INF3121 - Assignment 2

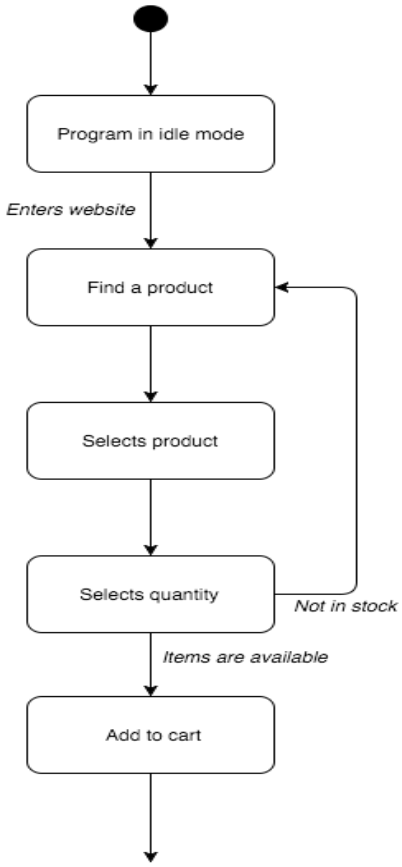
Requirement 1 – Manual tests, test design, bugs:

Test title	Objective of the test	Pre-conditions	List of steps	Post-conditions	Expected results
1. Registration Log-in Password resetting	Registration of account Log-in functionality Resetting password	---	1. Register account 2. Logs-out 3. Resets Password 4. Logs-out 5. Logs-in	The user was registered User was logged out. User was logged in. Password was resetted.	Expects the user to be able to perform these transactions.
2.Shopping	Selects products. Places them in the cart.	Expects the online-shop to have a variety of products. And the user must be logged in.	1. Find a product. 2. Select it and preview 3. Add to cart	Product is added to cart.	Find products and adds them to cart.
3.Checks out cart	Takes the products in the cart and checks them out.	Products in cart and starts in .	1. press the checkout button. 2. Follow the given steps to fulfill the transaction 3. Choose the shipping option 4. And press continue checkout (shipping and etc)	The information which is written is saved.	The information is saved and the cart is ready for finalizing the order.
4.Modifications in cart	Change number of item on items in cart	In cart and has products in cart.	1. Select item to be changed 2. Adds the quantity of an item 3. Removes a quantity 4. Removes a product	---	The cart is updated.
5.Finalize order	Place order Get confirmation and finalize	Has done the routine for checkout.	1. Press place order button 2. And wait till you get the order verification.	---	The order is finalized and the user gets a confirmation from the site.
6.Logs out	The user logs out of the site	The user is logged in	1. Click the logout button	---	The account gets logged out

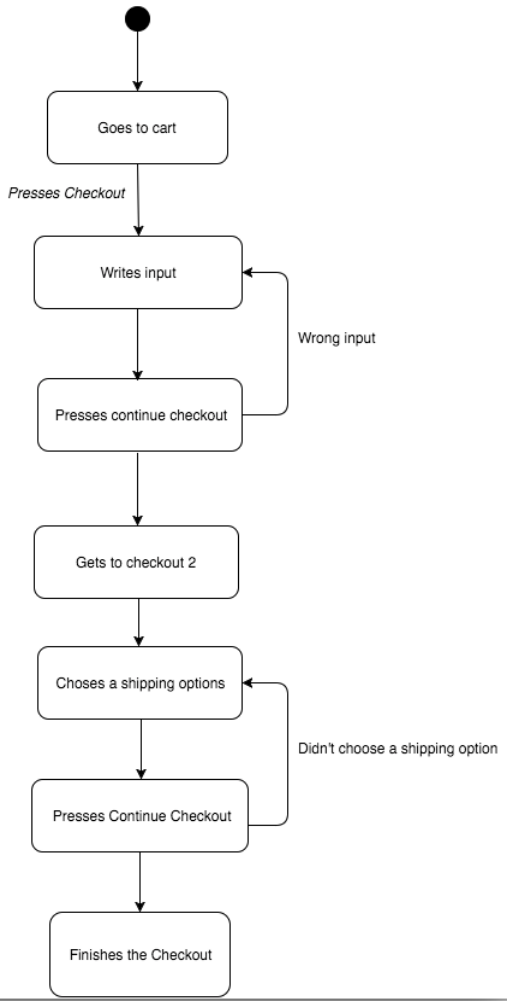
Test 1: Registration, Log-in, Password resetting

Conditions				
New user	True	False	False	False
Valid username	True	True	True	True
Valid password	True	True	Fail	True
Actions				
Register a new account	✓	✓	X	X
Failed registration message	X	X	✓	X
Login successful	✓	✓	X	X
Resetting password	✓	X	-	✓
Failed resetting password	X	✓	-	X

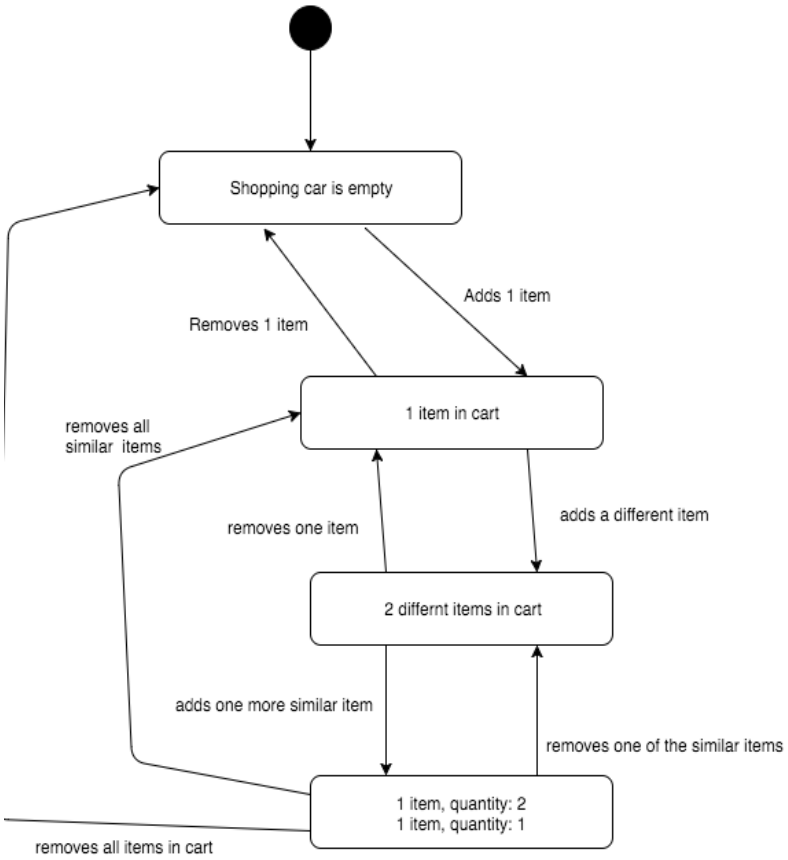
Test 2: Shopping



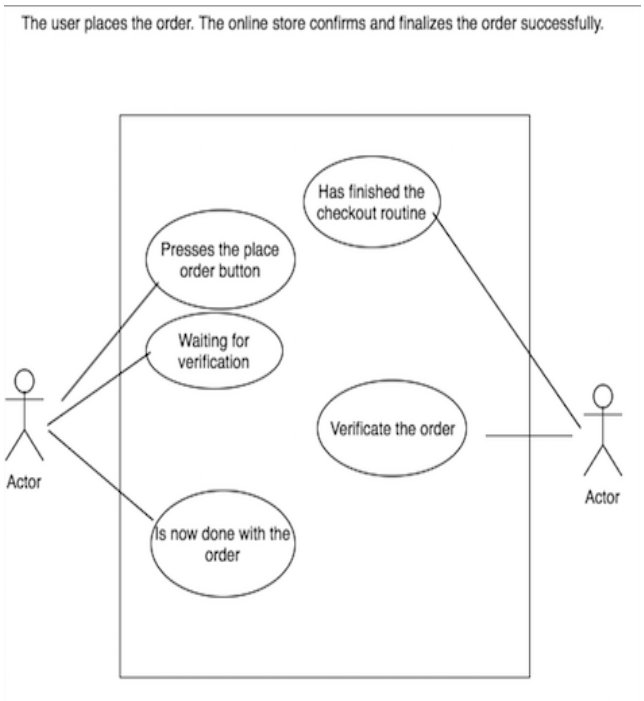
Test 3: Checks out cart



Test 4: Modification in cart



Test 5: Finalize order



Test 6: Logging out

Conditions				
Logged in	True	False	True	False
Items in cart	True	True	False	False
Actions				
Login successful	✓	X	✓	X

Incident reports:

Bugs:

Details of the incident report may include (cf. IEEE 829):
Date: **17.04.16**
Project: **INF3121 – Manual testing and test automation**
Programmer/Tester: **Johan**
Program/Module: **Login/Cart**
Build/Revision/Release: **X**
Software Environment: **Mac osX**
Hardware Environment: **Macbook retina 13'**
Status of the incident :**Unsolved**
Number of Occurrences: x Severity: x Impact x Priority: Important

Detailed Description: **When you log out the items in your cart won't be removed, so that if you log back in again the items will again be added to the cart which doubles the original items.**

Expected result / Actual result: Gives the cart
Change history : **X**
References (including the identity of the test case specification that revealed the
Problem: **X** Assigned To:**X**

Error:

Incident Resolution: Not add the previous items with the current once in cart, and not make the amout of itmes increase with n=n*2, each time the user log out/in.

Incident reports

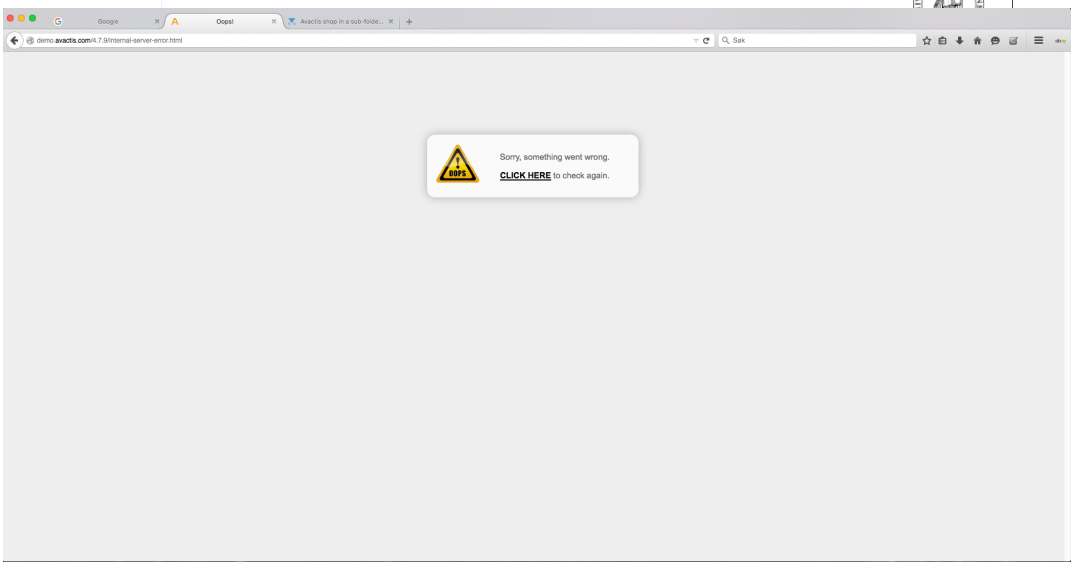
Details of the incident report may include (cf. IEEE 829):

Date: 24/04/16
Project: Assignment 2
Programmer: _____ Tester: Tanu & Brish

Program/Module: Registration Build/Revision/Release: x
Software Environment: Osx El Capitan Hardware Environment: Macbook pro 2015
Status of the incident Unsolved

Number of Occurrences: x Severity: x Impact x Priority _____
Detailed Description: Given below (logs, databases, screenshots)
Expected result / Actual result: Expected the program to handle the incorrect input
Change history _____
References (including the identity of the test case specification that revealed the problem_____

Assigned To: _____
Incident Resolution: _____



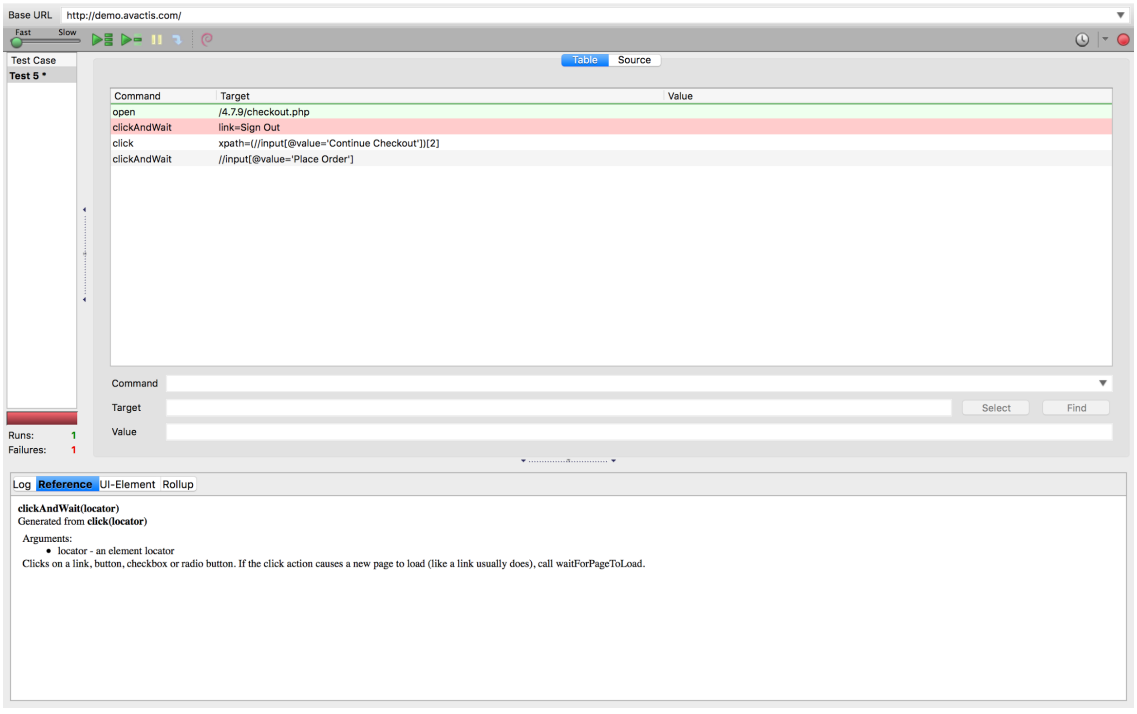
Requirement 2 - Automated tests and grouping:

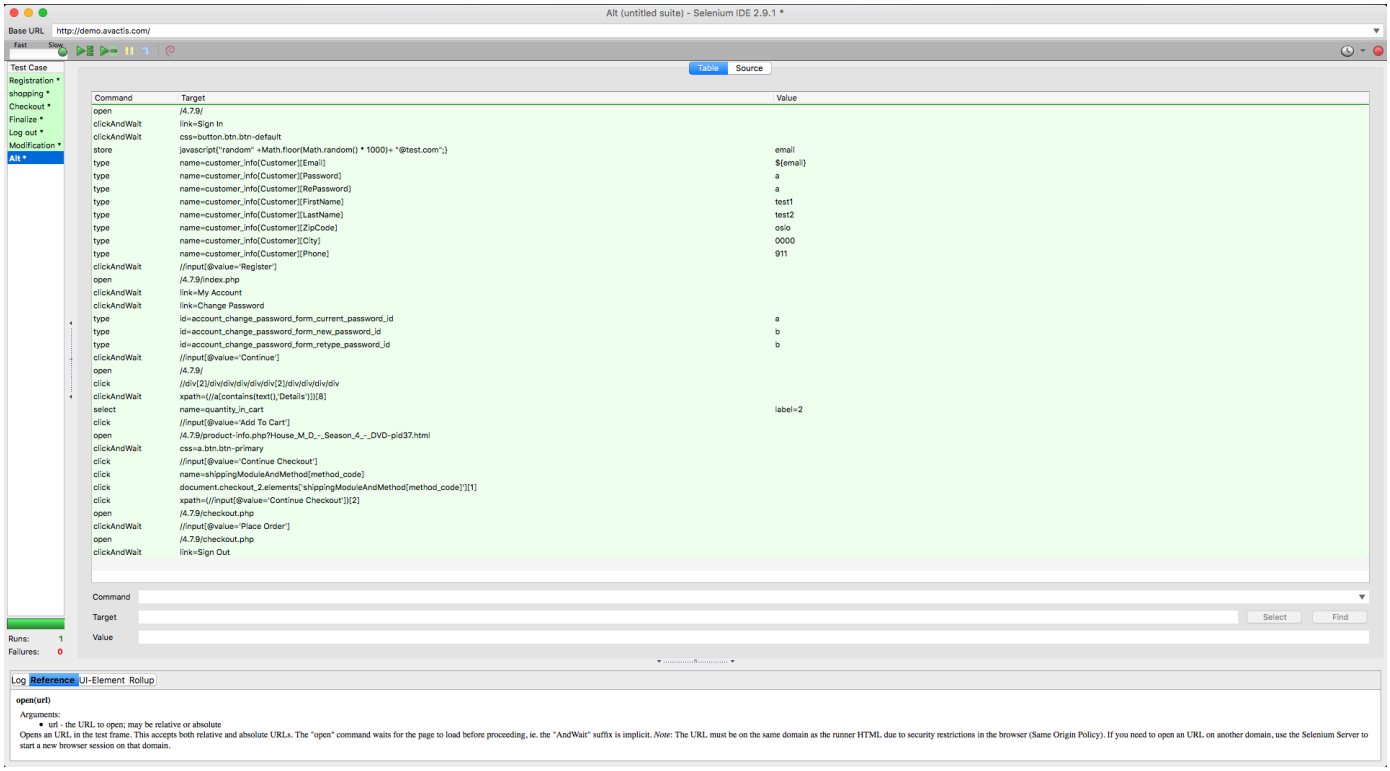
GitHub Link: https://github.com/brish100/Assignment2

A thing we have to point out is that in our tests, we fulfilled our preconditions in each script so that we can really run each of our test multiple times one by one.

The order we chose is the standard order, by registering first and logging out at the end. The modification is at the bottom, but it doesn't have any influence on the test with the way we have implemented the tests. So our order in our scripts don't really matter since they have already been set so that they can be runned as much as you like. Screenshots are given under.

What we have done in our automated tests:
Making smaller scripts to certain modules, won't work without the pre-conditions. So optimally we would in each of the tests combine them with their pre conditions so that they can be run separately. And with that have fully functional separate scripts. Right below, we can see the failed script. This is the last step "log out", just running one script without preconditions, as you can see it fails. This is because the preconditions were not implemented so we were not logged in before we tried to log out.





Here is a picture of all the test cases runned in a script. We can see here that the whole project got automated when we merged everything into one script. With this we made a fully functional “test suite” which does all the tests automatically. :)

Requirement 3 – Transitioning manual to automated tests:

All the manual tests are now covered by automated tests. The automated test is in a way partial, since they build up one another, but with the implementation of the automated tests we have in a way made it possible to test them individually.
But this again needs the precondition of the test be done in the script the given test is runned on.

Traceability matrix
In this traceability matrix we show where the different test requires each other, but for how the tests interrelate between the manual and the automated tests is written at the bottom.

	TC 01	TC 02	TC 03	TC 04	TC 05	TC 06
User Requriments						
SR-1.1 A user register account	✓					
SR-1.2 User logs in	✓				✓	✓
SR-1.3 A user try resetting password	✓					
SR-2.1 User find a product		✓	✓	✓	✓	
SR-2.2 User finds product and preview it		✓	✓	✓	✓	
SR-2.2 user adds product add to cart		✓	✓	✓	✓	
SR-3.1 User presses Check out button			✓	✓	✓	
SR-3.2 Choose shipping order			✓	✓	✓	
SR-4.1 A user adds a product				✓	✓	
SR-4.2 A User Removes product				✓	✓	
SR - 5.1 - User places a order					✓	
SR - 6.1 User logs out						✓

From manual to automated:

Test 1(Sign-in: Registration, Password reset):
Change in the Script:

```
<tr>
    <td>store</td>
    <td>javascript{&quot;random&quot; +Math.floor(Math.random() * 1000)+ &quot;@test.com&quot;;}</td>
    <td>email</td>
</tr>
<tr>
    <td>type</td>
    <td>name=customer_info[Customer][Email]</td>
    <td>${email}</td>
</tr>
```

To automate the registration phase, we had to randomly generate the email, so that the same email won't be inserted multiple times. Which would lead the site to display an error message. With this kind of tweaks in the scripts onwards we could manage to automate the problems to a good degree.(an example can be seen in the screenshot above). The resetting of the password part was also a bit tiresome, since it required the password to not be one of the old passwords. But we managed to cover this by making new accounts for each case so that we didn't have to think about all that. (this might look like a flaw, but in matter of automation we think this compensate for that in a good degree).

So in case of test 1 we managed to automate the tests in a partial/total manner, since some parts as the resetting of the password was not fully automated. But the automation of the registration phase was fully automated.

Test 2 (Shopping):
This part got automated by choosing various types of products and then selecting the quantity and adding the item to the cart. This coverage was total, in a way where the same objects will be tested repeatedly, but it won't test different items without proper change in the scripts which we don't have.

Test 3 (Checks out cart):
In our scenario the user is always logged in, so all the program have to do is to click on the buttons. This is a fairly simple function to test and the test automation was total.

Test 4 (Modification in cart):
We can say that it is possible, since by analysing the script we can see that it the information is stored in an array. And with that we can use that information to manipulate the given items in the cart to different numbers. But because this will require more knowledge of the script in use, we could not produce a fully functional automated test of that. In other words, the test was partial.

Test 5 (Finalize):
This case got automated easily too, it just had to click on a button and then wait for a confirmation. Finalize order was total.
Test 6 (Logs-out):
User logs out only requires the user to be logged in. Thereby the test automation was total, for all cases since it the only possible way this fails is when a account is not logged in.

We automated all the tests in a way, but it still lacks some parts to make it fully automated! :)