SVVI – Software Verification and Validation Instructions

TG

October 24, 2018

Contents

Document history

Ver.	Date	Resp.	Description
0.x	2018-09-27	TG	First draft version
0.2	2018-10-05	TG	Second draft version
1.1	2018-10-22	TG	FT2.4, FT3.7, FT3.11 expired

1 Introduction

This document presents detailed test instructions for the "Grupp 5 project".

2 Reference documents

1. Software Verification and Validation Specification, SVVS, v. 1.0

3 Test instructions

The detailed test instructions for the test cases introduced in ref. 1 are shown in the appendices below. Test cases are documented in the form of steps that are carried out by the tester or the result of the system. When steps are initiated with 'Check that' they describe result of the system, and the tester should check that this really happens.
Preferably the environment and database should be clean and empty before testing, but is not necessary if thorough logs and data are collected of the database before testing begins. The tests can be run in any order if the tester keeps track of the users and rides created during their testing, unless a new instance of the program is run for each test.

A System test

A.1 Quality requirements

ST1.1.a: Successful creating a ride within 20 seconds when the website has already been opened

Start state: User not logged in and has not created a ride.
End state: User logged in and has created a ride.

1. With the website open, log in existing user account.
2. Click "Create ride".
3. Enter legal departure time, arrival time, number of seats available, departure location and arrival location. For example: Departure time: 10/09-18:00, Arrival time: 10/09-21:00, Number of seats: 4, Departure location: Lund, Arrival location: Malmö.
4. Check that the message displaying the ride has been created appears within 20 seconds.
5. Check the database to see if the ride was created. Refer to ST3.2 how to access the BASE GUI.

ST1.1.b: Successful joining a ride within 20 seconds when the website has already been opened

Start state: User not logged in nad has not joined a ride.
End state: User logged in and has joined a ride.

1. With the website open, log in existing user account.
2. Click "Search ride".
3. Enter legal departure time, arrival time, number of seats available, departure location and arrival location. For example: Departure time: 10/09-18:00, Arrival time: 10/09-21:00, Number of seats: 4, Departure location: Lund, Arrival location: Malmö.
4. Click "Join ride" on one of the search results.
5. Check that the message displaying the user has been successfully joined a ride appears within 20 seconds.
6. Check the database to see if the user has joined ride above. Refer to ST3.2 how to access the BASE GUI.

ST1.2: Successful accessing the website within two seconds using one of the computer rooms in "E-Huset"

Start state: User not connected to the website using a computer in one of the computer rooms in "E-Huset.
End state: User connected to the website using a computer in one of the computer rooms in "E-Huset.
1. On a computer in one of the computer rooms in "E-Huset" open any of the browser and Operative system supported according to [SRS req. 9.1].

2. Enter the domain-name or IP-address of the website in the URL.
3. Check that the website's startpage appears within two seconds.

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"

Start state: User connected to the website using a computer in one of the computer rooms in "E-Huset.
End state: User redirected to any of the links or tabs on the web-page that belongs to the system

1. On the website click on any link or tabs that belongs to the system.
2. Check that the link or tab appears within two seconds.

A.2 Data

A.2.1 Creating account

ST2.1.1.a: Create account, illegal characters' type in Username

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input illegal characters' type in Username. For example: Ål@.
3. Check that the web-page does not crash and user not disconnected.
4. Check the system's log or console if the exception has been handled.
5. Check that there is a message telling user to input the correct characters.
6. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

ST2.1.1.b: Create account, illegal characters' length in Username

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input illegal characters' length in Username. For example: Al or Magneto-hydrodynamicsism.
3. Check that the web-page does not crash and user not disconnected.
4. Check the system's log or console if the exception has been handled.
5. Check that there is a message telling user to input the correct characters.
6. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

ST2.1.2.a: Create account, illegal characters' type in the email-field

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input illegal characters' type in the email-field. For example: Ål@gmail.com.
3. Input legal characters' type and length in username and password. For example: username Username and password Password.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct characters.
7. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

ST2.1.2.b: Create account, illegal characters' length in the email-field

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input illegal characters' length in the email-field. For example: Al@gmail.com or Magnetohydrodynamicsism@gmail.com.
3. Input legal characters' type and length in username and password. For example: username Username and password Password.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct characters.
7. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

ST2.1.3.a: Create account, illegal characters' type in Password

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input illegal characters' type in Password. For example: Ål@.
3. Check that the web-page does not crash and user not disconnected.
4. Check the system's log or console if the exception has been handled.
5. Check that there is a message telling user to input the correct characters.
6. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

ST2.1.3.b: Create account, illegal characters' length in Password

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input illegal characters' length in Password. For example: Al or Magneto-hydrodynamicsism.
3. Check that the web-page does not crash and user not disconnected.
4. Check the system's log or console if the exception has been handled.
5. Check that there is a message telling user to input the correct characters.
6. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

A.2.2 Creating rides

ST2.2.1.a: Create a ride, illegal format used in arrival-time field

Start state: User is logged in.
End state: No ride has been created.

1. Access the "Create a ride" function.
2. Input illegal format in arrival-time. For example: 32/13-25:61.
3. Input legal format in departure-time. For example: 30/12-23:50.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct format.
7. Check that no ride has been created in the database. Refer to ST3.2 how to access the BASE GUI.

ST2.2.1.b: Create a ride, illegal format used in departure-time field

Start state: User is logged in.
End state: No ride has been created.

1. Access the "Create a ride" function.
2. Input illegal format in departure-time. For example: 32/13-25:61.
3. Input legal format in arrival-time. For example: 30/12-23:50.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct format.
7. Check that no ride has been created in the database. Refer to ST3.2 how to access the BASE GUI.

ST2.2.2.a: Create a ride, illegal characters' type used in arrival-location field

Start state: User is logged in.
End state: No ride has been created.

1. Access the "Create a ride" function.
2. Input illegal characters' type in arrival-location field. For example $\mu\pi$.
3. Input legal characters' type in departure-location field. For example: Lund.
4. Check that the web-page does not crash and user not disconnected.

5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct format.
7. Check that no ride has been created in the database. Refer to ST3.2 how to access the BASE GUI.

ST2.2.2.b: Create a ride, illegal characters' type used in departure-location field

Start state: User is logged in.
End state: No ride has been created.

1. Access the "Create a ride" function.
2. Input illegal characters' type in departure-location field. For example $\mu\pi$.
3. Input legal characters' type in arrival-location field. For example: Lund.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct format.
7. Check that no ride has been created in the database. Refer to ST3.2 how to access the BASE GUI.

ST2.2.3: Create ride, user checks on being both driver and passenger

Start state: User is logged in.
End state: No ride has been created.

1. Access the "Create a ride" function.
2. User checks on being both driver and passenger.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to only choose one of the alternatives.
7. Check that no ride has been created in the database. Refer to ST3.2 how to access the BASE GUI.

A.2.3 Searching and joining ride

ST2.3.1.a: Search for a ride, illegal format used in arrival-time field

Start state: User is logged in.
End state: Arrival-time not found. Ride not found.

1. Access the search function.
2. Input illegal format in arrival-time. For example: 32/13-25:61.
3. Input legal format in departure-time. For example: 30/12-23:50.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct format.
7. Check that no data from the database has been fetched.

ST2.3.1.b: Search for a ride, illegal format used in departure-time field

Start state: User is logged in.
End state: Departure-time not found. Ride not found.

1. Access the search function.
2. Input illegal format in departure-time. For example: 32/13-25:61.
3. Input legal format in arrival-time. For example: 30/12-23:50.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled. 6. Check that there is a message telling user to input the correct format.
7. Check that no data from the database has been fetched.

ST2.3.2.a: Search for a ride, illegal characters' type used in arrival and departure-location field

Start state: User is logged in.
End state: Arrival-location not found. Ride not found.

1. Access the search function.
2. Input illegal characters' type in arrival-location. For example $\mu\pi$.
3. Input legal characters' type in departure-location. For example: Lund.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct format.
7. Check that no data from the database has been fetched.

ST2.3.2.b: Search for a ride, illegal characters' type used in departure-location field

Start state: User is logged in.
End state: Departure-location not found. Ride not found.

1. Access the search function.
2. Input illegal characters' type in departure-location. For example $\mu\pi$.
3. Input legal characters' type in arrival-location. For example: Lund.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to input the correct format.
7. Check that no data from the database has been fetched.

ST2.3.3: Search for a ride, user checks on being both driver and passenger

Start state: User is logged in.
End state: Ride not found.

1. Access the search function.

2. User checks on being both driver and passenger.
4. Check that the web-page does not crash and user not disconnected.
5. Check the system's log or console if the exception has been handled.
6. Check that there is a message telling user to only choose one of the alternatives.
7. Check that no data from the database has been fetched.

A.3 Delivery Requirements

ST3.1.a: Successful getting the application to work with Chrome on computers running Windows 10

Start state: User not connected to the website using Chrome on a computer running Windows 10.
End state: User connected to the website and successfully performed all of the function tests on the website

1. On a computer running Windows 10 open Chrome browser.
2. Enter the domain-name or IP-address of the website in the URL.
3. Check that the website's startpage appears.
4. Perform all of the function tests according to Appendix B.
5. Check that the functions work properly.

ST3.1.b: Successful getting the application to work with Chrome on computers running MacOs 10.12 "Sierra"

Start state: User not connected to the website using Chrome on a computer running MacOs 10.12 "Sierra".
End state: User connected to the website and successfully performed all of the function tests on the website.

1. On a computer running MacOs 10.12 "Sierra" open Chrome browser.
2. Enter the domain-name or IP-address of the website in the URL.
3. Check that the website's startpage appears.
4. Perform all of the function tests according to Appendix B.
5. Check that the functions work properly.

ST3.1.c: Successful getting the application to work with Chrome on computers running Ubuntu 18.04

Start state: User not connected to the website using Chrome on computers running Ubuntu 18.04.
End state: User connected to the website and successfully performed all of the function tests on the website.

1. On computers running Ubuntu 18.04 open Chrome browser.

2. Enter the domain-name or IP-address of the website in the URL.
3. Check that the website's startpage appears.
4. Perform all of the function tests according to Appendix B.
5. Check that the function works properly.

ST3.2: Successful compiling and running the back end server on computers that run Ubuntu 18.04 and have Java 8 installed

Start state: The server has not been started.
End state: The server has been started and user accessed the BASE GUI.

1. On a computer that run Ubuntu 18.04 and have Java 8 installed. Download or clone the systems repository at https://github.com/Puss18G5/documentLibrary.
2. Import the project in Eclipse.
3. Start the server using the file BaseServer.java.
4. Check that there's no error compiling the file.
5. Go to http://localhost:xxxx (URL updated later) and login using username Admin and password provided by the system-group.
6. Check that the BASE GUI appears.

ST3.3: Successful confirming the database is empty after the system is delivered

Start state: The system has been delivered.
End state: User confirms that the database is empty.

1. Go to http://localhost:xxxx (URL updated later) and login using username Admin and password provided by the system-group.
2. Check that the no entry of Users, Locations, Rides and PassengerRides are found in the BASE GUI.

B Function test

B.1 Creating account

FT1.1: Successful creation of account

Start state: User not logged in.
End state: Account has been created.

1. Access account creation page.
2. Input legal username, email and password for new account.
3. Click create account.
4. "Account created" is displayed
5. Try to login with the new account to see if it has been created
6. Login succeds.

FT1.2: Create account, illegal or no username

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input illegal username, for example "a/". Input legal email, for example "example@gmail.com. Input legal password, for example "password".
3. Click create account.
4. An error message is displayed.
5. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.
6. Access account creation page.
7. Input no username. Input legal email, for example "example@gmail.com". Input legal password, for example "password".
8. Click create account.
9. An error message is displayed.
10. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

FT1.3: Create account, illegal or no email

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input legal username, for example "username". Illegal email, for example "asd". Input legal password, for example "password".
3. Click create account.
4. An error message is displayed.
5. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

6. Access account creation page.
7. Input legal username, for example "username". Input no email. Input legal password, for example "password".
8. Click create account.
9. An error message is displayed.
10. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

FT1.4: Create account, illegal or no password

Start state: User not logged in.
End state: Account has not been created.

1. Access account creation page.
2. Input legal username, for example "username". Input legal email, for example "example@gmail.com. Input illegal password, for example "asd".
3. Click create account.
4. An error message is displayed.
5. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.
6. Access account creation page.
7. Input legal username, for example "username". Input legal email, for example "example@gmail.com". Input no password.
8. Click create account.
9. An error message is displayed.
10. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

FT1.5: Create account, email already belongs to an account

Start state: User not logged in, at least one user in database.
End state: Account has not been created.

1. Access account creation page.
2. Input the email of an already existing account.
3. Input legal username, for example "username". Input legal password, for example "password".
4. An error message is displayed.
5. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

FT1.6: Create account, username is already taken

Start state: User not logged in, at least one user in database.
End state: Account has not been created.

1. Access account creation page.
2. Input the username of an already existing account.

3. Input legal email, for example "example@gmail.com". Input legal password, for example "password".
4. Check database to see if the account has been created. Refer to ST3.2 how to access the BASE GUI.

B.2 Log in and log out

FT2.1: Successful login

Start state: User not logged in, user exists in database.
End state: User has been logged in.

1. Enter the username, email and password of a user.
2. Login succeds.

FT2.2: Incorrect username

Start state: User not logged in, user exists in database.
End state: User was not logged in

1. Enter the email and password of a user.
2. Enter an incorrect username.
3. An error message is shown.

FT2.3: Incorrect password

Start state: User not logged in, user exists in database.
End state: User was not logged in.

1. Enter the email and username of a user.
2. Enter an incorrect password.
3. An error message is shown.

FT2.4: Successful logout

Start state: User is logged in.
End state: User is logged out and taken back to the login screen.

1. Log out.

B.3 Creating rides

FT3.1: Successfully create ride

Start state: User is logged in. The user is not already booked in the specified time interval.
End state: Ride was created.

1. Click "Create ride".

2. Enter legal departure time, arrival time, number of seats available, departure location and arrival location.
3. A message that ride was created is displayed.
4. Check database to see if ride was created.

FT3.2: Create ride, illegal departure time

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".
2. Enter legal arrival time, number of seats available, departure location and arrival location.
3. Enter illegal departure time.
3. An error message is shown.
4. Check database to see if ride was created.

FT3.3: Create ride, illegal arrival time

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".
2. Enter legal departure time, number of seats available, departure location and arrival location.
3. Enter illegal arrival time.
3. An error message is shown.
4. Check database to see if ride was created.

FT3.4: Create ride, illegal number of seats available

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".
2. Enter legal departure time, arrival time, departure location and arrival location.
3. Enter illegal number of seats available.
3. An error message is shown.
4. Check database to see if ride was created.

FT3.5: Create ride, illegal departure location

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".

2. Enter legal departure time, arrival time, number of seats available, and arrival location.
3. Enter illegal departure location.
3. An error message is shown.
4. Check database to see if ride was created.

FT3.6: Create ride, illegal arrival location

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".
2. Enter legal departure time, arrival time, number of seats available and departure location.
3. Enter illegal arrival location.
3. An error message is shown.
4. Check database to see if ride was created.

FT3.7: Create ride, arrival time earlier than departure time

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".
2. Enter legal departure time, number of seats available, departure location and arrival location.
3. Enter arrival time that is earlier than the departure time.
4. Error message is displayed.
5. Check database to see if ride was created.

FT3.8: Create ride, departure time which has already passed

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".
2. Enter legal arrival time, number of seats available, departure location and arrival location.
3. Enter departure time that has already passed.
4. Error message is displayed.
5. Check database to see if ride was created.

FT3.9: Create ride, arrival time which has already passed

Start state: User is logged in.
End state: Ride was not created.

1. Click "Create ride".

2. Enter legal departure time, number of seats available, departure location and arrival location.
3. Enter arrival time that has already passed.
4. Error message is displayed.
5. Check database to see if ride was created.

FT3.10: Create ride, the user is already a driver in another ride at the same time

Start state: User is logged in and is already a driver in another ride.
End state: Ride was not created.

1. Click "create ride".
2. Enter legal arrival time, number of seats available, departure location and arrival location.
3. Enter departure time that is during the users already existing ride.
4. Error message is displayed.
5. Check database to see if ride was created.

FT3.11: Create ride, the user is already a passenger in another ride at the same

Start state: User is logged in and is already a passenger in another ride.
End state: Ride was not created.

1. Click "create ride".
2. Enter legal arrival time, number of seats available, departure location and arrival location.
3. Enter departure time that is during the users already existing ride.
4. Error message is displayed.
5. Check database to see if ride was created.

B.4 Searching and joining rides

FT4.1: Successfully joins a ride as a passenger

Start state: User is logged in, search ride-page.
End state: User is a passenger in a ride.

1. Select ride.
2. Click "Join".
3. Check is user is a passenger in the selected ride.

FT4.2: Join a ride as passenger, the user has already joined another ride that takes place during the same time

Start state: User is logged in, at search ride-page, and is already a passenger in another ride.

End state: User not a passenger in the second ride.

1. Select ride.
2. Click "Join".
3. Error message is shown.
4. Check that user is not a passenger in the selected ride.

FT4.3: Successful kick of another user from the ride, driver

Start state: User is logged in and is a driver in a ride with user 'abc123' as a passenger.
End state: User 'abc123' is kicked from the ride.

1. Select ride.
2. Select user 'abc123'.
3. Click "Kick".
4. Check that the user 'abc123' is kicked from the selected ride.

FT4.4: Successful ban of another user from joining the ride, driver

Start state: User is logged in and is a driver in a ride with user 'abc123' as a passenger.
End state: User 'abc123' is banned from the ride.

1. Select ride.
2. Select user 'abc123'.
3. Click "Ban".
4. Check that the user 'abc123' is banned from the selected ride.

B.5 Displaying joined rides and deleting rides

FT5.1: Successful deletion of a ride, driver

Start state: User is logged in and is the driver of a ride.
End state: Ride is deleted.

1. Click "Delete ride".
2. Check database if the ride was deleted.

B.6 Administration

FT6.1: Successful addition of a user, administrator

Start state: Administrator is logged in. User 'abc123' does not exist in the database.
End state: Administrator is logged in. User 'abc123' exist in the database.

1. Select the administration page.

2. Add new user 'abc123'.
3. Check database to see if user 'abc123' exists and has a legal password.

FT6.2: Successful ban of a user, administrator

Start state: Administrator is logged in. User 'abc123' exists in the database.
End state: Administrator is logged out. User 'abc123' is banned.

1. Select the administration page.
2. Ban user 'abc123'.
3. Log out.
4. Try to log in as user 'abc123'.
5. Error message is show and the user is not logged in.

FT6.3: Successful deletion of a user, administrator

Start state: Administrator is logged in. User 'abc123' exists in the database.
End state: Administrator is logged out. User 'abc123' is deleted from the database.

1. Select the administration page.
2. Delete user 'abc123'.
3. Check database to see if the user 'abc123' was deleted.

FT6.4: Successful deletion of a ride, administrator

Start state: Administrator is logged in. One ride exists in the system.
End state: Administrator is logged in. No rides exists in the system.

1. Display the list of created rides.
2. Select the ride and click delete.
3. Check database to see if the ride was deleted.