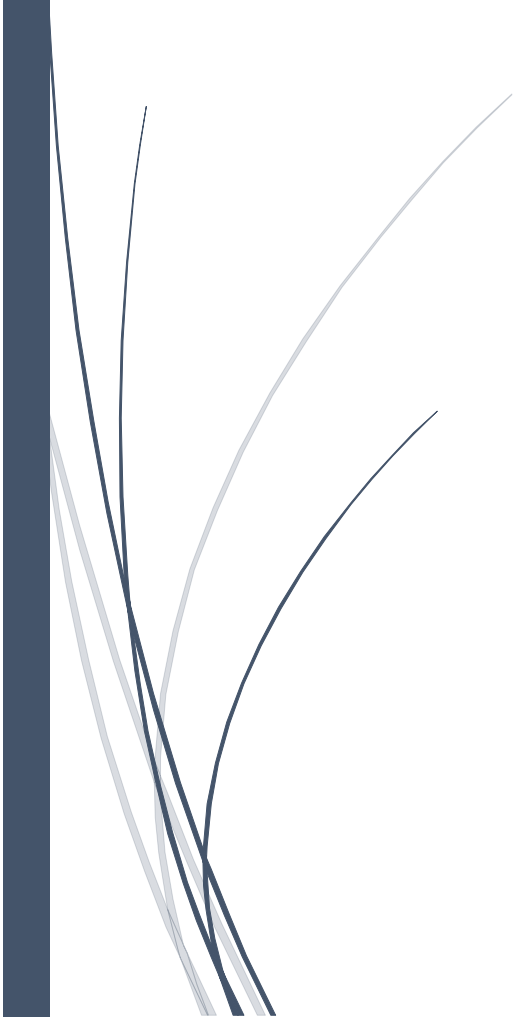
5/3/2016

Assignment 2

Manual tests, test design, bugs, automated tests and transition.



Kabza, Ewa

INF3121 – SOFTWARE TESTING

Manual tests, test designs and bugs

Registering a new user: testCase01

Based on previous experience (creating many accounts on web shops) I would use error guessing technique. I assumed that if all information were correctly filled out, then the unit would work and register a new user.

The purpose of this test is to check if a new user can be created. We assume that a user provides correct information and format he is supposed to fill out in the scheme.

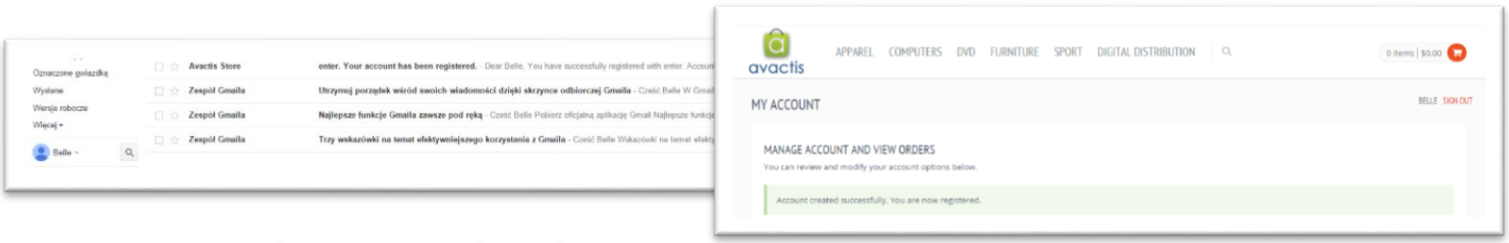
Preconditions: -

Test steps:

1. User goes to the website: http://demo.avactis.com/4.7.9/;
2. User clicks on "Register" under New Customer section;
3 User is sent to the website with a form he needs to fill out: http://demo.avactis.com/4.7.9/register.php;
4 User fills out all required information: e-mail address, password, name, surname, etc.;
5 User clicks on "Register".

Expected results: The account is created and users receives a feedback.

Actual result: The new account was created. User received a feedback on the site, as well as he was sent an information e-mail.



Post conditions: User is logged in.

Status: Positive, test passed.

Logging into website: testCase02

In this case, I am using a decision table because it allows to combine different inputs and presents clearly cause- effect relationship.

	A	B	C	D
Conditions				
Requested login	yes	yes	yes	yes
Valid e mail address inserted	yes	no	yes	no
Valid password inserted	yes	no	no	yes
Actions				
Logs into the website	TRUE	FALSE	FALSE	FALSE
Error message displayed	FALSE	TRUE	TRUE	TRUE
Recovery credentials offered (Forgot password)	TRUE	TRUE	TRUE	TRUE

The purpose of the test is to check if a user can log successfully into the website giving a correct e-mail address and password given during the registration.
Preconditions – user has to be registered on the website.
Test steps:

1. User goes to the website: http://demo.avactis.com/4.7.9/;
2. User inserts the correct e-mail and password;
3. (*Optional*) User can click on "Keep me signed…" if he does not want to enter data every time he logs into;
4. User clicks on "Sign in".

Expected results: User will be redirected to his account.
Actual result: User sent to "My account" page.
Post conditions: User is logged in.
Status: Positive, test passed.

Resetting a password: testCase03

In this case, I also use decision table because it visualize the combination of inputs (mail and password) while password is forgotten. In this case, it works 'yes-no' / 'true-false' combination.

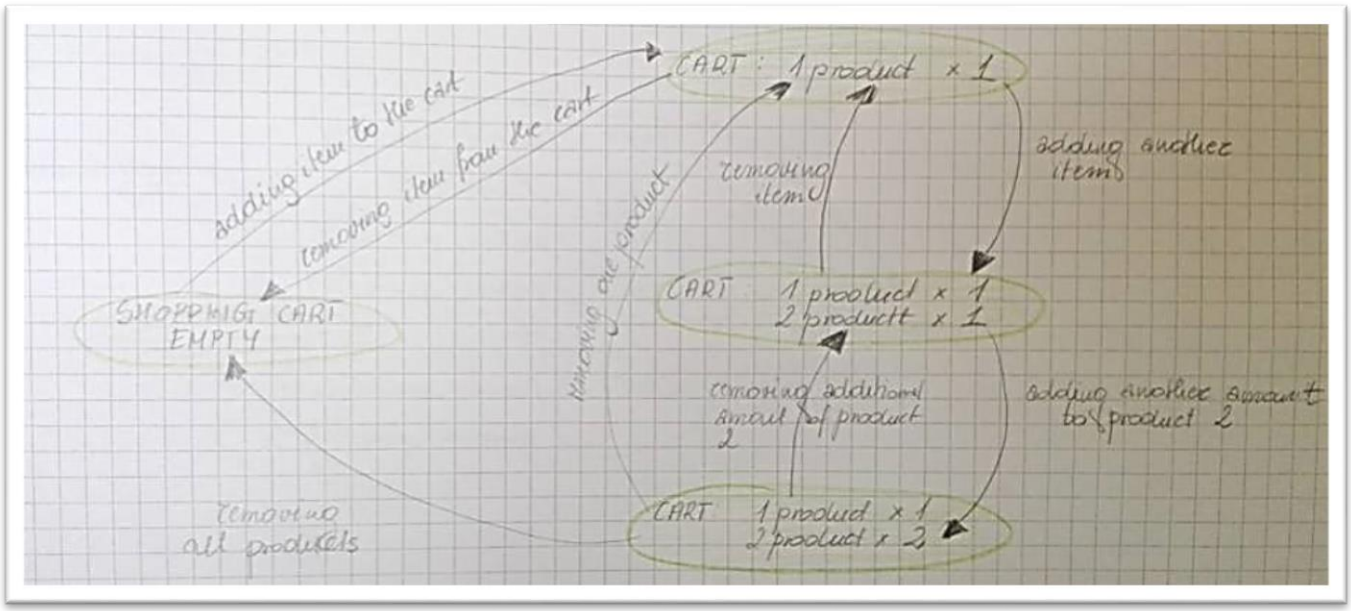
	A	B
Conditions		
Correct e mail provided	yes	no
Forgotten password	yes	yes
Actions		
Recovery e-mail sent	TRUE	FALSE
Error message appears	FALSE	TRUE
Password - reset	TRUE	FALSE

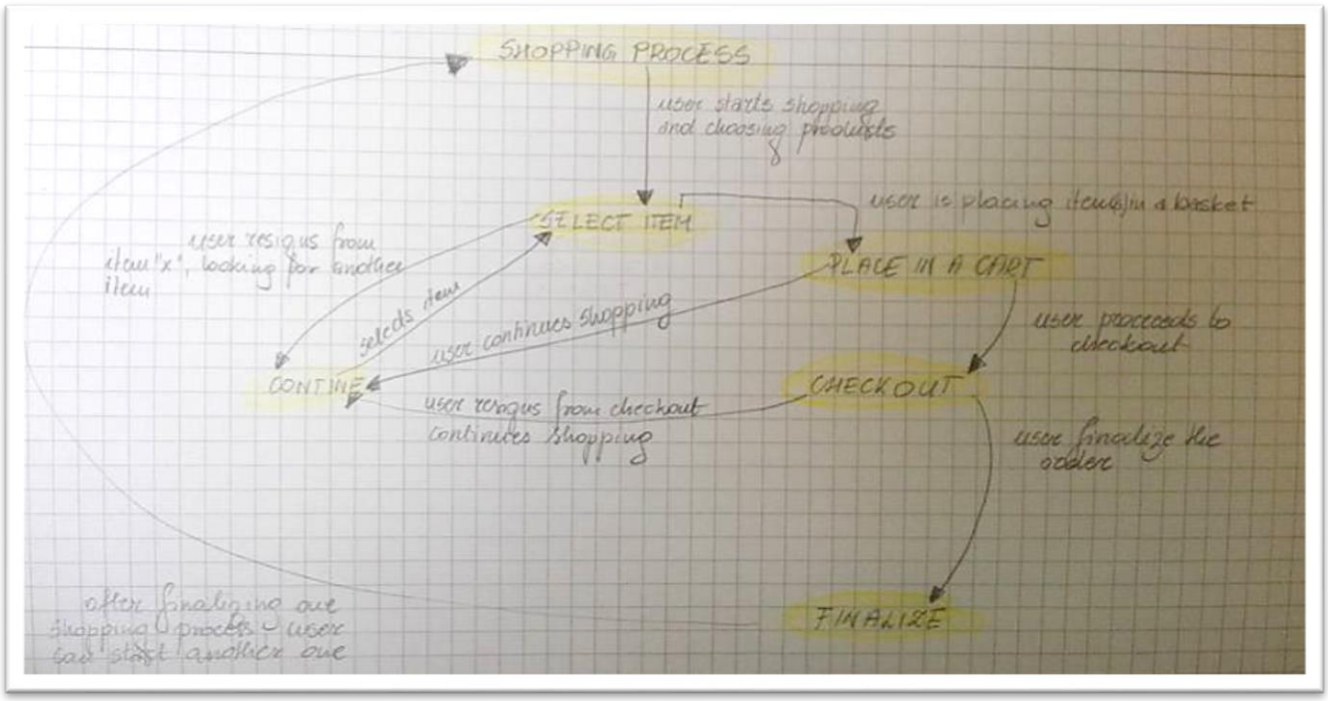
The purpose is to check if a user can restore forgotten password by clicking "Forget password" and providing correct required data.
Preconditions: user has to have an account on the website.
Test steps:

1. User goes to the website: http://demo.avactis.com/4.7.9/;
2. User clicks on "Forgot password";
3. User is redirected to the recovery page and enters the correct, requested e-mail address;
4. User clicks on "Continue".

Expected result: The recovery password e-mail is sent to the e-mail address provided by user.
Actual result: The recovery e-mail is sent to the user. User is asked to click on the provided link and he is redirected to the recovery password website. There he is asked to choose a new password, repeat it and click on "Continue". Password is reset.
Post conditions: User is logged to the website again.
Status: Positive, test passed.

To present shopping process and user's interaction in the basket, I have chosen state transition testing diagrams. I think diagrams present well the flow inside the online store.





Selecting products: testCase04

The purpose of this test is to check if user can easily and successfully select products which are automatically added to the basket.
Preconditions: User is logged into website.
Test steps:

1. User clicks on the main category of products he is interested to purchase;
2. (Optional) user chooses sub-category (if applicable);
3. User clicks "Add to cart" below the item he wants to purchase.

Optional steps (more detailed selecting):

4. User clicks on the item picture
5. User is redirected to the item site
6. User can choose amount of items by clicking on the amount button
7. User clicks "Add to cart".

Expected result: The item is added to the basket. The price is visible for user.
Actual result: Item is added to the basket. The price is updated.
Post-conditions: Item in the shopping cart.
Status: Positive, test passed.

Continuing shopping: testCase05

The purpose is to test if a user can continue shopping after adding an item to the shopping cart.
Preconditions: User has to add one item to shopping cart.
Test steps:

1. User can choose again category (and subcategory);
2. User clicks on "Add to cart" under the desired product.

Expected result: Another item is added to the basket. User is informed how many items are in the basked and what is the total price.
Actual result: Another item is added to the basket. User is informed how many items are in the basked and what is the total price.
Post conditions –
Status: Positive, test passed.

Checkout and finalizing order successfully: testCase06

The purpose of this test is to check if user can perform check out successfully after finishing shopping
Preconditions: User has to have some items in the cart
Test steps:

1. User clicks on "Checkout" on the right corner;
2. User is sent to "checkout" page;
3. User checks if all data is inserted correctly (address, name, phone, etc.);
4. If correct and desired, user clicks on "continue checkout";

5. User is sent to shipping and payment side;
6. User chooses preferred option of delivery;
7. User clicks on “continue checkout”;
8. User clicks on “place order” if everything is correct;

Expected results: User is redirected to the confirmation page. From there he can choose to continue shopping again, or sign out from the page. The basket is reset.

Actual result: User is redirected to the confirmation page. From there he can choose to continue shopping again, or sign out from the page. The cart is reset. Confirmation e-mail sent to the user.

ERROR: If user by mistake places more than one item of the same product (for ex: 2x DVD X) and he does not realize it during shopping, he cannot remove or change the quantity of product during checkout. He needs to quit checkout, go back to the basket and remove it from there.

Status: Semi positive, semi negative. Test passed, but the lack of change quantity/remove option.

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

Date: 03.05.2016

Project: Assignment 2

Programmer: XY Tester: AB

Program/Module: Checkout Build/Revision/Release: 01.01.2016

Software Environment: Windows 8 Hardware Environment: Lenovo Yoga 500

Status of the incident: Not solved

Number of Occurrences: 1 Severity: 1 Impact: 1 Priority

Detailed Description: While checking out user cannot change the amount of item. He can only remove the product.

Expected result: Only the desired amount of items removed.

Actual result: All instances of item removed

Change history ……. To be created…..

References (including the identity of the test case specification that revealed the problem xxxx

Assigned To: ZU

Incident Resolution: In progress

Changing numbers of items: testCase07

The purpose is to test if user can reduce number of items by removing them from basket.
Preconditions: User has to have some items in the basket.
Test steps:

1. User clicks on my cart;
2. The list of orders is presented;
3. (Optional) User clicks on the amount button – if wants to remove duplicated instance of the item
4. User clicks on "x" if wants to remove item from the cart

Expected results: The desired item is removed.
Actual result: The desired number of items removed.
Status: Positive, test passed.

Logging out: testCase08

State table:

Status	Input	Next status
Logged in	Click on 'sign out'	Logged out
Logged out	Click on 'sign in', insert correct e mail address and password	Logged in

When the status is set as logged in, a user has to click on "sign out" to successfully log out from the website. When the status is logged out, user has to click on "sign in" and providing the correct data (e mail and password) to log in to the website again.

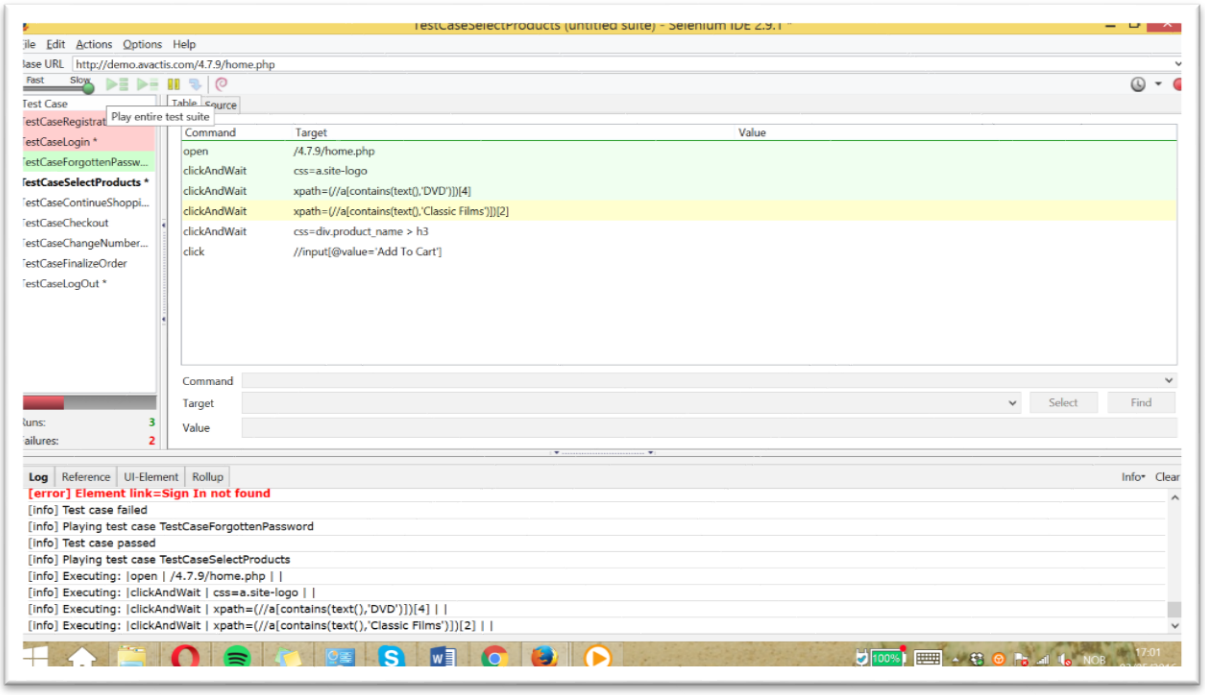
The purpose is to check if user can log out successfully from the website.
Preconditions: user has to be logged into the website.
Test steps:

1. User is logged into the website;
2. User clicks on "Sign out";

Expected result: User is logged off from the website.
Actual result: User is logged off from the website.
Post conditions: User is redirected to main – log in page.
Status: Positive, test passed.

Automated tests and grouping

Firstly, I have placed all automated test like they were done in assignment. However, tests were crushing all the time. I assumed that they were not executed properly because it was not the natural order of user's performance on the website. What I mean is that if user registered on the website, he does not immediately log out, forget password and ask for reset.



(NOTE: Due to some problems with website itself I needed to redo tests and by mistake I gave them different name – they check the same units, but nomenclature is different. Because my GitHub doesn't work since last time, I recorded tests using screen recorder and uploaded them to YouTube)

To check my assumption, I have decided to split test cases into logical step-by-step processes and recorded separated videos:

Video 1 presents following order of tests:

1) TestCaseLogIn – user logs into the website;
2) TestCaseSelectingItems – user does shopping by selecting item and putting them to the cart;
3) TestCaseChangingQt – user changes quantity of items in the basket and continues shopping;
4) TestCaseCheckoutAndFin – user proceeds to checkout and finalizes the order/placing the order;
5) TestCaseSignOut – user signs out from the website.

https://www.youtube.com/watch?v=7FPrbVWzRA0

Video 2 presents resetting password test:

https://www.youtube.com/watch?v=e6EwC8GP7GY

Video 3 presents registration test:

https://www.youtube.com/watch?v=fmXIbYR6_NY

(*I do not know why but at the beginning it redirected me to the website confirming the registration of account. But after reporting the info on Piazza that website does not work, the info about account already exists but test does not fail.)*

I have impression that grouping tests by their functionality helps to conduct more reliable tests, because it imitates user's behavior on the website. In addition, it also allows to keep to minimalize errors which does not come from the website itself, but which are consequences of bad/poor testing.

Transitioning manual to automated tests

Manual test	Automated test	Comment
TestCase01	TestCaseRegistration	Manual test was successfully automated. The coverage was total and a new user account was created. *On the video it shows the message that the account is already taken, but the test does not fail.*
TestCase02	TestCaseLogin	Successfully automated. Logged into website. Coverage total
TestCase03	TestCaseForgottenPassword	Test failed. I think that the problem was that to reset password user has to go to Gmail and click on recovery link from there. And, Selenium is not able to jump from internet store website to Gmail without crushing.
TestCase04	TestCaseSelectingItems	Those tests cases performed desired actions, however test steps were not held 1:1 (manual: automated). The manual tests seemed to be more structured then automated ones.
TestCase05	TestCaseContinueShopping	
	TestCaseChangingQt	
TestCase06	TestCaseCheckoutAndFin	
TestCase07	TestCaseLogOut	Successfully automated. Logged out from website. Coverage total