FireAlertScanner

Version <1.0>

[Note: Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

[Note: Documents described in this section are based on the IEEE 829 standard on testing documentation. Note that we omitted certain sections and documents (e.g., the Test Item Transmittal Report) for the sake of simplicity. Refer to the standard for a complete description of these documents [IEEE Std. 829-2008].]

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/1/2013 | 1.0 | Did everything | Nick |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Relationship to other documents 4

3. System overview 4

4. Features to be tested/not to be tested 4

5. Pass/Fail criteria 5

6. Approach 5

7. Suspension and resumption 5

8. Testing materials (hardware/software requirements) 5

9. Test cases 5

10. Testing schedule 6

# Introduction

[This Section describes the objectives of the tests. The goal is to provide a framework that can be used by managers and testers to plan and execute the necessary tests in a timely and cost-effective manner.]

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.

# Relationship to other documents

[This section explains the relationship of the test plan to the other documents produced during the development effort such as the SRS, and SDS. It explains how all the tests are related to the functional and non-functional requirements, as well as to the system design stated in the respective 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.

# System overview

[This section is focusing on the structural aspects of testing, provides an overview of the system in terms of the components that are tested during the unit test. The granularity of components and their dependencies are defined in this section.]

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.

# Features to be tested/not to be tested

[This section is focusing on the functional aspects of testing, identifies all features and combinations of features to be tested. It also describes all those features that are not to be tested and the reasons for not testing them.]

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.

# Pass/Fail criteria

[This section specifies generic pass/fail criteria for the tests covered in this plan.]

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.

# Approach

[This section describes the general approach to the testing process. It discusses the reasons for the selected integration testing strategy. Different strategies are often needed to test different parts of the system.]

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.

# Suspension and resumption

[This section specifies the criteria for suspending the testing on the test items associated with the plan. It also specifies the test activities that must be repeated when testing is resumed.]

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.

# Testing materials (hardware/software requirements)

[This section identifies the resources that are needed for testing. This should include the physical characteristics of the facilities, including the hardware, software, special test tools, and other resources needed (office space, etc.) to support the tests.]

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.

# Test cases

[This section is the core of the test plan, lists the test cases that are used during testing. Each test case is described in detail in a separate Test Case Specification document. Each execution of these tests will be documented in a Test Incident Report document.]

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

# Testing schedule

[This section covers responsibilities, staffing and training needs, risks and contingencies, and the test schedule.]

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