

Assignment 1

Introduction to Testing and Bug Tracking

Due Date: Jan. 16, 2019

Yasaman Amannejad, 2020

TABLE OF CONTENTS

Introduction to Testing and Bug Tracking	1
1 INTRODUCTION.....	3
1.1 SOFTWARE UNDER TEST (SUT)	3
1.2 REPORTING TOOL.....	3
1.3 GROUP WORK	3
1.4 TESTING PROCESS.....	3
2 PREPARE YOURSELF	4
2.1 EXPLORATORY TESTING	4
2.2 MANUAL SCRIPTED TESTING	4
2.3 REGRESSION TESTING	4
2.4 BUG TRACKING SYSTEMS	4
4 SYSTEM UNDER TEST	5
4.1 PETSTORE APPLICATION.....	5
4.2 INSTALLATION	6
5 BUG TRACKING SYSTEM	6
6 INSTRUCTIONS	8
6.1 EXPLORATORY (MANUAL NON-SCRIPTED) TESTING	8
6.2 MANUAL SCRIPTED TESTING	8
6.3 REGRESSION TESTING (VERIFICATION OF DEFECT FIXES)	8
6.4 SUMMARY	9
8 SUBMISSION	10
9 REFERENCES.....	9
APPENDIX - FUNCTIONAL TEST SUITE FOR MANUAL SCRIPTED TESTING	11

Interesting Quotes on Defect (Bug) Reports

- “Bug Advocacy: Lesson 59 - Take the time to make your bug reports valuable”
- “Bug Advocacy: Lesson 57 - Make your bug report an effective sales tool”
- “Bug Advocacy: Lesson 62 - Report perceived quality gaps as bugs”
- “Bug Advocacy: Lesson 67 - Report defects promptly”
- “Bug Advocacy: Lesson 68 - Never assume that an obvious bug has already been filed”
- “Bug Advocacy: Lesson 72 - Minor bugs are worth reporting and fixing”
- “Bug Advocacy: Lesson 84 - Never exaggerate your bugs”
- “Bug Advocacy: Lesson 85 - Report the problem clearly, but don't try to solve it”
- “Bug Advocacy: Lesson 94 - Verify bug fixes promptly”

From Cem Kaner's book “Lessons Learned in Software Testing”

1 INTRODUCTION

This lab is an introduction to some of the concepts inherent to software testing. Specifically, students will learn the fundamentals involved in the following areas of testing:

- Hands-on experience in testing a software system
- The difference between exploratory (manual non-scripted) testing versus manual scripted testing, and regression testing
- Defect tracking systems, processes and practices
- Different phases of defects' life cycle.

1.1 SOFTWARE UNDER TEST (SUT)

Here are the links for the two versions of the Petstore application that you will be using for this assignment: **version 1.0:** <https://github.com/Yasaman-A/Assign1-1.0.git>

version 1.1: <https://github.com/Yasaman-A/Assign1-1.1.git>

1.2 REPORTING TOOL

Bug tracking tool – Bugzilla is available at <http://35.224.228.161/bugzilla/>. For more information, see Section 5. If you have entered your group information in the spreadsheet, you should have received username and password to login in the system.

Note: you may experience a glitch during the login to Bugzilla. You may need to repeat the login process twice to enter to the system.

1.3 GROUP WORK

This lab work is a group activity. Students will complete this lab in groups of 2-3 students. Each team member will receive a username and password to use the Petstore application, and also the Bugzilla tool.

1.4 ASSIGNMENT PROCESS

Follow the instructions in the Google cloud activity to setup version 1.0 of the PetStore Web application. Share the URL with your team members to start testing this version of the application. Complete the exploratory testing (Section 6.1), and scripted testing steps (Section 6.2), and enter the issues you identified in the bug-tracking tool. When you finish your tests, export all your bugs into a CSV format. (Include all the important columns, using the *Change Columns* feature in Bugzilla report.) Next, deploy version 1.1 of the application and conduct your regression tests (Section 6.3). When you complete your tests, export the new status of your system into a new CSV file.

Complete the assignment report. Submit your report, the CSV file from version 1.0, and the CSV file from version 1.1 in Blackboard.

2 PREPARE YOURSELF

2.1 EXPLORATORY TESTING

Exploratory testing is a commonly used term for software testing performed without documentation. Exploratory testing is performed manually and usually without any test script.

The tests are intended to be run only once, unless a defect is discovered. Exploratory testing has been criticized because it isn't structured, but this can also be a strength. By doing exploratory testing, major issues can be found quickly. It is performed with improvisation; the tester seeks to find bugs with any means that seem appropriate. Exploratory testing is most often used as a complement to other types of testing.

2.2 MANUAL SCRIPTED TESTING

Manual scripted testing is the oldest and one of the most rigorous types of software testing. In this particular type of testing, test cases are designed and reviewed by the team before executing it. There are many variation of this basic approach, test cases can be created at the basic functionality level or they can be created at the scenario level.

Value of the scripted testing has been questioned by some experts in the field. They claim that scripted manual testing closes the mind of tester and inhibit them to use their creativity. Also, this approach is very heavy on the documentation and requires considerable amount of resources to create the test scripts in the first place and they often get outdated because of the inevitable changes in the system.

Despite these drawbacks, manual scripted testing is used in many organizations of all sizes. They make test cases repeatable and easy enough for a new person to come on board and start testing with minimum supervision. Manual scripted testing is also used in places where contractual agreement states that written specification of the software must be met for the successful implementation of the project. Scripted test cases might be useful where tests are used for the benchmarking purpose and tests have to be executed exactly in the same way, every time.

2.3 REGRESSION TESTING

Regression testing is any type of software testing that seeks to uncover new errors, or regressions, in existing functionality after changes have been made to a system, such as functional enhancements, patches or configuration changes.

The intent of regression testing is to ensure that a change, such as a bug fix, did not introduce new faults. One of the main reasons for regression testing is that it's often extremely difficult for a programmer to figure out how a change in one part of the software will echo in other parts of the software. Regression testing can be done in both manual and automated testing fashion. In this assignment, we will do only manual regression testing.

2.4 BUG TRACKING SYSTEMS

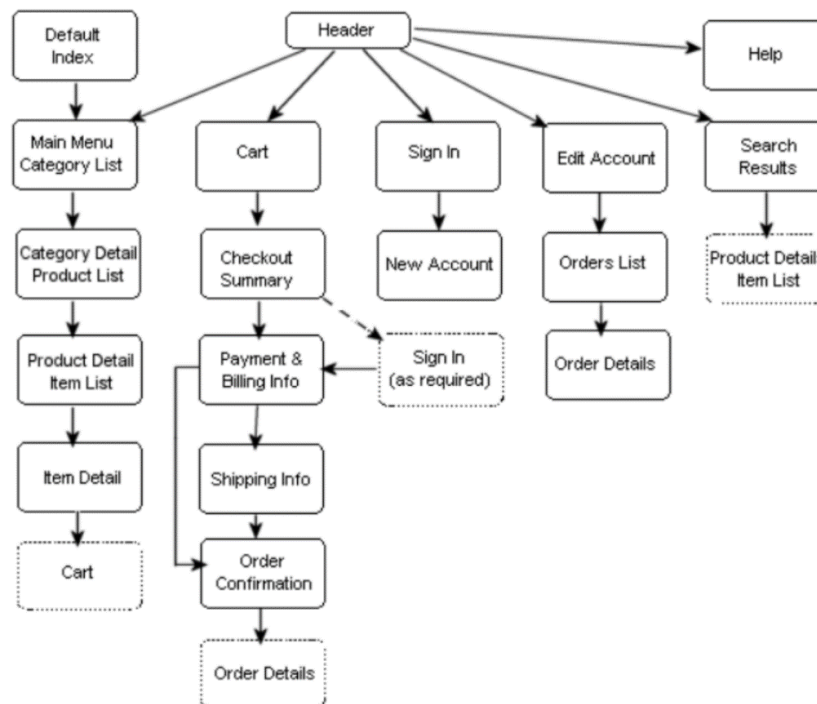
There are many issue tracking systems out there, including bug (defect) tracking systems, help desk and service desk issue tracking systems, and asset management systems. Our focus in this assignment and the course are on bug (defect) tracking systems, and not the others. A [nice list](#) has been provided in the Wikipedia.

4 SYSTEM UNDER TEST

4.1 PETSTORE APPLICATION

The SUT for this assignment is an online shopping application, PetStore. This application was originally developed for learning Java Spring and exploring JVM tools. The application provides an e-commerce shopping cart because that is a common use case on the internet. Different items within several categories (of pets) are available for browsing and purchase. The application supports functionalities like registering a new user, editing existing user information, browsing different product items, and ordering them.

Here are the high-level functionalities of the application.

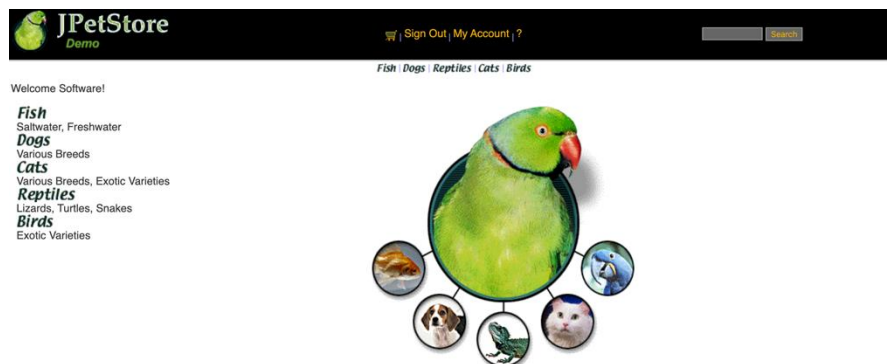


You will test two versions (versions 1.0 & 1.1) of this application in this assignment. These are two versions of the PetStore application which represent two consecutive releases of the software with bugs and bug fixes. To use the application, you can use the following account information.

Username: quality1, password: 12345678

Username: quality2, password: 12345678

Username: quality3, password: 12345678



4.2 INSTALLATION

Here are the links for the two versions of the Petstore application that you will be using for this assignment:

Version 1.0: <https://github.com/Yasaman-A/Assign1-1.0.git>

Version 1.1: <https://github.com/Yasaman-A/Assign1-1.1.git>

Follow the instructions in the Google cloud activity to setup version 1.0 of the PetStore Web application. Share the URL with your team members to start testing this version of the application.

Note 1: Please note that the application is using an in-memory database. If you re-start your application for any reason, your application will start from a fresh database, meaning that if you have created a new user account, or you have registered some orders, they will disappear.

Note 2: If you restart your cloud instance, the IP address may change. If it happens, make sure to share it with your teammate.

5 BUG TRACKING SYSTEM

Bugzilla is a robust, featureful and mature bug-tracking system. Bug-tracking systems allow software teams to keep track of outstanding bugs, problems, issues, enhancement and other change requests in their products effectively. Simple defect-tracking capabilities are often built into integrated source code management environments such as Github or other web-based or locally-installed equivalents.

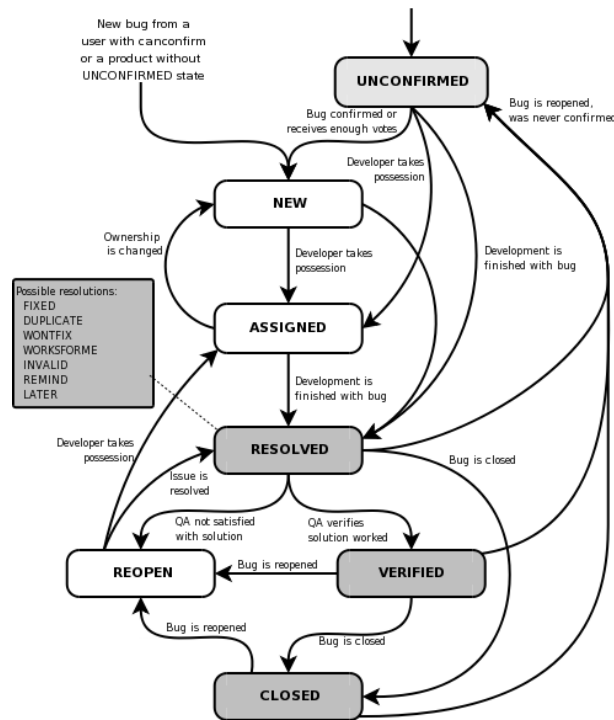
To learn more about features of Bugzilla, please visit <https://www.bugzilla.org/features/>

A tester, must record various information about a bug when recording it. The important fields in Bugzilla are described in this page: https://www.bugzilla.org/docs/2.18/html/bug_page.html

Here are some additional suggestion that may help you:

1. *Assigned To:* The person responsible for fixing the bug. You can assign your bugs to `developer@qualitylab.com`.
2. *Keywords:* You can use to tag and categorise bugs. To tag the bugs that you find in each step of the process, three keywords are provided for you: Exploratory, Scripted, Regression. You can choose an appropriate keyword, based on how you identify a defect.
3. *CC list:* A list of people who get mail when the bug changes. Email server is not setup for this Bugzilla installation, you can leave this field empty, or CC your own .qualitylab email (the email account you used to login to Bugzilla).

To learn about the life cycle of a bug in Bugzilla, please review this page: <https://www.bugzilla.org/docs/2.18/html/lifecycle.html>



Now, login to Bugzilla bug tracking system at <http://35.224.228.161/bugzilla> and familiarize yourself with the system. Use the .qualitylab email address that you have received from your instructor.

Note: You may experience a glitch when you login to the system. This might be an installation issue. If this happens, go back to the original URL (<http://35.224.228.161/bugzilla/>) and try the login process once again and you should enter to the system successfully. This may happen only during the login time and when you are inside the system, you should not experience any issue.

Bugzilla – Main Page

Home | New | Browse | Search | Search [?] | Reports | Preferences | Administration | Log out yamannejad@mtroyal.ca

Welcome to Bugzilla






File a Bug Search User Preferences Documentation

Enter a bug # or some search terms Quick Search [?]

Common Queries:

- Open bugs assigned to me (0) 
- Open bugs reported by me (2) 
- Bugs reported in the last 24 hours  | last 7 days 
- Bugs changed in the last 24 hours  | last 7 days 

You can use the “New” menu to add a bug to the system. You can search all your bugs using the “Search” window.

6 INSTRUCTIONS

6.1 EXPLORATORY (MANUAL NON-SCRIPTED) TESTING

Before beginning your test, try to come up with a high-level exploratory test plan for how you intend to test the system. Record key details of this plan, as it will be required in the assignment report. This plan could include but is not limited to, information such as: functions being targeted, the approach to be taken (test most functions a little bit, or test a few functions extensively, etc.), and how you plan to come up with test cases (test most common paths, or exceptional paths, etc.). Keep in mind that this does not need to necessarily be the best plan, as long as it is justifiable.

Reporting defects: Carry out your devised exploratory test plan for roughly half an hour. Each group needs to perform exploratory testing and record defects. Record defects as they are found following the heuristic of reporting defects promptly¹ (before forgetting the defect and its detailed conditions). You should report the defects in the bug-tracking tool. Read the lecture materials and the following reading materials to improve the quality of your bug report.

Reading Material - You can learn about what developers in the industry think make good bug reports. The Saarland University in Germany has put together a good resource of papers and talks that gauge what they often see missing from bug reports that make them less useful or take more time to fix. You can access it here: <https://www.st.cs.uni-saarland.de/publications/files/bettenburg-tr-2007.pdf>. For example, they have a nice online slide set on the quality of bug reports in the Eclipse project that you can find here: <http://www.cs.mcgill.ca/~martin/etx2007/papers/5.pdf>

6.2 MANUAL SCRIPTED TESTING

In Appendix, a basic test suite has been provided for this SUT. Execute each of the test cases at least once, verifying that the actual results match the expected results for each case. Report any defects found. In order to differentiate between defects found during this stage and the previous stage, you can use the tags provided under the keyword field and/or in the summary field type “SFT” (Scripted Functional Testing) before the summary of the defect.

Test case design - One of the important activities during the testing process is to update the existing documents, such as test case scripts. Before starting the tests, you should design test cases for new functionalities added to the system. Assuming that the purchase functionality is a newly added feature in the system, there are no existing test cases in your scripts. You need to write down test cases for this functionality of the system and submit them in your assignment report.

Improving existing test cases – During the testing process, testers should keep an eye on the test cases that are obsolete, missing, or may need to be improved. Document test cases that need such improvements and include them in your assignment report.

Upon completion of testing, review all defects reported, and export the list of your defects to CSV format. You need to include this in your submission.

6.3 REGRESSION TESTING (VERIFICATION OF DEFECT FIXES)

Now, assume that system developers have partially fixed the system based on the defects you identified and have released a new version of the software: Version 1.1. Get this version from the URL provided in the introduction section and install it. You may stop the previous version and start this one instead.

¹ “Lesson 67: Report defects promptly” – from the book “Lessons Learned in Software Testing”

The defects reported in the two previous stages of testing can be divided among the group members to be retested. Retest each of these defects to determine which have been fixed and which have not. Since we do not know which defects have been fixed exactly, assume that all defects have had an attempt to fix them. Update all defect status to Resolved (Fixed) by changing its status appropriately. If the defect has actually been fixed in the Petstore application version 1.1, change the status again to **Verified**. If the defect has not actually been fixed in version 1.1, change the status to **Reopened** and write a comment.

Also, test the system again for possible new bugs. Use version 1.1 to separate new bugs that you may find in this system. If a defect is found which had previously been reported, do not report it again. When reporting these defects, ensure that version 1.1 is selected.

6.4 SUMMARY

Within your group, you should now each be familiar with the main features of a bug tracking system, and have a general understanding of how to use it to effectively report and track defects. You have also progressed through a short iteration of exploratory testing, manual functional testing, and regression testing.

7 REFERENCES

- Cem Kaner's slides on Bug Advocacy: <http://www.kaner.com/pdfs/bugadvoc.pdf>
- Sun Developer Network (SDN) Article - How to Write a Helpful Bug Report:
<http://www.oracle.com/technetwork/articles/java/bugreport-howto-135155.html>
- Bug Reporting Best Practices by Apple Developer Connection (for testers of iPod and iPhone!):
<http://developer.apple.com/bugreporter/bugbestpractices.html>

9 SUBMISSION

You are expected to submit three files:

- 1) Assignment report
- 2) CSV export of your bugs after version 1.0
- 3) CSV export of all your bugs after testing version 1.1

Make sure that all the important columns show up in your report. You can “Change Columns” and customize your report.

The grading criteria (rubric) for defect reports are as follows:

Item	
Correctness: Does the defect report contain detailed defect information? Does the report contain all the defects in the same level of detail? Does it contain the input, the expected output, and the faulty output for each defect?	20%
Clarity and adherence to defect reporting guidelines: Is it obvious where to start (what state to bring the program to) to replicate the defect? Is it obvious what you would type? Is it obvious what files to use (if any)?	20%
Number of defects found: Note that not all defects need to be found. But if it appears that not enough effort was made in finding defects, marks may be deducted.	10%
Assignment report - Questions Your answers will be evaluated based on their correctness and clarity.	20%
Assignment report - A high-level description of the exploratory testing plan Your answers will be evaluated based on their correctness and clarity.	10%
Assignment report - New test cases Your answers will be evaluated based on their correctness and clarity.	12%
Assignment report – improving existing test cases Your answers will be evaluated based on their correctness and clarity.	5%
Assignment report - Any difficulties encountered, challenges overcome, and lessons learned from performing the assignment Your answers will be evaluated based on their correctness and clarity.	3%
Assignment report - Comments/feedback on the assignment itself.	+2%

APPENDIX - FUNCTIONAL TEST SUITE FOR MANUAL SCRIPTED TESTING

Test Case #	Use Case	Function Being Tested	Initial System State	Input	Expected Output
1	Account	Sign in	The application is open. User is not logged in.	Valid username & password	User logs into the system. Username is shown in the page.
2	Account	Sign in	The application is open. User is not logged in.	Invalid username & password.	An error message is shown to the user. User cannot login to the system.
3	Account	Sign in	The application is open. User is not logged in.	Valid username & invalid password.	An error message is shown to the user. User cannot login to the system.
4	Account	Sign in	The application is open. User is not logged in.	Invalid username & an existing Password.	An error message is shown to the user. User cannot login to the system.
5	Account	Sign in	The application is open. User is not logged in.	Empty username & password.	An error message is shown to the user. User cannot login to the system.
6	Account	Sign out	User is signed in to the application.	Click sign out option. Then press the back button on the browser.	User should not have access to the services that require sign in, e.g., change account information.
7	Account	Sign out	Sign in to the application in two browser tabs	Click sign out option in one browser tab.	In the second tab, user should also not have access to the services that require sign in, e.g., change account information.
8	Account	Change user info – password	Sign in to the application and go to My account page	Enter a new non-empty password. Enter a matching password repeat. Save the information.	The password will change, and user can login with the new password.

9	Account	Change user info – password	Sign in to the application and go to My account page.	Enter a new non-empty password. Enter a non-matching password repeat. Save the information.	An error message is shown to the user. The password will not change.
10	Account	Change user info – password	Sign in to the application and go to My account page	Enter a new non-empty password. Leave the password repeat, empty.	An error message is shown to the user. The password will not change.
11	Account	Change user info – password	Sign in to the application and go to My account page	Enter a new password and password repeat with 1 character.	An error message is shown to the user. The password will not change.
12	Account	Change account – first name	Sign in to the application and go to My account page	Leave user information and profile information as it is. Enter a new valid first name. Leave the other fields of the account information as they are.	The name will change. User can sign out and sign in and see the new name.
13	Account	Change account – first name	Sign in to the application and go to My account page	Leave user information and profile information as it is. Enter the first name empty. Leave the other fields as they are.	First name is a mandatory field. Therefore, there should be an error message, and the account information should not change.
14	Account	Change account	Sign in to the application and go to My account page	Repeat step 12 for the following fields. Run it multiple time, with changing only the value of one field at a time to a valid value: last name, email, phone,	The value of the target field will change. User can sign out, sign in, and see the new information.

				address1, address2, city, state, zip, country.	
15	Account	Change account	Sign in to the application and go to My account page	Leave user information and profile information as it is. Run it multiple time, with only one field being empty at each step: last name, email, phone, address1, city, state, zip, country. Leave the other fields as they are.	There should be an error message, and the account information should not change.
16	Account	Change account – address2	Sign in to the application and go to My account page. Make sure that the address 2 field is non-empty, if not create as user with non-empty address 2.	Leave user information and profile information as it is. Enter empty value for address 2. Leave the other fields as they are.	Address 2 is not a mandatory field. The system should change the information successfully.
17	Account	Change profile info	Sign in to the application and go to My account page	Change your favorite category to a value other than its current value.	The image and the text in the footer of the page should be updated, accordingly.
18	Account	Register	Open the application, go to the login page, and click on the registration form.	Enter your student id as a userid. Enter non-matching passwords. Enter valid inputs in all other fields and click save.	You should receive an error message, and no user should be created.
19	Account	Register	Open the application, go to the login page, and click on the registration form.	Leave all fields empty, and click on “Save”.	You should receive an error message, and no user should be created.

20	Account	Register	Open the application, go to the login page, and click on the registration form.	Enter your student id as a userid. Enter an empty value for the first name. Enter valid inputs in all other fields and click save.	First name is mandatory, and therefore, you should receive an error message, and user should not be created.
21	Account	Register	Open the application, go to the login page, and click on the registration form.	Enter your student id as a userid. Enter an empty value for each of the following fields, one at a time. Enter valid inputs in all other fields and click save. Fields: last name, email, phone, address1, city, state, zip, country	You should receive an error message, and user should not be created.
22	Account	Register	Open the application, go to the login page, and click on the registration form.	Enter your lastname as a userid. Enter valid inputs in all other fields and click save.	New user will be created and you should be able to sign in with this new user.
23	Account	Register	Open the application, go to the login page, and click on the registration form.	Enter your student id as a userid. Enter an empty value for address 2. Enter valid inputs in all other fields and click save.	New user will be created and you should be able to sign in with this new user.
24	Account	Register	Open the application, go to the login page, and click on the registration form.	Enter an existing user id. Enter valid inputs in all other fields and click save.	An error should be shown since the user id should be unique.

25	Account	Register	Open the application, go to the login page, and click on the registration form.	Select your favorite category in the profile section. Enter all other values correctly.	New user will be created and you should be able to sign in with this new user. The image in the footer of the page, must match with the category you selected.
26	Shopping Cart	Browse	Open the application, go to the shopping cart and make sure it is empty. Then, return to the menu.	Select a product from the category and add it to your shopping cart. Using the same browser, return to the menu and go back to the shopping cart.	The product you added to the shopping cart should still be there.
27	Shopping Cart	Browse	Using the same browser, you used in test case 26, return to the catalog page.	Select a product from the category and add it to your shopping cart. Using the same browser, go back to the shopping cart page.	There should be two products in your shopping cart.
28	Shopping Cart	Browse	Using the same browser, you used in test case 27, go to the shopping cart page with two items in it.	Delete the first item from the shopping cart.	The first product is removed and one product item remains in the shopping cart.
29	Shopping Cart	Browse	Using the same browser, you used in test case 28, go to the shopping cart page	Increase the quantity of one product in your list, and enter the value in the quantity field. Without clicking on update, go to the shopping page again (reload the page).	The quantity of the product must stay the same.
30	Shopping Cart	Browse	Using the same browser, you used in test case 28,	Increase the quantity of one product in your list, and	The quantity of the product must have been increased.

			go to the shopping cart page	enter the value in the quantity field. After clicking on the update button, go to the shopping page again (reload the page).	
31	Shopping Cart	Browse	Open the application, go to the shopping cart and make sure that there is one item in it. If there is not, add one product to your shopping list.	Go to the menu and select the same product, and add it to your shopping cart.	The quantity of the product must increase.
32	Shopping Cart	Browse	Open the application, go to the shopping cart and make sure that there is one item in it. If there is not, add one product to your shopping list.	Change the quantity of a product to a negative value, and update the cart.	The system must show an error message and the quantity should not be updated.
33	Shopping Cart	Browse	Open the application, go to the shopping cart and make sure that there is one item in it. If there is not, add one product to your shopping list.	Change the quantity of a product to zero, and update the cart.	The system must show an error message and the quantity should not be updated.
34	Shopping Cart	Browse	Open the application, go to the shopping cart and make sure that there is at least one item in it. If there is not, add one product to your shopping list.	Remove all product items from the list.	The shopping list must get empty.

35	Shopping Cart	Browse	Close your browser tab, and open a new tab using the same browser that you used in test case 27, go to the shopping cart page	-	There should still be one product item in your shopping cart.
36	Shopping Cart	Browse	Switch to a different browser and open the application, then go to the shopping cart page.	-	The shopping cart must be empty.
37	Shopping Cart	Browse	Open the application, sign in to the system. Go to the shopping cart and make sure that there is one item in it. If there is not, add one product to your shopping list. Sign out from the application.	Sign in again and go to the shopping cart page.	The item should still be in the shopping cart.
38	Shopping Cart	Browse	Open the application, sign in to the system. Go to the shopping cart and make sure that there is one item in it. If there is not, add one product to your shopping list.	Sign in again and go to the shopping cart page.	The item should still be in the shopping cart.

			Sign out from the application. Open the application using a different browser.		
39	Catalog	Search	Open the application.	Enter the full name of an existing product item in the search box, e.g., Manx and click on the search button.	You should see one matching record in the search result.
40	Catalog	Search	Open the application.	Enter the full name of an existing product item (All in capital letter) in the search box, e.g., FINCH and click on the search button.	You should see one matching record in the result.
41	Catalog	Search	Open the application.	Enter the partial name that matches with some product names, e.g., Fi, and click on the search button.	All the product items that their name contain this text must appear in the search results.
42	Catalog	Search	Open the application.	Enter the text that does not match with any product name, e.g., UVWXYZ, and click on the search button.	The search result should be empty.