

Advance Software Engineering (SE-406)

LAB A1-G3

Laboratory Manual



Department of Software Engineering

DELHI TECHNOLOGICAL UNIVERSITY(DTU)

Shahbad Daulatpur, Bawana Road, Delhi-110042

Submitted to: -

Ms. Parul Sharma

Submitted by:-

Name: ASHISH KUMAR

Roll number: 2K18/SE/041

EXPERIMENT 2

- ASHISH KUMAR

- 2K18/SE/041

Aim:- To create test cases from the requirement specification document(any case study) using the following Black Box Testing techniques.

Introduction:- I have chosen this requirement specification document “E-Commerce System” written by Andrew Blossom, Derek Gebhard, Steven Emelander, Robert Meyer.

Here is the link: <https://cse.msu.edu/~chengb/RE-491/Papers/SRS-BECS-2007.pdf>

This Software Requirements Specification (SRS) illustrates, in clear terms, the system’s primary uses and required functionality as specified by our customer.

But first let’s talk about what are SRS in software engineering?

The Software Requirements Specification is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested. Its primary purpose is to provide a clear and descriptive “statement of user requirements” that can be used as a reference in further development of the software system.

Now in this experiment we need to write test cases using black box testing. **Black Box Testing** is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

Types of Black Box Testing

There are many types of Black Box Testing but the following are the prominent ones –

- **Functional testing** – This black box testing type is related to the functional requirements of a system; it is done by software testers.
- **Non-functional testing** – This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.

Note:- I am doing **Functional Testing**.

General Test Cases for E-commerce system:

1. Verify that the user is able to navigate through all the products across different categories.
2. Verify that all the links and banners are redirecting to correct product/category pages and none of the links are broken.
3. Verify that the company logo is clearly visible.
4. Verify that all the text – product, category name, price, and product description are clearly visible.
5. Verify that all the images – product and banner are clearly visible.
6. Verify that category pages have a relevant product listed specific to the category.
7. Verify that the correct count of total products is listed on the category pages.
8. Search – Verify that on searching the product satisfying the search criteria are visible on the search result page.
9. Search – Verify the more relevant product for the search term is displayed on the top for a particular search term.
10. Search – Verify that count of products is correctly displayed on the search result page for a particular search term.
11. Filtering – Verify that filtering functionality correctly filters products based on the filter applied.
12. Filtering – Verify that filtering works correctly on category pages.
13. Filtering – Verify that filtering works correctly on the search result page.
14. Filtering – Verify that the correct count of total products is displayed after a filter is applied.

Product Buy Flow – Test cases for E-commerce system:

1. Verify that on the product page, the user can select the desired attribute of the product e.g. size, color, etc.
2. Verify that the user can add to cart one or more products.
3. Verify that users can add products to the wishlist.
4. Verify that the user can buy products added to the cart after signing in to the application (or as per the functionality of the website).
5. Verify that the user can successfully buy more than one product that was added to his/her cart.
6. Verify that the user cannot add more than the available inventory of the product.
7. Verify that the limit to the number of products a user can buy is working correctly by displaying an error message and preventing the user from buying more than the limit.
8. Verify that the delivery can be declined for the places where shipping is not available.
9. Verify that the Cash on Delivery / online payment option is working fine.
10. Verify that product return/refund functionality works fine.

User Registration – Test cases:

1. Verify that all the specified fields are present on the registration page.
2. Verify that the required/mandatory fields are marked with * against the field.
3. Verify that for better user interface dropdowns, radio buttons and checkboxes; etc fields are displayed wherever possible instead of just textboxes.
4. Verify the page has both submit and cancel/reset buttons at the end.

5. Verify that clicking submits button after entering all the required fields, submits the data to the server.
6. Verify that clicking cancels/reset button after entering all the required fields, cancels the submit request, and reset all the fields.
7. Verify that not filling the mandatory fields and clicking the submit button will lead to validation error.
8. Verify that not filling the optional fields and clicking the submit button will still send data to the server without any validation error.
9. Check the upper limit of the textboxes.
10. Check validation on the date and email fields (only valid dates and valid email Ids should be allowed).
11. Check validation on numeric fields by entering alphabets and special characters.
12. Verify that entering blank spaces on mandatory fields leads to validation error.

SHOPPING CART TEST CASES:

1. Website users can easily add/remove products to/from a shopping cart.
2. Users can add the same product multiple times and change their quantity in the cart directly.
3. It is possible to add the same products in different variations – color, size, etc.
4. Users can add products from different categories to the cart.
5. Items in the cart are displayed with correct names, images, and prices.
6. The items are clickable, and the links lead to corresponding product pages.
7. An order price updates when a user adds/removes a new item to/from the cart.
8. Shipping charges are included in the total price but displayed separately along with the other parameters.
9. If an item is out of stock, a user cannot add it to the cart.
10. When a user removes all items from the cart, nothing is displayed and the total price equals zero.
11. When a user closes a tab with a shopping cart, the items should remain in the cart.

CHECKOUT FLOW TEST CASES:

1. There is a prompt to log in or register before completing the purchase.
2. Customers can check out and pay as guest users without registration.
3. Returning logged-in users can use pre-saved shipping and billing information.
4. All types of supported payment methods work correctly.
5. Sensitive information, including payment details, isn't stored after the payment.
6. An order confirmation page appears after a successful payment.
7. A user receives an order confirmation message via an email or text message. Order status is available in the account for registered users.
8. After completing the payment, a user is able to continue navigating through the site.

UI TEST CASES:

1. A user can enter their credentials in the signing and logging forms.
2. Users can type texts in the text fields, such as comments, reviews, etc.
3. All buttons are clickable and perform the programmed actions.
4. Breadcrumbs, tags, and other navigational elements are clickable and linked to the correct pages.
5. Toggles change their position after a click/slide, and each position is visually distinctive.
6. The performance and design of UI elements don't vary depending on a user's device.

UX TEST CASES:

1. Sorting functionality considers all the key parameters.
2. If search results go beyond one page, it is easy to navigate between the pages.
3. The font color and size make information easy to read.
4. Product descriptions are relevant and free of mistakes.
5. All category pages feature relevant products.
6. A user has a good overview of the page, without pop-up elements covering much information.
7. The design is consistent across different devices and screens, as well as the branding in general.

Learning from experiment:- We have successfully learnt about black box testing and its methods. We are successfully able to test e-commerce system by creating test cases of respective use case scenarios.