**ASQA Engineer Training Tasks**

**Name: Nasir Hussain**

**Employee ID: 878**

**Buddy: Zohaib Ali**

**17-May-2022**

**Task 1:**

**Prepare a document stating why software quality is important, and what impact it has over the development teams?**

Software Quality refers to a software system which is bug and defect free, delivered in time and within budget, meets the requirements, is maintainable and of high quality. Quality of software is an important concern for both clients and organization, as clients always demand high quality and error free software and organizations also try to deliver a product which is error free and perform its functionality according to user’s requirements. Delivering quality software gains customer’s trust and increases the reputation of an organization. It ensures that the end product is competitive, secure, and smoothly performs its expected functions. Software quality drives predictability. Do it once and do it right, and there will be less re-work, less variation in productivity and better performance overall. Products get delivered on time, and they get built more productively. High quality software applications which are free from errors and bugs can enhance the credibility and reputation of the business. Thus, the owners of the business will be capable of boosting the profits as well as the growth of the business. A quality product satisfies the customer. A satisfied customer comes back for more and provides positive referrals. Quality of software reflects the skills of the development team; a good and experienced team always develops quality software by utilizing all possible and suitable design patterns and optimized code. Quality software are easy to maintain and reusability of these software is high.

**Task 2:**

**Prepare a document listing the parallel comparison of the steps in SDLC and STLC, with examples.**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **SDLC** | **STLC** |
| Abbreviation | Software Development Life Cycle | Software Testing Life Cycle |
| Origin | Development Life Cycle | Testing Life Cycle |
| Definition | SDLC aims to produce a high-quality system that meets or exceeds customer expectations, work effectively and efficiently. | STLC identifies what test activities to carry out and when to accomplish those test activities. |
| Objective | The objective of SDLC is to complete successful development of the software. | The only objective of the STLC phase is testing |
| Phases | 1. Requirement Gathering 2. Design Phase 3. Development Phase 4. Testing Phase 5. Deployment & Maintenance | 1. Requirement Analysis 2. Test Planning 3. Test Design 4. Environment Setup 5. Test Execution and Closure |
| Requirement Gathering | Business analyst gathers the requirements and create Development Plan.  **Deliverable: Software Requirement Specification** | In STLC, the QA team analyze requirement documents and create System Test Plan  **Deliverable: RTM Requirement Traceability Matrix** |
| Design Phase | Development team creates the high and low-level design plans  **Deliverable: Design Document** | In STLC, the test analyst creates the Integration Test Plan  **Deliverable: Test Plan** |
| Development Phase | The real code is developed, and actual work takes place as per the design documents.  **Deliverable: Code** | The testing team prepares the test environment and test cases  **Deliverable: Test Case, Test Data** |
| Testing Phase | Testing is done in this phase, which include Unit, Integration, System, Regression etc.  **Deliverable: Testing Document** | Test cases are executed in this phase and defect report and retesting is also done in this phase.  **Deliverable: Defect Report** |
| Deployment & Maintenance | SDLC phase also includes post-deployment supports and updates.  **Deliverable: BRD Versions** | Testers performs regression testing to updated and evolved system.  **Deliverable: Regression Test Suite** |
| Outcome | A good quality software product | A bug free software |

**18-May-2022**

**Task 3:**

**Prepare a document and enlist 3 major differences between BRD, SRS, FSD, with examples.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **BRD** | **SRS** | **FSD** |
| Abbreviation | Business Requirement Document | Software Requirements Specifications | Functional Specification Document |
| Definition | BRD is a formal document which describe requirement provided by clients. | SRS is a document which describe functional and nonfunctional requirement of the software to be developed. | FSD is a document which describe all the details around the functional requirements on a project. |
| Contains | BRD describes very high-level business and stakeholder requirements. | It describes at a high level, the functional and technical specification of the software. | It describes granular functional requirements, data flow and UML diagrams. |
| Created By | Business Analyst | System/Business Analyst | System/Business Analyst, Implementation Team |
| Origin | It is derived from client interaction and requirements. | It is derived from the BRS. | It is derived from SRS. |
| Example | Improve user experience by providing tracking facility of parcel. | Tracking software will contain Login, Admin, Customer and Organization modules. | Login module will contain fields like Enter Username, Enter Password, Sign in Button. |

**20-May-2022**

**Task 4:**

**Statement Coverage and Path Finding is performed in QA or QC? Explain with justifications according to your understanding.**

* **Quality Assurance** is a process by which entities review the quality of all factors involved in production, and to manage the quality of product.
* **Quality Control** is a part of quality management focused on fulfilling quality requirements. It is used to test the deliverable project in order to catch bugs and to make it bugs free. It is usually done by testers.
* **Statement Coverage** is white box testing technique which demands execution of all statements at least once in the source code. It is a metric, to calculate and measure number of executed statements. It is done by tester to check code coverage.
* **Path Finding** is used to find all possible paths from source point to end point. In Software Development path finding is used to check all possible executable path using source code. We draw Control Flow Graph from code and using CFG we make paths.

**Statement Coverage is performed in Quality Control.**

* Statement Coverage is done by Tester.
* It is performed by executing code.
* It is used to identify bugs and errors in code.

Based on these statements we consider Statement coverage as a Quality Control. As in statement coverage our main focus is to execute each line of code to check the existence of errors in source code.

**Path Finding is performed in Quality Assurance.**

* We find path by just studying code and draw control flow graph.
* It doesn't need to execute the program.
* It is basically used to find all possible path and from these paths we select best paths for testing.

We consider Path finding as Quality Assurance as it doesn't need to execute the code, and it also provide paths which are used to select best path in this way we save our resources and time of testing. QA is preventive technique and Path finding is also preventive as we prevent extra effort of testing by selecting best paths to gain maximum coverage.

**24-May-2022**

**Task 5:**

**Identify and report maximum bugs present on** [**https://www.demoblaze.com/**](https://www.demoblaze.com/)

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-001** | **URL** | [*https://www.demoblaze.com/*](https://www.demoblaze.com/) |
| **Summary** | | Previous Button on Home Screen | |
| **Description** | | When website is opened, on bottom of page after products list there is Previous and Next Button, as this is Home Page, there should be Next Button, but no need of that Previous Button, as web don’t have any previous page to load. | |
| **Bug Type** | | UI Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)*)* 2. Scroll down to bottom of page 3. After product list, there is Previous and Next Button | |
| **Actual Results** | | Previous and Next Button on Home Screen | |
| **Expected Results** | | Just Next Button on Home Screen | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-002** | **URL** | [*https://www.demoblaze.com/*](https://www.demoblaze.com/) |
| **Summary** | | Products missing from Product List | |
| **Description** | | There are total of 15 products on web store, initially on first page there are 9 products and on 2nd page there are 6 products, but when you click next button and then click previous button, one product from 1st page is no longer shown on web. And again, if you click Next button, now on 2nd page there are 5 products instead of 6 products. | |
| **Bug Type** | | UI Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)*)* 2. Scroll down to view list of products (Samsung Galaxy S6 is on top) 3. Click on Next Button (there will be 6 products on Next Page) 4. Now Click on Previous Button (Samsung Galaxy S6 is no longer in list) 5. Now click again on Next Button (There are 5 products instead of 6.) | |
| **Actual Results** | | One product (Samsung Galaxy S6) is no longer available in list | |
| **Expected Results** | | There should be 15 products, 9 on 1st page and 6 on 2nd page. | |
| **Screenshot** | | [*https://www.loom.com/share/dcca0d24fe2146d199ff5deb68ac92aa*](https://www.loom.com/share/dcca0d24fe2146d199ff5deb68ac92aa) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-003** | **URL** | [*https://www.demoblaze.com/*](https://www.demoblaze.com/) |
| **Summary** | | Previous Button not working on last page. | |
| **Description** | | When product store is opened, at the end of product list there is Next and Previous Button, when Previous Button is clicked it load some new product in list, but when Previous Button is again clicked, it doesn’t response anything, so there may be functional issues as previous button didn’t perform its role. And if we don’t have any previous page here then its UI Bug, then this page doesn’t show Previous button as there is no need of previous button on this page. | |
| **Bug Type** | | Functional and UI Both | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)) 2. Scroll down to bottom of page 3. After product list, there is Previous Button, click it 4. Again, click Previous Button, it doesn’t response this time. | |
| **Actual Results** | | Previous button didn’t response. | |
| **Expected Results** | | Previous button should load previous page and show some new product in list. | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-004** | **URL** | [*https://www.demoblaze.com/*](https://www.demoblaze.com/) |
| **Summary** | | Empty message without message body, recipient email and contact name send. | |
| **Description** | | System allows to send message with empty fields, without providing contact email, contact name and message, when user click Send Message, system display message “Thanks for the message”. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)) 2. Click on Contact from Menu Bar 3. Without providing any detail, just click Send Message. | |
| **Actual Results** | | Message send and System display “Thanks for message” | |
| **Expected Results** | | System should display “Enter fill out fields to send message”. | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-005** | **URL** | [*https://www.demoblaze.com/*](https://www.demoblaze.com/) |
| **Summary** | | No validation used on Email and Username fields of Contact form. | |
| **Description** | | Validation for email and name is not used in Contact Email and Contact Name fields respectively, and system allow to send message with fake data. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)*)* 2. Click on Contact from Menu Bar 3. Enter “bgf” in Contact Email 4. Enter “457jh” in Contact Name 5. Enter “Associate Software Quality Assurance Engineer at Devsinc” in message body. 6. Click on Send Message. | |
| **Actual Results** | | Message send and System display “Thanks for message” | |
| **Expected Results** | | Please provide Valid Email and User Name to send message. | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-006** | **URL** | [*https://www.demoblaze.com/cart.html*](https://www.demoblaze.com/cart.html) |
| **Summary** | | With Empty Cart, order is placed. | |
| **Description** | | Without selection of any product and with empty cart, system allow to place order. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)*)* 2. Click on Cart from Menu Bar 3. Click Place Order 4. Enter random details in field of Place Order form. 5. Enter Purchase | |
| **Actual Results** | | Order placed and System display “Thanks you for your purchase” | |
| **Expected Results** | | System should display “Enter product in cart to place order”. | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-007** | **URL** | [*https://www.demoblaze.com/cart.html*](https://www.demoblaze.com/cart.html) |
| **Summary** | | Same products should be merged as one with number of quantities. | |
| **Description** | | Order multiple quantity of same product merge in group, and there must be number of quantities display on screen in cart. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)*)* 2. Click any of product from product list 3. Click on Add to Cart for 2-3 times. 4. Click on Home from Menu Bar and again select any product. 5. Click on Add to Cart. 6. Now click on Cart from Menu Bar. | |
| **Actual Results** | | Same products are displayed separately in cart. | |
| **Expected Results** | | Same products merge in one and should show number of quantities. | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-008** | **URL** | [*https://www.demoblaze.com/cart.html*](https://www.demoblaze.com/cart.html) |
| **Summary** | | Date on Order Bill is wrong. | |
| **Description** | | Date on order bill is not correct, I placed order on 25th of May, but on bill system display 25th of April. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)*)* 2. Select product and add it to cart. 3. Click on Cart from Menu Bar 4. Click Place Order 5. Enter “Nasir” in Name field 6. Enter “0000-000000-000” in Credit Card field. 7. Enter Purchase | |
| **Actual Results** | | Date on Bill is 25/4/2022 | |
| **Expected Results** | | Date on Bill should be the actual date of order placement. | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-009** | **URL** | [*https://www.demoblaze.com/cart.html*](https://www.demoblaze.com/cart.html) |
| **Summary** | | No Validation on Order Place form. | |
| **Description** | | There are Name, Country, City, Credit Card, Month and Year fields on Order Place form, and no validation is used in all of these fields. Moreover, by providing just fake Name and Credit Card, order will be placed. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)) 2. Select product and add it to cart. 3. Click on Cart from Menu Bar 4. Click Place Order 5. Enter random data in Name and Credit Card field. 6. Leave all other fields as empty. 7. Enter Purchase | |
| **Actual Results** | | Order placed and System display “Thanks you for your purchase” | |
| **Expected Results** | | System should display error message “Provide valid details and fill complete form” | |
| **Screenshot** | | [*https://www.loom.com/share/c4e49f67fa884338b680fed0f5191223*](https://www.loom.com/share/c4e49f67fa884338b680fed0f5191223) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-010** | **URL** | [*https://www.demoblaze.com/cart.html*](https://www.demoblaze.com/cart.html) |
| **Summary** | | All products in cart become hide, when we login to system. | |
| **Description** | | If we add products in cart before login, and then after adding products in cart when we login to the system, all our products in cart become hide, and then if we sign out of our system then products in cart become visible. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Add products in cart 2. Now open cart to verify products in cart 3. Now click on login from main menu and provide user name and password 4. Click on login button. (Products in cart are not visible now) 5. Click on Logout from Main Menu. 6. Now click on Cart (All those products are now visible in cart) | |
| **Actual Results** | | Products in cart are visible after login. | |
| **Expected Results** | | Product list should be visible after login in system. | |
| **Screenshot** | | [*https://www.loom.com/share/65f5cc46c79a4bfc9958b5ba094b4209*](https://www.loom.com/share/65f5cc46c79a4bfc9958b5ba094b4209) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-011** | **URL** | [*https://www.demoblaze.com/index.html*](https://www.demoblaze.com/index.html) |
| **Summary** | | No validation on Signup page | |
| **Description** | | No validation for username and password used on Signup page. | |
| **Bug Type** | | Functional Bug | |
| **Steps to Reproduce** | | 1. Click on Sign up from menu. 2. Provide “@$#%” in user name 3. Provide “000” value in password 4. Click on Signup button. | |
| **Actual Results** | | Signup successful. | |
| **Expected Results** | | Follow valid username and password format. | |
| **Screenshot** | |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Bug Report (Product Store)** | | | |
| **Bug ID** | **PS-012** | **URL** | [*https://www.demoblaze.com/*](https://www.demoblaze.com/) |
| **Summary** | | Layout Issues | |
| **Description** | | On different Menus, layout have different layout, when click on Home menu will be at top right side, when click on Cart menu move to mid-point, | |
| **Bug Type** | | UI Bug | |
| **Steps to Reproduce** | | 1. Open the URL ([*https://www.demoblaze.com/*](https://www.demoblaze.com/)) 2. Click on Home (Note the position of Menu) 3. Click on Cart (Note Position of Menu) | |
| **Actual Results** | | Menu bar moves towards mid-point. | |
| **Expected Results** | | Menu bar have proper alignment and position. | |
| **Screenshot** | | **Home Page Layout:**    **Cart Page Layout:** | |

**25-May-2022**

**Task 7:**

**Prepare a short summary describing the advantages and disadvantages of TDD and BDD.**

Test Driven Development and Behavior Driven Development are used as test strategies for software applications, both require writing and creating tests before code development. TDD is a development practice in which developers first write test cases in programming language, then write code for that. While in BDD we used behavior of application instead of test cases. Both TDD and BDD have advantages in software development, but along with advantages there are also disadvantages of both.

**Test Driven Development**

* **Advantages of TDD**

TDD helps reduce the time required for rework and increase reusability, its helps to write only specific code that we need in this way it saves our time and effort. It also helps to explore and identify bugs and errors more quickly in initial phases of system development, so we get feedback very quickly and fast. As test cases are already designed and code are written according to these test cases so it encourages cleaner and better designs development. TDD promote collaboration and knowledge sharing by allowing any member of team to start working on the code in the absence of a specific team member. As bugs are identified and corrected during development so it results in the creation of extensive code that is flexible and easy to maintain. TDD aims to provide high test coverage. In TDD, architecture of the code is also good, which make our code testable. TDD provide celerity in writing a code, which is helpful for changing in code. TDD writes the code in small steps which makes the design of code better to understand. TDD result in effective and efficient system as testing is done side by side throughout the program.

* **Disadvantages of TDD**

In TDD test cases are difficult to maintain as they are dependent on external dependencies. In TDD we write test cases from user requirements and as requirements change, we have to change our test cases, which result in changing of code. We have to write test cases first then to write code, which make our development process slow. Developed code in TDD is tough to understand as writing a code and writing a code well is different. In TDD, focus is only on code coverage as functionality of TDD is not discussed. As learning point, TDD require more effort and time as it is not easy for anyone to learn. If we want to refactor in an early stage then we have to refactor the test classes as well.

**Behavior Driven Development**

* **Advantages of BDD**

BDD used non-technical language to write behavior scenarios so it helps to reach a large number audience, as natural language is easy to understand by all participants. Its main focus is how the system behave from the developer's and customer’s perspective. It is also a cost-effective technique and it reduces efforts needed to verify any post-deployment defects. BDD identify scenarios in the start of development which help in less rework at the end of the project, as a result it uncovers additional capabilities and complexities from the project. BDD improve the communication and avoids misunderstandings as domain specific language is used to write behavior scenarios. As the details behavior specifications are written in the starting and we write tests before the code. So, this is the plus point for the BDD as the developers only need to focus on implementing which makes the test cases pass.

* **Disadvantages of BDD**

BDD require a lot of time and effort to create the Scenarios and maintaining the files, so it is not good for a short project which we need to complete in a short interval of time. BDD requires good communication between the person who is developing the automation code and the person who is writing the features files, as person who is writing the automation needs these files and scenarios for developing the automation script. If they Don’t have a mutual understanding of the files then it’s hard to develop the project. In BDD, test cases are dependent on the external data which often causing the problem when execute the test cases. BDD require a complete and managed documentation because there are many files and scenarios which need to be understood. In BDD, correspondences between the client and the developer are basic, so if the client is not available, it will be hard to work past ambiguities and questions created by the client stories.

**27-May-2022**

**Task 6a:**

**Prepare Use cases, Test scenarios and Test Cases for the work in Task 5.**

**Use Case Description:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | US-001 | | **Use Case Name** | Signup |
| **Use Case Description** | | This use case allows users to register themselves with system. | | |
| **Actor** | | Customers | | |
| **Pre-Condition** | | * User must be connected to internet * User have a valid URL | | |
| **Basic Workflow** | | 1. User click on “Signup” from menu. 2. System should display “Sign up” form. 3. User enter username in User Name field. 4. User enter password in Password field. 5. User click on Signup Button. 6. System validates user name. 7. System registered user and display “Signup successful”. | | |
| **Alternative Workflow** | | 5a. User click on Signup Button without providing username and password.  System display error message “Please fill out Username and Password”  5b. User click on Signup Button by providing only username.  System display error message “Please fill out Username and Password”  5c. User click on Signup Button by providing only password.  System display error message “Please fill out Username and Password”  5d. User clicks on Close Button.  System close Signup form.  6a. User name already exist.  System display error message “This user already exists”. | | |
| **Post-Condition** | | User is registered and system display “Signup successful” | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | US-002 | | **Use Case Name** | Login |
| **Use Case Description** | | This use case allows customers to login to system. | | |
| **Actor** | | Customer | | |
| **Pre-Condition** | | * User must be registered user of system. | | |
| **Basic Workflow** | | 1. User click on “Login” from menu. 2. System should display “Login” form. 3. User enter username in User Name field. 4. User enter password in Password field. 5. User click on Login Button. 6. System validates user name and password. 7. System redirect user to homepage. | | |
| **Alternative Workflow** | | 5a. User click on Login Button without providing username and password.  System display error message “Please fill out Username and Password”  5b. User click on Login Button by providing only username.  System display error message “Please fill out Username and Password”  5c. User click on Login Button by providing only password.  System display error message “Please fill out Username and Password”  5d. User clicks on Close Button.  System close Login form.  6a. User enter wrong username in User Name field.  System display error message “User does not exist”  6b. User enter wrong password in Password field.  System display error message “Wrong password”  6c. User enter wrong username and password.  System display error message “User does not exist” | | |
| **Post-Condition** | | The customer is now logged into the system. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | US-003 | | **Use Case Name** | Send Message |
| **Use Case Description** | | This use case allow user to Send Message to communicate with admin and to provide feedback. | | |
| **Actor** | | Customer | | |
| **Pre-Condition** | | * Customer should be registered user. * Customer should login to system. | | |
| **Basic Workflow** | | 1. User click on “Contact” on Main Menu. 2. System should show New Message form. 3. User enter a valid email in Contact Email field. 4. User enter name in field of Contact Name. 5. User write message/feedback in body of Message. 6. User click on Send Message Button. 7. System validate email. 8. System display popup dialogue. | | |
| **Alternative Workflow** | | 6a. User click on Send Message without providing data in fields.  System display error message “Fill out all fields of forms.”  6b. User clicks on Close Button.  System clears all fields and close New Message form.  7a. User enter invalid email in Contact Email field.  System should show error message “Provide valid email” | | |
| **Post-Condition** | | Message/Feedback sent and System show “Thanks for the message!” | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | US-004 | | **Use Case Name** | View Product |
| **Use Case Description** | | This use case allow user to view and explore products from product list. | | |
| **Actor** | | Customer | | |
| **Pre-Condition** | | * User must be connected to internet * User have a valid URL | | |
| **Basic Workflow** | | 1. User click home from Menu. 2. System redirect user to home page 3. User scroll down to products. 4. User clicks on desired product. 5. System redirect user to new page of that specific product description. | | |
| **Alternative Workflow** | |  | | |
| **Post-Condition** | | User can add product in cart. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | US-005 | | **Use Case Name** | Add to Cart |
| **Use Case Description** | | This use case allow user to add products in cart. | | |
| **Actor** | | Customer | | |
| **Pre-Condition** | | * User must be connected to internet * User have a valid URL | | |
| **Basic Workflow** | | 1. User click home from Menu. 2. System redirect user to home page 3. User scroll down to products. 4. User clicks on desired product. 5. System redirect user to new page of that specific product description. 6. User click Add to cart. 7. System display message “Product added”. | | |
| **Alternative Workflow** | |  | | |
| **Post-Condition** | | Product added in cart and user can place order. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | US-006 | | **Use Case Name** | Delete from Cart |
| **Use Case Description** | | This use case allow user to view product list and delete products from cart. | | |
| **Actor** | | Customer | | |
| **Pre-Condition** | | * User must be connected to internet * User have a valid URL | | |
| **Basic Workflow** | | 1. User click cart from menu. 2. System display product list in cart. 3. User click delete against any product. 4. System removes that product from list. 5. User can view complete list of product and total bill. | | |
| **Alternative Workflow** | |  | | |
| **Post-Condition** | | Product added in cart and user can place order. | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | US-007 | | **Use Case Name** | Place order |
| **Use Case Description** | | This use case allow user to place order. | | |
| **Actor** | | Customer | | |
| **Pre-Condition** | | * User must be login in system. * User must have products in cart. | | |
| **Basic Workflow** | | 1. User click cart from menu. 2. System display product list in cart. 3. User click Place Order 4. System display Place Order form. 5. User enter name 6. User enter country 7. User enter city 8. User enter credit card detail 9. User enter month 10. User enter year 11. User click Purchase Button 12. System validates details 13. System display popup dialogue. | | |
| **Alternative Workflow** | | 3a. User click Place Order but there is no product in Cart.  System display message “Please enter products in cart.”  11a. User click on Purchase Button without providing any details.  System shows error message “Please fill out Name and Credit card”  11b. User click Close Button  System redirect user to cart page.  12a. User enter invalid Name  System shows error message “Please enter valid Name.”  12b. User enter invalid Credit card detail.  System shows error message “Please enter valid Credit card”. | | |
| **Post-Condition** | | System generates receipt and display “Thank you for your purchase!” | | |

**Test Case Description:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | TS-001 | | | | | | | | |
| **Test Scenario** | Check Signup Functionality | | | | | | | | |
|  | | | | | | | | | |
| **Test Case ID** | TC-001 | | **Test Case Name** | | | Valid username and password | | | |
| **Test Steps** | 1. Click on “Sign up” from Main Menu. 2. Enter username 3. Enter password 4. Click on signup button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | | **Actual Outcome** | **Result** |
| User Name: nasirhussain625  Password: nasir625 | | | | Signup successful | | | | Signup successful | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-002 | **Test Case Name** | | | Valid username without password | | | | |
| **Test Steps** | 1. Click on “Sign up” from Main Menu. 2. Enter username 3. Click on signup button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | | **Actual Outcome** | **Result** |
| User Name: nasirhussain628  Password: | | | | Error Message “Fill out Username and Password” | | | | Fill out Username and Password | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-003 | **Test Case Name** | | | Only password without username | | | | |
| **Test Steps** | 1. Click on “Sign up” from Main Menu. 2. Enter password 3. Click on signup button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | **Actual Outcome** | | **Result** |
| User Name:  Password: nasir625 | | | | Error Message “Fill out Username and Password” | | | Fill out Username and Password | | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-004 | **Test Case Name** | | | Empty Fields | | | | |
| **Test Steps** | 1. Click on “Sign up” from Main Menu. 2. Click on signup button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | | **Actual Outcome** | **Result** |
| User Name:  Password: | | | | Error Message “Fill out Username and Password” | | | | Fill out Username and Password | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-005 | **Test Case Name** | | | Already used username and password | | | | |
| **Test Steps** | 1. Click on “Sign up” from Main Menu. 2. Enter username. 3. Enter password. 4. Click on signup button. | | | | | | | | |
| **Test Data** | | **Expected Outcome** | | | | | **Actual Outcome** | | **Result** |
| User Name: nasirhussain625  Password: nasir625 | | Error Message “This user already exists”. | | | | | This user already exists | | Test Pass |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | TS-002 | | | | | | | | |
| **Test Scenario** | Check Login Functionality | | | | | | | | |
|  | | | | | | | | | |
| **Test Case ID** | TC-006 | | **Test Case Name** | | | Valid username and valid password | | | |
| **Test Steps** | 1. Click on “Login” from Main Menu. 2. Enter valid username 3. Enter valid password 4. Click on Login button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | | **Actual Outcome** | **Result** |
| User Name: nasirhussain625  Password: nasir625 | | | | Login successful | | | | Login successful | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-007 | **Test Case Name** | | | Valid username without password | | | | |
| **Test Steps** | 1. Click on “Login” from Main Menu. 2. Enter valid username 3. Click on Login button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | | **Actual Outcome** | **Result** |
| User Name: nasirhussain625  Password: | | | | Error Message “Fill out Username and Password” | | | | Fill out Username and Password | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-008 | **Test Case Name** | | | Only password without username | | | | |
| **Test Steps** | 1. Click on “Login” from Main Menu. 2. Enter password 3. Click on Login button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | **Actual Outcome** | | **Result** |
| User Name:  Password: nasir625 | | | | Error Message “Fill out Username and Password” | | | Fill out Username and Password | | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-009 | **Test Case Name** | | | Empty Fields | | | | |
| **Test Steps** | 1. Click on “Login” from Main Menu. 2. Click on Login button. | | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | | **Actual Outcome** | **Result** |
| User Name:  Password: | | | | Error Message “Fill out Username and Password” | | | | Fill out Username and Password | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-010 | **Test Case Name** | | | Invalid username and valid password | | | | |
| **Test Steps** | 1. Click on “Login” from Main Menu. 2. Enter invalid username. 3. Enter valid password. 4. Click on Login button. | | | | | | | | |
| **Test Data** | | **Expected Outcome** | | | | | **Actual Outcome** | | **Result** |
| User Name: nasirhussain62  Password: nasir625 | | Error Message “User does not exist”. | | | | | User does not exist | | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-011 | **Test Case Name** | | | Valid username and Invalid password | | | | |
| **Test Steps** | 1. Click on “Login” from Main Menu. 2. Enter valid username. 3. Enter invalid password. 4. Click on Login button. | | | | | | | | |
| **Test Data** | | **Expected Outcome** | | | | | **Actual Outcome** | | **Result** |
| User Name: nasirhussain625  Password: nasir6 | | Error Message “Wrong password”. | | | | | Wrong password | | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-012 | **Test Case Name** | | | Invalid username and invalid password | | | | |
| **Test Steps** | 1. Click on “Login” from Main Menu. 2. Enter invalid username. 3. Enter invalid password. 4. Click on Login button. | | | | | | | | |
| **Test Data** | | **Expected Outcome** | | | | | **Actual Outcome** | | **Result** |
| User Name: nasirhussain62  Password: nasir62 | | Error Message “User does not exist”. | | | | | User does not exist | | Test Pass |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | TS-003 | | | | | | | |
| **Test Scenario** | Check Send Message Functionality | | | | | | | |
|  | | | | | | | | |
| **Test Case ID** | TC-013 | | **Test Case Name** | | | Valid Email with all filled fields | | |
| **Test Steps** | 1. Click on “Contact” on Main Menu. 2. Enter Valid Email 3. Enter Name 4. Enter Message 5. Click on Send Message | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| Contact Email: nasirhussainpro@gmail.com  Contact Name: Nasir Hussain  Message: Associate Software Quality Assurance Engineer at Devsinc | | | | Message Send Successfully | | | Message Send Successfully | Test Pass |
|  | | | | | | | | |
| **Test Case ID** | TC-014 | **Test Case Name** | | | Invalid Email with all filled fields | | | |
| **Test Steps** | 1. Click on “Contact” on Main Menu. 2. Enter Invalid Email 3. Enter Name 4. Enter Message 5. Click on Send Message | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| Contact Email: nasirhussain  Contact Name: Nasir Hussain  Message: Associate Software Quality Assurance Engineer at Devsinc | | | | Error Message “Provide Valid Email” | | | Message Send Successfully | Test Fail |
|  | | | | | | | | |
| **Test Case ID** | TC-015 | **Test Case Name** | | | Empty Fields | | | |
| **Test Steps** | 1. Click on “Contact” on Main Menu. 2. Click on Send Message | | | | | | | |
| **Test Data** | | | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| Contact Email:  Contact Name:  Message: | | | | Error Message “Fill New Message Form to send Message” | | | Message Send Successfully | Test Fail |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | TS-004 | | | | | |
| **Test Scenario** | Check View Product Functionality | | | | | |
|  | | | | | | |
| **Test Case ID** | TC-016 | | **Test Case Name** | Product Selection | | |
| **Test Steps** | 1. Click on “Home” on Main Menu. 2. Scroll down to products 3. Click on desired Product | | | | | |
| **Test Data** | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| * Click on desired product | | Product Description Opened | | | Product Description Opened | Test Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | TS-005 | | | | | |
| **Test Scenario** | Check Add to Cart Functionality | | | | | |
|  | | | | | | |
| **Test Case ID** | TC-017 | | **Test Case Name** | Add to Cart | | |
| **Test Steps** | 1. Click on “Home” on Main Menu. 2. Scroll down to products 3. Click on desired Product 4. Click on add to cart | | | | | |
| **Test Data** | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| * Click on desired product * Click on add to cart | | Product added | | | Product added | Test Pass |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | TS-006 | | | | | |
| **Test Scenario** | Check Delete Cart Functionality | | | | | |
|  | | | | | | |
| **Test Case ID** | TC-018 | | **Test Case Name** | Delete Product | | |
| **Test Steps** | 1. Click on “Cart” on Main Menu. 2. Click on Delete Button | | | | | |
| **Test Data** | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| * Click on delete button | | Product removed from list | | | Product removed from list | Test Pass |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Scenario ID** | TS-007 | | | | | | | | |
| **Test Scenario** | Check Place Order Functionality | | | | | | | | |
|  | | | | | | | | | |
| **Test Case ID** | TC-019 | | | **Test Case Name** | | | Place order with all filled field | | |
| **Test Steps** | 1. Click on “Cart” on Main Menu. 2. Click on Place Order 3. Enter Name 4. Enter Country and City 5. Enter Credit Card Detail 6. Enter Month and Year 7. Click on Purchase Button | | | | | | | | |
| **Test Data** | | | | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| Name: Nasir  Country: Pakistan  City: Islamabad  Credit Card Detail: 0000-0000-0000-0000  Month: May  Year: 2022 | | | | | Order Placed Successfully | | | Order Placed Successfully | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-020 | | **Test Case Name** | | | Place order with empty cart | | | |
| **Test Steps** | 1. Click on “Cart” on Main Menu. 2. Click on Place Order | | | | | | | | |
| **Test Data** | | **Expected Outcome** | | | | | | **Actual Outcome** | **Result** |
| * Click on Place Order | | Error Message “Please enter products in cart” | | | | | | Order form Opened | Test Fail |
|  | | | | | | | | | |
| **Test Case ID** | TC-021 | | **Test Case Name** | | | Place order with all Empty Fields | | | |
| **Test Steps** | 1. Click on “Cart” on Main Menu. 2. Click on Place Order 3. Click on Purchase Button | | | | | | | | |
| **Test Data** | | **Expected Outcome** | | | | | | **Actual Outcome** | **Result** |
| Name:  Country:  City:  Credit Card Detail:  Month:  Year: | | Error Message “Please fill out Name and Credit card” | | | | | | Please fill out Name and Credit card | Test Pass |
|  | | | | | | | | | |
| **Test Case ID** | TC-022 | | | **Test Case Name** | | | Place order with invalid name | | |
| **Test Steps** | 1. Click on “Cart” on Main Menu. 2. Click on Place Order 3. Enter invalid Name 4. Enter Country 5. Enter City 6. Enter Credit Card Detail 7. Enter Month 8. Enter Year 9. Click on Purchase Button | | | | | | | | |
| **Test Data** | | | | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| Name: a  Country: Pakistan  City: Islamabad  Credit Card Detail: 0000-0000-0000-0000  Month: May  Year: 2022 | | | | | Error Message “Please provide valid name” | | | Order Placed Successfully | Test Fail |
|  | | | | | | | | | |
| **Test Case ID** | TC-023 | | | **Test Case Name** | | | Place order with invalid Credit Card | | |
| **Test Steps** | 1. Click on “Cart” on Main Menu. 2. Click on Place Order 3. Enter Name 4. Enter Country 5. Enter City 6. Enter invalid Credit Card Detail 7. Enter Month 8. Enter Year 9. Click on Purchase Button | | | | | | | | |
| **Test Data** | | | | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| Name: Nasir  Country: Pakistan  City: Islamabad  Credit Card Detail: a2c  Month: May  Year: 2022 | | | | | Error Message “Please provide valid credit card” | | | Order Placed Successfully | Test Fail |
|  | | | | | | | | | |
| **Test Case ID** | TC-019 | | | **Test Case Name** | | | Place order with all fake and random data | | |
| **Test Steps** | 1. Click on “Cart” on Main Menu. 2. Click on Place Order 3. Enter Name 4. Enter Country 5. Entry City 6. Enter Credit Card Detail 7. Enter Month 8. Enter Year 9. Click on Purchase Button | | | | | | | | |
| **Test Data** | | | | | **Expected Outcome** | | | **Actual Outcome** | **Result** |
| Name: a  Country: a  City: a  Credit Card Detail: a  Month: a  Year: a | | | | | Error Message “Please provide valid data in fields” | | | Order Placed Successfully | Test Fail |

**Task 6b:**

**Create a Test Suite containing the test scenarios, test cases, and relevant stuff.**

**Drive Link:**

[*https://drive.google.com/drive/folders/1XQ4KC42AUbG9mDsBBGR0Ssnl3JOg6QMu?usp=sharing*](https://drive.google.com/drive/folders/1XQ4KC42AUbG9mDsBBGR0Ssnl3JOg6QMu?usp=sharing)

.