

# Project Assignment 2

Eskil Opdahl Nordland and Julian Jark.

Project repository: <https://github.com/Eskalol/INF3121>

## Requirement 1 - Manual tests, test design, bugs

### Test 1: Create new user

The purpose of this test is to make sure new users can be successfully created and that they are automatically logged in on creation.

#### Pre-conditions

None

#### Steps

1. User clicks "sign in"
2. User clicks "register"
3. User fills in valid user information(email, password, first/last-name, country, state, zip code, city, address, phone) in form.
4. User clicks "register"-button

#### Expected results

The system confirms the creation of the user account.

#### Post-conditions

The user is logged into the system.

### Test 2: User adds (selects) multiple product to the cart

The purpose of this test is to make sure that a user can add multiple items into the shopping cart and the subtotal is updated correctly.

#### Pre-conditions

User is logged into the system.

#### Steps

1. User clicks on a product in the "random products" widget
2. User clicks "add to cart"-button
3. User hovers over cart icon and then clicks the "view cart"-button
4. User clicks "continue shopping"-button

5. User repeats step 1, 2 and 3

#### Expected results

Both items are added in the shopping cart.

The subtotal is the correct sum of the total price of the two items.

#### Post-conditions

Both items added are in the shopping cart.

The subtotal is the correct sum of the total price of the two items.

### Test 3: User increments number product items in the cart

The purpose of this test is to make sure the the user can change number of items for a product in the cart.

#### Pre-conditions

There are 2 products in the cart.

#### Steps

1. User navigates to the cart page
2. User clicks the number dropdown for the second product
3. User selects 2 as the number of items

#### Expected results

The second product items is incremented to 2.

The total price for the product is changed.

#### Post-conditions

The number of items for the second product is 2.

The subtotal is updated accordingly.

### Test 4: User removes a product from the cart

The purpose of this test is to make sure that a user can remove an item from the shopping cart.

#### Pre-conditions

There are 2 products in the cart

#### Steps

1. User navigates to the cart page
2. User clicks the x button right next to the first product

#### Expected results

The first product is removed.

#### Post-conditions

There are 1 product in the cart.

The subtotal price is the same the total price of the item in cart.

### Test 5: User places an order

The purpose of this test is to make sure a user can fully checkout his/her shopping cart and place the order.

#### Pre-conditions

The cart is not empty.

#### Steps

1. User clicks "checkout"-button from the cart page.
2. User validates "Billing and Shipping Addresses" and clicks "Continue checkout"
3. User selects "shipping option" and clicks "continue checkout"
4. User clicks "place order"

#### Expected results

The system confirms that the order has been placed

#### Post-conditions

The order is placed, and confirmation email is sent.

### Test 6: User change password

The purpose of this test is to make sure that a user should be able to change password.

#### Pre-conditions

User is logged in.

#### Steps

1. User clicks "My account"
2. User clicks "Change password"
3. User types in the current password
4. User types in a new password in both new password fields.
5. User clicks "Continue"

#### Expected results

The system confirms that the password has been changed.

#### Post-conditions

Password is changed.

### Test 7: User logs out

The purpose of this test is to check the user is able to sign out.

#### Pre-conditions

User is logged in.

#### Steps

6. User clicks "Sign Out"

#### Expected results

User is logged out and redirected to the sign in page.

#### Post-conditions

User is logged out.

### Test 8: User signs in

The purpose of this test is to make sure a valid user can successfully log in.

#### Pre-conditions

None (user is logged out).

#### Steps

1. User clicks "sign in"
2. User fills in valid e-mail and password
3. User clicks "sign in"

#### Expected results

User is successfully authenticated, and redirected to "my account"-page.

#### Post-conditions

User is logged into the system.

## Test 9: User resets password

The purpose of this test to make sure a user can reset his/hers password when it's forgotten.

### Pre-conditions

None (user is logged out)

### Steps

1. User clicks "sign in"
2. User clicks "Forgot your password?"
3. User fills in valid e-mail
4. User clicks on the link provided in e-mail.
5. User types in a new valid password
6. User clicks "continue"

### Expected results

The system confirms the change of password, and redirected to "my account"-page

### Post-conditions

User is logged into the system.

## Error report

**Date:** 19.04.16

**Project:** Avactis shopping cart

**Programmer:** Avactis

**Tester:** Julian Jark and Eskil Nordland

**Program/Module:** /product-info.php **Build/Revision/Release:** 4.7.9

**Software Environment:** PHP

**Hardware Environment:** n/a

**Status of incident:** Open

**Number of Occurrences:** 1 on all product review pages.

**Severity:** 4    **Impact:** 4    **Priority:** minor

**Detailed Description:** The server does not validate the number of stars added to a review. A malicious user can exploit this to add an insane amount of stars to a review and does make the page weird and/or break it.

Image 1: Example of how to change the stars amount to what you want

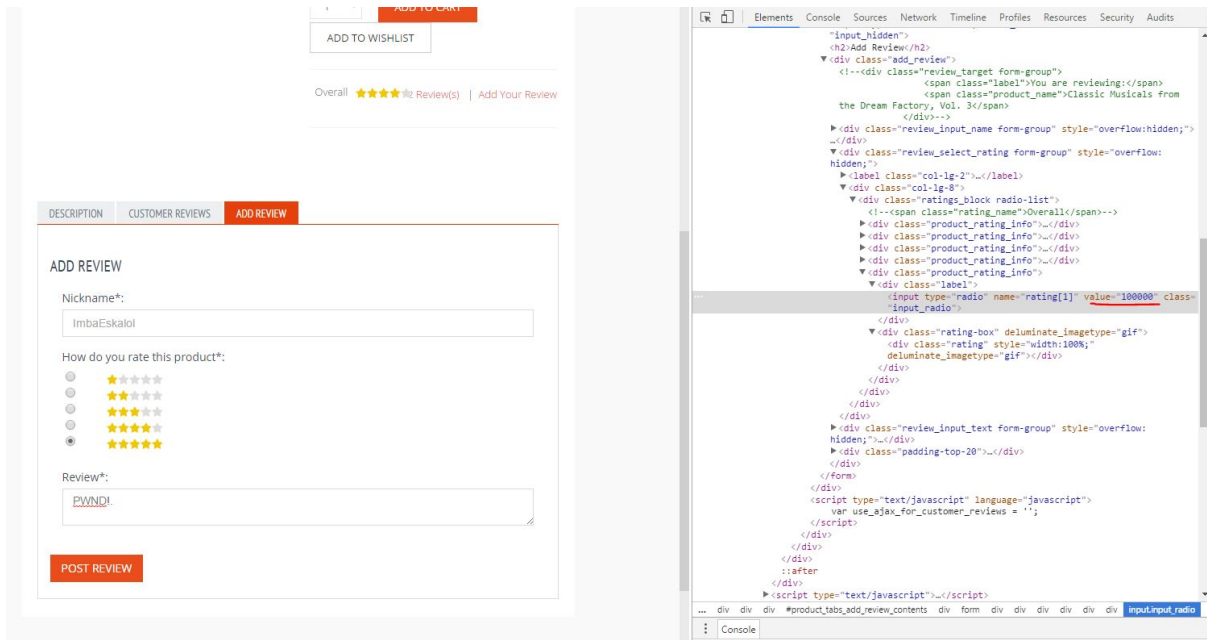


Image 2: Result

SALE

## CLASSIC MUSICALS FROM THE DREAM FACTORY, VOL. 3

**\$23.99** ~~\$26.70~~

A never-completed stage musical was the source for the MGM superproduction *Born to Dance*. The plot is another three-sailors-on-leave affair, with Ted (James Stewart), Mush (Buddy Ebsen) and Gunny (Sid Silvers, who also co-wrote the script) romancing the eminently romanceable Nora (Eleanor Powell), Peppy (Frances Langford) and Jenny (Una Merkel).

1

ADD TO CART

ADD TO WISHLIST

Overall ★★★★★★★★★★★★★★★★★★★★

DESCRIPTION	CUSTOMER REVIEWS	ADD REVIEW
ImbaEskalol, Apr 19, 2016 PWND!.	★★★★★★★★	
<alert("XSS");//<, Apr 19, 2016 <alert("XSS");//<	★★★★☆	

**Expected result:** When an invalid number of stars are sent, the server should not accept it.

**Actual result:** The server accepts it.

**Change history:** n/a

**References:** <https://www.exploit-db.com/docs/39690.pdf> section 4.2

**Assigned to:** Avactis

**Incident resolution:** blank (*suggestion: Validate that the stars are between 0 and 5.*)

## Requirement 2 - Automated tests and grouping

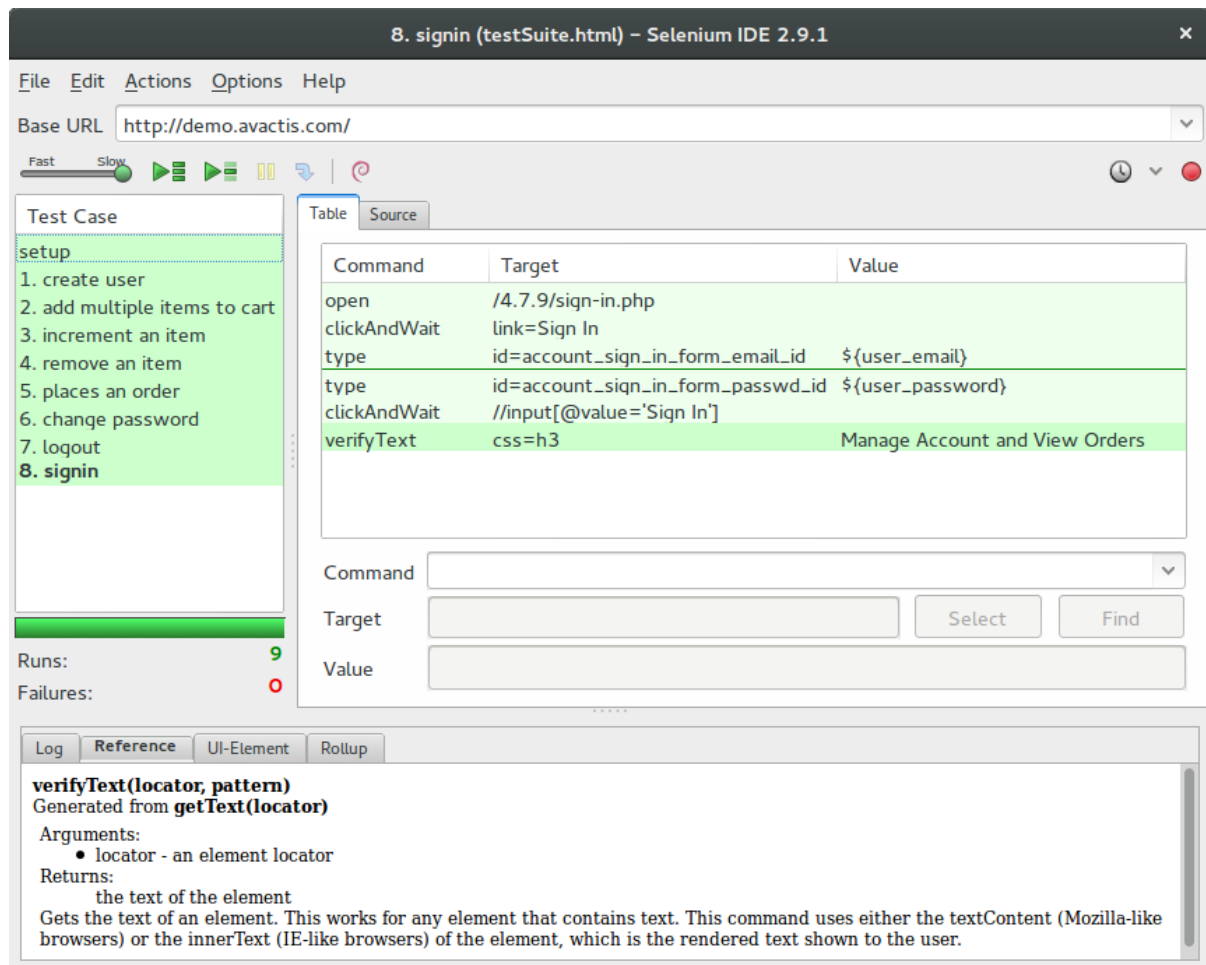
### Notes

- If the server has load problems the tests might timeout and fail.
- We have created a setup "test case" which just acts a setup function would do in other test frameworks. It uses selenium's store command in addition to some javascript to create user data, that we use for registration and validation.
- The sign-up test was difficult at first glance because the registration data needs to be unique each time (at least until the db resets). This is solved by using javascript in the selenium tests to generate random data.
- We have used javascript in the tests to make the more dynamic and depend less on string specific data (e.g. "KARLSTAD modular sofa" or "\$828.00")
- For the checkout test, we ran into a problem with the verification of billing and shipping address. The html input tag id's were random, and selenium would by default select the id as a selector. To solve this we manually changed the selector from id to the name attribute.

### Test ordering

The tests are ordered after the use case flow. This is important because most of the tests depend on the state created by the previous tests, e.g. you cannot login without being registered and you cannot log out without being logged in. The ordering is: create user -> increment an item -> remove an item -> place an order -> logout -> sign-in.

## Test screenshot



## Requirement 3 - Transitioning manual to automated tests

### Traceability matrix

In the assignment there is a list of uses cases, we have divided those up into the requirements below. In addition we found out by coincidence from friends that "change password" was a requirement, even though it does not say so in the assignment pdf (!). So we added that too and made tests for it.

1. Create user.
2. Sign in.
3. Reset password.
4. Select products and place in cart.
5. Increment number of an item in shopping cart.
6. Remove item from shopping cart.
7. Checkout the shopping cart and place order.
8. Logout.
9. Change password.



	Reqs	1	2	3	4	5	6	7	8	9
Test cases		1	1	1	1	1	1	1	1	1
1	1	x								
2	1				x					
3	1					x				
4	1						x			
5	1							x		
6	1									x
7	1								x	
8	1		x							
9	1			x						

For all (excluding password reset) manual tests, we created an identical automated test.

## Coverage

Our automated testing coverage is only partial. Excluding password reset our automated testing has full coverage in relation to the manual tests and requirements.

## Tests not automated

We have not automated password reset. Reset of password is not possible to automate without integrating email into test. Changing the password when already logged in is not sufficient as that is \_not\_ the same as resetting the password. The purpose of password reset is when the user no longer remember his/hers password and cannot gain access to the system.