entries	Thomas Trovato

Table of Contents

Test Case Specification (TCS)

1. Test case specification identifier

This test case shall be referred to as Manage User Accounts Test Case, to denote its relation to the Manage User Accounts use case.

2. Test items

- Alternate account authorization
- Writing new entry to user account collection

3. Input specifications

1. Click “Register User” button with username field empty (password is irrelevant)
2. Click “Register User” button with password field empty (username is irrelevant)
3. Click “Register User” button with username already existing in UserAccounts.txt (password is irrelevant)
4. Click “Register User” button with password less than 4 characters long (username is irrelevant)
5. Click “Register User” button with username that does not exist in UserAccounts.txt and a valid password
6. Enter an invalid admin key into the admin key field when prompted
7. Enter a valid admin key into the admin key field when prompted

4. Output specifications

1.1 Prompt is displayed indicating that the username field is required.

2.1 Prompt is displayed indicating that the password field is required.

3.1 Prompt is displayed indicating that the given username is already taken.

4.1 Prompt is displayed indicating that the password is invalid.

5.1 Prompt is opened asking for an administrator key to validate the registration.

6.1 Prompt is displayed indicating that the administrator key is invalid.

7.1 The given username and password are added to UserAccounts.txt, and the user proceeds to the DataEntryMain activity.

5. Environmental needs

N/A

6. Special procedural requirements

N/A

7. Intercase dependencies

N/A

SE 3350y - SOFTWARE ENGINEERING DESIGN I

Fire Alert Scanner
Test Case Specification (TCS)

Version <1.0>

Revision History

Date	Version	Description	Author
<01/12/13>	<1.0>	Entered all required info	Zach Hoggard

Table of Contents

1. Test case specification identifier

This test case shall be referred to as Get Code Manually, indicating that it covers the functionality of the Get Code Manually use case.

2. Test items

- No code entered.
- Wrong code/nonexistent code entered (test prompt to user).
- Code entered does not contain letters.
- Correct code entered (test expanding list process).

3. Input specifications

1. Attempt to find equipment without inputting barcode. The field is left blank.
2. Attempt to input barcode that does not correspond to any barcode contained in Winmate device.
3. Attempt to input barcode containing just letters.
4. Attempt to input barcode that contains both letters an numbers.
5. Attempt input barcode that corresponds to an existing piece of equipment.

4. Output specifications

1.1 Prompt is displayed indicating that barcode was not entered.

2.1 Prompt is displayed indicating that equipment does not exist.

3.1 Prompt is displayed indicating that the barcode is incorrect.

4.1 Prompt is displayed indicating that the barcode is incorrect.

5.1 Barcode entry is successful; the correct equipment list is expanded and ready to test results.

5. Environmental needs

The device must have a file called Inspection_Data.xml in external storage with equipment data.

6. Special procedural requirements

N/A

7. Intercase dependencies

N/A

SE 3350y - SOFTWARE ENGINEERING DESIGN I

Fire Alert Scanner
Test Case Specification (TCS)

Version <1.0>

Revision History

Date	Version	Description	Author
<01/12/13>	<1.0>	Entered all required info	Thomas Trovato

Table of Contents

Test Case Specification (TCS)

1. Test case specification identifier

This test case shall be referred to as Authenticate User Test, indicating that it covers the functionality of the Authenticate User use case.

2. Test items

- Login with username that does not exist (test prompt to user)
- Login with incorrect username/password combination (test prompt to user)
- Login without password
- Login with correct credentials (test sign-in process)

3. Input specifications

1. Attempt login with username not contained within the mobile device’s “UserAccounts.txt” (password is irrelevant)
2. Attempt login with username field empty (password is irrelevant).
3. Attempt login with password field empty (username is irrelevant).
4. Attempt login with username contained within UserAccounts.txt, but with an incorrect password
5. Attempt login with username contained within UserAccounts.txt and its correct password

4. Output specifications

1.1 Prompt is displayed indicating that the user account does not exist.

2.1 Prompt is displayed indicating that the username field is required to have a value.

3.1 Prompt is displayed indicating that the password field is required to have a value.

4.1 Prompt is displayed indicating that the password field’s value is incorrect for the given account.

5.1 Sign-in is successful, and the user proceeds to the MainDataEntry activity.

5. Environmental needs

The device must have a file called UserAccounts.txt in the private internal storage with sample user accounts, stored in the format “*hashedUsername:*:*hashedPassword*,*hashedUsername2*” etc.

6. Special procedural requirements

N/A

7. Intercase dependencies

N/A

SE 3350y - SOFTWARE ENGINEERING DESIGN I

FireAlertScanner
Test Case Specification (TCS)

Version 1.0

Revision History

Date	Version	Description	Author
02/Dec/2013	1.0	Completed form	Benjamin Schubert

Table of Contents

Test Case Specification (TCS)

1. Test case specification identifier

Send results

2. Test items

Send with incorrect port/IP address
Send with correct credentials

3. Input specifications

Navigate to the MainDataEntry activity
Press the *Send Results* button
Enter an incorrect port
Press *Ok* button
Enter an incorrect IP
Press *Ok* button
Enter correct port and IP
Press *Ok* button

4. Output specifications

1. Entering an incorrect port
   - 1.1. Connection times out
   - 1.2. A Toast indicates that the results were not sent
2. Entering an incorrect IP address
   - 2.1. Connection times out
   - 2.2. A Toast indicates that the results were not sent
3. Entering correct credentials
   - 3.1. *InspectionData.xml* is sent over TCP to the specified recipient
   - 3.2. A Toast indicates that the results were sent successfully

5. Environmental needs

- *InspectionData.xml* must be located on the external storage
- A server must be listening for the file to be received

6. Special procedural requirements

N/A

7. Intercase dependencies

N/A

Aleph

Inspect Equipment
Test Case Specification (TCS)

Version <1.0>

Revision History

Date	Version	Description	Author
12/3/2013	1.0	Did everything	Nicholas Ward

Table of Contents

Test Case Specification (TCS)

1. Test case specification identifier

Inspect Equipment

2. Test items

The test input was designed to test these items, but also sub-components of each items. Note that stress testing is required to guarantee complete testing of each item and its sub-components.

Changing Floors

Changing Rooms

Inputting inspection values

Completing Equipment Inspections

Completing Rooms

3. Input specifications

These were written to be sequential and should be performed as such. The outputs may not match if you do not perform the entire input atomically.

1. Launch the activity
2. Change Floors with the floor spinner
3. Change Rooms with the room spinner
4. Open a piece of equipment
5. Select a Pass or Yes Radio Button
6. Select a Fail or No Radio Button
7. Fill out everything else in the piece of equipment with random inputs.
8. Fill out every other piece of equipment
9. Change Rooms with the room spinner
10. Fill out all equipment
11. Click Save Button
12. Change Rooms with the room spinner

4. Output specifications

These outputs should be matched to the corresponding input in section 3: Input Specifications.

1. Activity Launched
    - 1.1. Populates Floor spinner
    - 1.2. Populates Room spinner
    - 1.3. Populates ExpandableList

2. Floor spinner changed

   2.1. Pop up saying you haven’t finished the floor. (assume you click to continue anyways)

   2.2. Populates Room Spinner

   2.3. Populates ExpandableList

3. Room spinner changed

   3.1. Pop up saying you haven’t finished the floor. (assume you click to continue anyways)

   3.2. Populates ExpandableList

4. Expanded piece of Equipment form List

   4.1. Populates child inspection elements

5. Selected Pass or Yes Radio Button

   5.1. Inspection Element name turns green

   5.2. Inspection Element name has asterisk next to it

   5.3. Equipment name has asterisk next to it

6. Select a Fail or No Radio Button

   6.1. Pop up asking for reason for fail (assume you enter some text then click ok)

   6.2. Inspection Element name turns green

   6.3. Inspection Element name has asterisk next to it

   6.4. Equipment name has asterisk next to it

7. Fill out rest of equipment

   7.1. Equipment name turns green and have asterisk

8. Fill out all equipment

   8.1. All Equipment names are green and have asterisk

9. Room spinner changed

   9.1. Pop up saying you haven’t saved yet (assume you click yes to save)

10. Fill out all equipment

    10.1. All Equipment names are green and have asterisk

11. Click Save Button

    11.1. Pop up asking to confirm saving (assume you click yes)

      11.1.1. All Equipment asterisk disappear

12. Room Spinner Changed

    12.1. Populates ExpandableList

5. Environmental needs

A dummy xml file in the folder FireAlertScanner on the SD card of the device. The dummy xml file should have no inspections made on it in any room.

6. Special procedural requirements

N/A

7. Intercase dependencies

This test case depends on the following test cases:

Test – XML File Does not Exist

Test – Record Results

Aleph

FireAlertScanner
Test Case Specification (TCS)

Version 1.0

Revision History

Date	Version	Description	Author
12/1/2013	1.0	Did everything	Nicholas Ward

Table of Contents

Test Case Specification (TCS)

1. Test case specification identifier

Display List of Clients

2. Test items

Load Clients spinner

3. Input specifications

1. Launch MainDataEntry activity
2. Open Clients spinner

4. Output specifications

1. Activity launched
   1.1. Client spinner populates
2. Clients spinner opened
   2.1. List of Clients presented

5. Environmental needs

A dummy xml file in the folder FireAlertScanner on the SD card of the device.

6. Special procedural requirements

This test should be repeated multiple times with varying amounts of clients, franchisees etc. to be considered effective.

7. Intercase dependencies

N/A

Aleph

FireAlertScanner
Test Case Specification (TCS)

Version 1.0

Revision History

Date	Version	Description	Author
12/1/2013	1.0	Did everything	Nicholas Ward

Table of Contents

Test Case Specification (TCS)

1. Test case specification identifier

Display Client Locations

2. Test items

Load Service Address spinner

3. Input specifications

1. Launch MainDataEntry activity
2. Open Service Address spinner

4. Output specifications

1. Activity launched
   1.1. Service Address populates
2. Service Address spinner opened
   2.1. List of Service Addresses presented

5. Environmental needs

A dummy xml file in the folder FireAlertScanner on the SD card of the device.

6. Special procedural requirements

This test should be repeated multiple times with varying amounts of clients, franchisees etc. to be considered effective.

7. Intercase dependencies

This test case depends on the following test cases:

Test – Display List of Clients