

TEAM 2

Software Verification and Validation Specification

Authors of this document:

Måns Andersson
Hanna Autio
Moa Eklöf
Oskar Fällström
Ulf Hörndahl
Jonathan Lundholm

Version History

Version	Date	Responsible	Description
1.0	150915	OF	Baseline
1.1	150925	UH & ME	Added test A.1.12

Contents

1	Reference Documents	1
2	Introduction	1
3	Terminology	1
4	Testing	1
4.1	White-box Testing	1
4.2	Black-box Testing	2
5	Reviews	2
5.1	Formal Reviews	2
5.2	Informal Reviews	2
6	Test Environments	2
7	Types of Tests	2
7.1	Unit Tests	2
7.2	Function Tests	3
7.3	System Tests	3
7.4	Regression Tests	3
7.5	Acceptance Tests	3
Appendix A Function Test Specification		4
A.1	MyDevices View Tests	4
A.2	Sensor View Tests	4
A.3	Light Bulb View	5
Appendix B System Test Specification		6
B.1	Use Cases	6
B.2	Quality Tests	6

1 Reference Documents

1. PUSS154212 v1.2 - System Requirements Specification for the current project
2. Programvaruutveckling för Stora System - Projekthandledning v2.2 (*Institutionen för datavetenskap*, Lunds Univeritet 2015)
3. PUSS154253 - Test Matrices for SVVS.

2 Introduction

The Software Verification and Validation Specification (SVVS) describes all tests and reviews conducted during the development of the project. The SVVS acts as a base for the SVVI (Software Verification and Validation Instructions). In the SVVS, there are instructions for reviews as well as information about what types of tests should be performed and by whom. In the appendices, test cases are specified.

3 Terminology

SDP Software Development Plan

SRS Software Requirements Specification

SVVS Software Verification and Validation Specification

SVVI Software Verification and Validation Instructions

STLDD Software Top Level Design Document

SDDD Software Detailed Design Document

SVVR Software Verification and Validation Report

SSD System Specification Document

PFR Project Final Report

PG Project Group Leaders

4 Testing

In order to validate and verify that the system fulfills the customers requirements, various tests are conducted during the development phases. Types of test i.e. unit tests, function tests etc. can be divided into two categories: white-box tests and black-box tests, which are described below. Both of these should be performed.

The tests specified in this document are mapped to the requirements (in ref. 1) via test matrices (see ref. 3).

4.1 White-box Testing

In a white-box test the developer tests the internal structure of the system, by means of making sure that all lines/units of code are executed at least once. These are performed by the developer group during development.

4.2 Black-box Testing

A function test or black-box test examines the functionality of a system from an external point of view. Combinations of input data are chosen and the output data from the system is compared to a specification.

5 Reviews

5.1 Formal Reviews

Formal reviews are conducted according to reference 2.

The scheduled formal reviews, and the documents up for review, in this project are:

1. Software Specification Review (SSR)
 - SDP
 - SRS
 - SVVS
2. Preliminary Design Review (PDR)
 - SVVI
 - STLDD
3. Product Review (PR)
 - SVVR
 - SSD
 - PFR

5.2 Informal Reviews

Informal reviews are to be held before the deadline for the formal review, with enough time to correct any problems found during the informal review. The documents up for review should be made available in the project library at least 24 hours before the informal review.

PG are the responsible coordinators for the informal review.

6 Test Environments

All tests specified in this document should be performed on appropriate emulator software. If possible, the tests should also be run on a physical device.

7 Types of Tests

7.1 Unit Tests

The developer group should perform unit tests, both black-box tests and white-box tests. These tests should be performed continuously during development.

7.2 Function Tests

All functions of the system should be tested according to appendix A. The test group is responsible for performing these tests.

7.3 System Tests

The entire system should be tested according to appendix B. The test group is responsible for performing these tests.

7.4 Regression Tests

After any changes, all system tests should be performed again to ensure that previously functional code is still working. The test group is responsible for performing these tests.

7.5 Acceptance Tests

Acceptance tests are performed by the customer before completion of the project. The customer is responsible for selecting tests to ensure that the software meets their demands.

A Function Test Specification

This is appendix A where we list all function test specifications.

A.1 MyDevices View Tests

Test A.1.1 The first screen that is shown on startup is the MyDevicesView (Req. 5.2.1).

Test A.1.2 The list of available devices is empty on application start up. (Req. 5.2.2)

Test A.1.3 The list on the MyDevices view is scrollable. (Req. 5.2.3)

Test A.1.4 The devices on the list are selectable. (Req 5.2.4)

Test A.1.5 Only one device can be selected at a time. (Req. 5.2.5)

Test A.1.6 When no device is selected and the "Control Device" -button is pressed, a pop-up message "Please select a device" is displayed. (Req. 5.2.6)

Test A.1.7 Sensors are displayed in the list of available devices with "Sensor" as name, and it's MAC address as address and their identifier as id. (Req. 5.2.7)

Test A.1.8 Light bulbs are displayed in the list of available devices with "Light Bulb" as name and it's MAC address as address and their identifier as id. (Req. 5.2.8)

Test A.1.9 The "Get Devices" -button performs a scan for available devices when pressed. (Req. 5.2.9)

Test A.1.10 When the back button is pressed the application is closed (Req. 5.2.10)

Test A.1.11 The layout of the MyDevices View resemble figure 1 in appendix Ref 1 (Req. 5.2.11).

Test A.1.12 The "Get Devices" button only displays the devices with the MAC addresses in Ref 1 (Req. 5.2.12, 5.2.13)

A.2 Sensor View Tests

Test A.2.1 When the "Control Device"-button in the MyDevices View is pressed and a sensor is selected, the Sensor view is opened (Req. 5.3.1).

Test A.2.2 The sensor name and mac-address is shown in the top of the view (Req. 5.3.2, 5.3.3).

Test A.2.3 It is possible to change the on/off -status of the selected sensor with a switch (Req. 5.3.4).

Test A.2.4 There are text fields preceded by "T", "P", "H", "M", "G", "A" that are used to, respectively, display temperature, pressure, humidity, magnetic field strength, gyroscopic data and acceleration (Req. 5.3.5, 5.3.6, 5.3.7, 5.3.8, 5.3.9, 5.3.10).

Test A.2.5 By pressing the corresponding "Get"-button, the values of the temperature, pressure, humidity, magnetic field strength, gyroscopic and acceleration sensors are retrieved if available and displayed (Req. 5.3.11, 5.3.12, 5.3.13, 5.3.14, 5.3.15, 5.3.16).

Test A.2.6 The "Get All" -button gets the values for all six sensors and displays them (Req. 5.3.17).

- Test A.2.7** The "Clear All" button clears all sensor text fields from data (Req. 5.3.18).
- Test A.2.8** If there is no data to retrieve for any of the physical quantities: temperature, pressure, humidity, magnetic field strength, gyroscopic data and acceleration when the corresponding "Get" button is pressed the corresponding, text fields display "No data available" (Req. 5.3.19, 5.3.20, 5.3.21, 5.3.22, 5.3.23, 5.3.24).
- Test A.2.9** The on/off-switch is set according to the information from the REST API (Req. 5.3.25).
- Test A.2.10** When the back button is pressed in the Sensor View, the MyDevices View is opened (Req. 5.3.26).
- Test A.2.11** When the sensor view is opened the temperature, pressure, humidity, magnetic field strength, gyroscopic data and acceleration text fields are empty (Req. 5.3.27).
- Test A.2.12** The layout of the Sensor View resemble figure 2 in appendix Ref 1 (Req. 5.3.28).

A.3 Light Bulb View

- Test A.3.1** The Light Bulb View opens when a light bulb is chosen in the MyDevices View and the "Control device" button is pressed. (Req 5.4.1)
- Test A.3.2** The name of the selected light bulb is shown at the top of the View (Req 5.4.2).
- Test A.3.3** The selected light bulb can be turned on/off with a switch (Req. 5.4.3).
- Test A.3.4** The field of R-, G-, B-, W-value is editable and preceded by "R:", "G:", "B:" and "W:" respectively (Req. 5.4.4, 5.4.5, 5.4.6, 5.4.7).
- Test A.3.5** When the Light Bulb View is opened, the fields are empty (Req. 5.4.8).
- Test A.3.6** The "Get"-button retrieves the R-, G-, B-, W-values and present them in their corresponding fields (Req. 5.4.9).
- Test A.3.7** The "Set button" sets the color of the light bulb (Req. 5.4.10).
- Test A.3.8** If an input value is left blank the value is interpreted as 00 (Req. 5.4.11) .
- Test A.3.9** The fields only accepts two characters that represent hexadecimal numbers (e.g. 00 to FF and all combinations inbetween) (Req 5.4.12, Req 5.4.13).
- Test A.3.10** A pop-up message saying "Color successfully changed" is displayed when the values of the light bulb were successfully set. (Req 5.4.14)
- Test A.3.11** A pop-up message saying "Error: Could not change color." is displayed when the values of the light bulb were unsuccessfully set. (Req 5.4.15)
- Test A.3.12** When the light bulb is off, the "Set"-button is unavailable (Req. 5.4.16).
- Test A.3.13** When the back button is pressed the system switches to MyDevices View (Req. 5.4.17).
- Test A.3.14** The layout of the Light Bulb View resemble figure 3 in appendix Ref 1 (Req. 5.4.18).

B System Test Specification

This is appendix B where we list all system test specifications.

B.1 Use Cases

- Test B.1.1** Scenario 5.1.1 in Ref 1 is supported (Req. 5.1.1).
- Test B.1.2** Exception 1 described in scenario 5.1.1 can be generated by removing the devices from range while attempting scenario 5.1.1 (Req 5.1.1).
- Test B.1.3** Scenario 5.1.2 in Ref 1 is supported (Req. 5.1.2).
- Test B.1.4** Exception 1 in scenario 5.1.2 can be generated by not selecting a device in step 1 (Req. 5.1.2).
- Test B.1.5** Scenario 5.1.3 in Ref 1 is supported (Req. 5.1.3).
- Test B.1.6** Exception 1 described in scenario 5.1.3 can be generated by not selecting a device in step 1 (Req. 5.1.3).
- Test B.1.7** Scenario 5.1.4 in Ref 1 is supported (Req. 5.1.4).
- Test B.1.8** Scenario 5.1.5 in Ref 1 is supported (Req. 5.1.5).
- Test B.1.9** Scenario 5.1.6 in Ref 1 is supported (Req. 5.1.6).
- Test B.1.10** Scenario 5.1.7 in Ref 1 is supported (Req. 5.1.7).
- Test B.1.11** Scenario 5.1.8 in ref 1 is supported (Req. 5.1.8).
- Test B.1.12** Scenario 5.1.9 in Ref 1 is supported (Req. 5.1.9).
- Test B.1.13** If some data is unavailable when attempting scenario 5.1.9, the text in the corresponding field is set to "No data available." (Req. 5.1.9).
- Test B.1.14** Scenario 5.1.10 in Ref 1 is supported (Req. 5.1.10).
- Test B.1.15** If some data is unavailable when attempting scenario 5.1.10, the text in the corresponding field is set to "No data available." (Req. 5.1.10).
- Test B.1.16** Scenario 5.1.11 in Ref 1 is supported (Req. 5.1.11).
- Test B.1.17** If some data is unavailable when attempting scenario 5.1.11, the text in the corresponding field is set to "No data available." (Req. 5.1.11).
- Test B.1.18** Scenario 5.1.12 in Ref 1 is supported (Req. 5.1.12).

B.2 Quality Tests

- Test B.2.1** Test of usability using persons without prior knowledge of the project or the app (Req. 6.1).
- Test B.2.2** The response time for any interaction with the system is below two seconds (Req. 6.2).
- Test B.2.3** The time-out time for the system is 15 seconds (Req. 6.3).