

SVVS – Software Verification and Validation Specification

TG

October 2, 2018

Contents

1	Introduction	2
2	Reference documents	2
3	Overview	2
4	Test cases	3
5	Requirements checked in reviews	3
6	Appendices	3
A	System test specification	4
A.1	Quality requirements	4
A.2	Data	4
A.2.1	Creating account	4
A.2.2	Creating rides	4
A.2.3	Searching and joining rides	4
A.3	Delivery requirements	5
B	Function test specification	6
B.1	Creating account	6
B.2	Log in and log out	6
B.3	Creating rides	6
B.4	Searching and joining rides	6
B.5	Displaying joined rides and deleting a ride	7
B.6	Administration	7

Document history

Ver.	Date	Resp.	Description
0.x	2018-09-16	TG	First draft version
0.2	2018-09-22	TG	Second draft version
0.3	2018-09-26	TG	Thrid draft version
1.0	2018-10-02	TG	Baseline

1 Introduction

This document describes how the "Grupp 5 project" should be verified

2 Reference documents

1. Programvaruutveckling för Stora System – Projekthandledning, v 2.5
2. Software Requirements Specifications, v. 0.x

3 Overview

During the project, three formal reviews are carried out

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

These formal reviews are carried out as described in Ref. 1.

Each formal review is proceeded by an informal review, without any customer representative involved, as described in Ref. 1. The informal reviews are scheduled so that updated material can be delivered to the formal review.

In addition to the informal reviews listed above the the SDDD should also be reviewed by the development group.

4 Test cases

The tests are carried out in the following phases:

Unit test: Basic code functionality testing is carried out by individual developers and developer groups before the tested parts are integrated with any other group's code. Primarily this is done using junit, but can also be done using manually written basic shell scripts. This is done informally, and no documentation is needed.

Regression test: All previously specified test cases are run during regression test. This is done automatically, whenever a pull request is created or a branch is merged into master, through a Jenkins solution and is not manually documented.

Function test: Here individual, but complete, functions are tested as far as possible before the function is merged with other newly developed functions in the system. This is done with mainly shell scripts and manual running by the test group. The test group also documents and approves the tests.

System test: Here, general tests for the whole system are carried out. This is done manually by the test group. The test group also documents and approves the tests.

Acceptance test: These tests are carried out by the customer during the last review (PR). Samples from all specified test cases are used together with a number of test cases that are decided by the customer.

5 Requirements checked in reviews

In reviews all requirements should be checked by the reviewers.

The following requirements (according to Ref. 2) are only verified through reviews: 6.4.3, 6.7.4, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 8.1, 8.2, 8.3.

6 Appendices

There are two appendices:

A System test specification

B Function test specification

A System test specification

After each test case the tested requirements according to Ref. 2 are listed.

A.1 Quality requirements

- ST1.1 a. Successful creating a ride within 20 seconds when the website has already been opened [SRS req. 7.3].
b. Successful joining a ride within 20 seconds when the website has already been opened [SRS req. 7.3].
- ST1.2 Successful accessing the website within two seconds using one of the computer rooms in "E-Huset" [SRS req. 7.5]
- ST1.3 Successful accessing any of the redirect links on the web-page that belongs to the system within two seconds using one of the computer rooms in "E-Huset" [SRS req. 7.5]

A.2 Data

A.2.1 Creating account

- ST2.1.1 a. Create account, illegal characters' type in username [SRS reg 6.6.1]
b. Create account, illegal characters' length in username [SRS reg 6.6.1]
- ST2.1.2 a. Create account, illegal characters' type in the email field [SRS reg 6.6.2]
b. Create account, illegal characters' length in the email field [SRS reg 6.6.2]
- ST2.1.3 a. Create account, illegal characters' type in Passwords [SRS reg 6.6.7]
b. Create account, illegal characters' length in Password [SRS reg 6.6.7]

A.2.2 Creating rides

- ST2.2.1 a. Create a ride, illegal format used in arrival time field [SRS reg 6.6.3]
b. Create a ride, illegal format used in departure time field [SRS reg 6.6.3]
- ST2.2.2 a. Create a ride, illegal characters' type used in arrival location field [SRS reg 6.6.5]
b. Create a ride, illegal characters' type used in departure location field [SRS reg 6.6.6]
- ST2.2.3 Create ride, user checks on being both driver and passenger [SRS reg 6.8.1]

A.2.3 Searching and joining rides

- ST2.3.1 a. Search for a ride, illegal format used in arrival time field [SRS reg 6.6.4]
b. Search for a ride, illegal format used in departure time field [SRS reg 6.6.4]

- ST2.3.2 a. Search for a ride, illegal characters' type used in departure location field [SRS reg 6.6.5]
 - b. Search for a ride, illegal characters' type used in departure location field [SRS reg 6.6.6]
- ST2.3.3 Search for a ride, user checks on being both driver and passenger [SRS reg 6.8.1]

A.3 Delivery requirements

- ST3.1 a. Successful getting the application to work with Google Chrome on computers running Windows 10. By working means all of the function tests has been executed successfully [SRS reg 9.1].
 - b. Successful getting the application to work with Google Chrome using on computers running Ubuntu 18.04. By working means all of the function tests has been executed successfully [SRS reg 9.1].
 - c. Successful getting the application to work with Google Chrome using on computers running MacOS 10.13: "High Sierra". By working means all of the function tests has been executed successfully [SRS reg 9.1].
- ST3.2 Successful compiling and running the back end server on computers that run Ubuntu 18.04 and have Java 8 installed [SRS reg 9.2].
- ST3.3 Successful confirming the database is empty after the system is delivered [SRS reg 9.3].

B Function test specification

After each test case the tested requirements according to Ref. 2 are listed.

B.1 Creating account

- FT1.1 Successful creation of account [SRS req. 6.1.1]
- FT1.2 Create account, illegal or no username [SRS req. 6.1.2, 6.6.1]
- FT1.3 Create account, illegal or no email [SRS req. 6.1.2, 6.6.2]
- FT1.4 Create account, illegal or no password [SRS req. 6.1.2, 6.6.1]
- FT1.5 Create account, email already belongs to an account [SRS req. 6.1.3]
- FT1.6 Create account, username is already taken [SRS req. 6.1.3]

B.2 Log in and log out

- FT2.1 Successful login [SRS req. 6.2.1, 6.6.1, 6.6.2]
- FT2.2 Login, incorrect username [SRS req. 6.2.2]
- FT2.3 Login, incorrect password [SRS req. 6.2.2]
- FT2.4 Login five times, incorrect username or password [SRS req. 6.2.2, 6.2.3]
- FT2.5 Successful logout [SRS req. 6.2.4]

B.3 Creating rides

- FT3.1 Successfully create ride [SRS req. 6.3.1]
- FT3.2 Create ride, illegal departure time [SRS req. 6.3.2, 6.6.4]
- FT3.3 Create ride, illegal arrival time [SRS req. 6.3.2, 6.6.4]
- FT3.4 Create ride, illegal number of free seats [SRS req. 6.3.2]
- FT3.5 Create ride, illegal departure location [SRS req. 6.3.2]
- FT3.6 Create ride, illegal arrival location [SRS req. 6.3.2]
- FT3.7 Successful cancellation of ride [SRS req. 6.3.3]
- FT3.8 Create ride, arrival time earlier than departure time [SRS req. 6.3.5]
- FT3.9 Create ride, departure time which has already passed [SRS req. 6.3.4]
- FT3.10 Create ride, arrival time which has already passed [SRS req. 6.3.4]
- FT3.11 Create ride, unreasonable departure time based on previous rides [SRS req. 6.3.7]
- FT3.12 Create ride, the user is already a driver in another ride at the same time [SRS req. 6.3.6, 6.6.5]
- FT3.13 Create ride, the user is already a passenger in another ride at the same time [SRS req. 6.3.6, 6.6.5]

B.4 Searching and joining rides

- FT4.1 Successfully joins a ride as a passenger [SRS req. 6.4.1]
- FT4.2 Join a ride as passenger, the user has already joined another ride that takes place during the same time [SRS req. 6.4.2]
- FT4.3 Successful kick of another user from the ride, driver [SRS req. 6.4.4]

FT4.4 Successful ban of another user from joining the ride, driver [SRS req. 6.4.4]

FT4.5 Search ride [SRS req. 6.4.5]

B.5 Displaying joined rides and deleting a ride

FT5.1 Successful deletion of a ride, driver [SRS req. 6.5.1]

B.6 Administration

FT6.1 Successful addition of a user, administrator [SRS req. 6.7.1]

FT6.2 Successful ban of a user, administrator [SRS req. 6.7.2]

FT6.3 Successful deletion of a user, administrator [SRS req. 6.7.3]

FT6.4 Successful deletion of a ride, administrator [SRS req. 6.7.5]