Test Strategy document

**BEWAKOOF Shopping Web Application**

Document Details:

|  |  |
| --- | --- |
| Title | Test strategy document for Bewakoof Shopping web application |
| Version | 8.16.24.3 |
| Date | 15-07-2021 |
| File name | Test Strategy Bewakoof Shopping |
| Author | Team Bewakoof shopping |
| Contributor | Yogiramu, Nandana, konani, jagadeesh |

**TABLE OF CONTENT**

1. Scope 5
2. Application Overview 5
3. Test Approach 6
4. Test Types 6
5. Roles and Responsibilities 7
6. Environment Requirements 8
7. Testing Tools 8
8. Industry Standards to follow 8
9. Test deliverables 10

9.1 Testing Metrics

9.2 Requirement Traceability matrix

9.3 Test data

1. Risk and mitigation 14
2. Reporting tool 14
3. Test summary 14
4. Approvals 14

**1.SCOPE**

The Shopping application is termed as selling and buying behaviour of products and services over the internet. This online shopping system provides a 24×7 service, that is customers can surf the website, place orders anytime they wish to. It is also referred to as the sales of different items on the marketplaces in which money transaction activity takes place.

**Functional Requirements:**

* The main page consists of Logo of Application, Login/signup page, offers page, product page, profile page and helpline details under the bottom of Home page.
* The page consists of options such as product size, color, and type. There is a sorting feature to filter out products based on price, model, size. There is also the “Add to Cart” or “Go to cart” feature present in the category pages**.**
* The page Consists of the product title, description, product images, related products, Add to Cart feature, Product comparison, additional product information.
* The Page consist of list view, removing the product from the list, cash on delivery option, Select delivery option, card payment, pay now option.

**Non Functional Requirements:**

Capturing the behavior when a large number of people are using the software at the same time. Most of the time it is experienced that the servers are busy or unavailable due to heavy load.

* Validates that the system meets the expected response time. Evaluates that the significant elements of the application meet the desired response time.
* Whether the Application easy to use.

**2.OVERVIEW**  
Online Shopping is a lifestyle e-commerce web application, which retails various fashion and lifestyle products (Currently Men’s and Women’s wear). This project allows viewing various products available enables registered users to purchase desired products instantly using Card payment, Net banking and UPI processor (Instant Pay) and also can place order by using Cash on Delivery (Pay Later) option. This project provides an easy access to Administrators and Managers to view orders placed using Pay Later and Instant Pay options. The main purpose of this Shopping application is where product like clothes can be bought from the comfort of home through the Internet.

**3.TEST APPROACH**

* Test levels
* Test types
* Roles and responsibilities
* Environment requirements ( hardware and software requirements).

**4.TEST LEVELS**

During the testing process the testers tested application in various levels. Such as, Unit testing, Integration testing, system testing and user acceptance testing.

**Test Types:**

* **FUNCTIONAL TESTING:**

A Test is a set of preconditions, procedures (inputs or actions), and expected results used to determine whether a system works correctly. Test cases should have the following structure: a brief statement of purpose, description of precondition, actual test case inputs, expected outputs, description of expected postconditions, and execution history (date/person in charge/product version/pass or fail result).

* **USABILITY TESTING :**

Usability testing is defined as the evaluation of a product by testing it on potential users. To test how users will add multiple items in the cart. Is it easy for them to set the prices in their desired currency? Can they choose their preferred payment method without any hassle? if users can pick the payment methods they want, you can evaluate whether the website shows payment methods valid for the user’s country.

* **INTERFACE TESTING :**

ensure that end-users or customer should not encounter any problem when using Application. to check its user-friendliness as well. To verify security requirements while communication propagates between the systems and check if a solution is capable to handle network failures between an application server and website.

* **COMPACTIBILITY TESTING :**

It helps avoid issues related to versions updates, navigation flows, screen size adaptation, broken tables or frames, etc.  and the testing will done in

* Testing on PC, on different browsers like Safari, Chrome, Firefox, IE.
* Testing on different mobile devices that have different platforms like iOS, Android or Windows.
* Testing on networks like 4G, 3G or wifi.
* Testing on multiple operating systems such as Mac, Windows, Linux.
* **PERFORMANCE TESTING :**

Performance Testing is that type of software testing that pinpoints on how a system running the system performs under a particular circumstance. Performance testing measures depending on the benchmarks and standards. Performance testing helps the developers to eliminate the bottlenecks. Performance of a mobile or a web application is basically its capability of performing all the functions which it is supposed to do flawlessly without causing any delay or complication. its primary work, such as loading pages, showing the products, bringing out proper search results for the viewers, and loading the pages on time as well.

**5.ROLES AND RESPONSIBILITIES**

* Creating, enhancing, debugging, and running the test cases.
* Collating and monitoring the defect management process.
* Managing the changes and executing regression tests.
* Coming up with exact solutions for problems related to object identity and error handling.
* Interacting with customers/clients to solve the various issues they face and updating on the situation.
* Automating the design of a framework.
* Implementing it as per the structure of the project.
* Creating an automation test plan and getting approval.
* Identifying and selecting the automation test cases.
* Applying various designs and documenting the automation test strategy.
* Configuring Selenium Test Environment (STE) in order to set it up.
* Participating in Selenium Environment Setup with an Integrated Development Environment (IDE).

**6. ENVIRONMENT REQUIREMENTS**

**Software Environment:**

* Operating System- windows
* Java development toolkit.

**Hardware Environment:**

* Processor: Dual core
* RAM: 2GB
* Hard disk: 512GB.

**7.TESTING TOOLS**

Software Requirements:

* Selenium Testing tool.
* JIRA

Hardware Requirements:

* Ram -2gb.
* Operation system- windows, android.

**8.INDUSTRY STANDARDS USED**

* [**IEEE 1008-1987 - IEEE Standard for Software Unit Testing**](https://standards.ieee.org/standard/1008-1987.html)

An integrated approach to systematic and documented unit testing is defined. It uses unit design and unit implementation information, in addition to unit requirements, to determine the completeness of the testing. The testing process described composed of a hierarchy of phases, activities, and tasks and defines a minimum set of tasks for each activity. The standard can be applied to the unit testing of any digital computer software or firmware and to the testing of both newly developed and modified units.

* [**IEEE/ISO/IEC 29119-2-2013 - ISO/IEC/IEEE International Standard - Software and systems engineering —Software testing —Test processes**](https://standards.ieee.org/standard/29119-2-2013.html)

The purpose of the ISO/IEC/IEEE 29119 series of software testing standards is to define an internationally-agreed set of standards for software testing that can be used by any organization when performing any form of software testing. ISO/IEC/IEEE 29119-2 comprises test process descriptions that define the software testing processes at the organizational level, test management level and dynamic test levels. It supports dynamic testing, functional and non-functional testing, manual and automated testing, and scripted and unscripted testing. The processes defined in ISO/IEC/IEEE 29119-2 can be used in conjunction with any software development lifecycle model. Since testing is a key approach to risk-mitigation in software development, ISO/IEC/IEEE 29119-2 follows a risk-based approach to testing. Risk-based testing is a common industry approach to strategizing and managing testing. Risk-based testing allows testing to be prioritized and focused on the most important features and functions.

* [**IEEE/ISO/IEC 29119-3-2013 - ISO/IEC/IEEE International Standard - Software and systems engineering — Software testing —Test documentation**](https://standards.ieee.org/standard/29119-3-2013.html)

The purpose of the ISO/IEC/IEEE 29119 series of software testing standards is to define an internationally-agreed set of standards for software testing that can be used by any organization when performing any form of software testing. ISO/IEC/IEEE 29119-3 includes templates and examples of test documentation. The templates are arranged within clauses reflecting the overall test process description structure in ISO/IEC/IEEE 29119-2, i.e. by the test process in which they are being produced. Annex A contains outlines of the contents of each document. Annex B contains mappings ISO/IEC/IEEE 29119-2. Annex C contains an overview of the examples. Annexes D to S contain examples of the application of the templates. Annex T provides mappings to existing standards. The Bibliography for this part of ISO/IEC/IEEE 29119 is at the end of the document. ISO/IEC/IEEE 29119-3 supports dynamic testing, functional and non-functional testing, manual and automated testing, and scripted and unscripted testing.

**9.TEST DELIVERABLES**

* Test Scenario
* Test cases and data
* Requirement traceability matrix.
* Test summary report
* Test closure report

**9.1 Test Scenarios For shopping Web application:**

|  |  |  |
| --- | --- | --- |
| **SN** | **Test scenario ID** | **Test Objective/Test scenarios** |
| 1 | TS\_bewakoof\_001 | Validating the "Personal Details" feature as a new user. |
| 2 | TS\_bewakoof\_002 | Validate the "Username and Password" feature for the new user |
| 3 | TS\_bewakoof\_003 | Validate already existing users username and password |
| 4 | TS\_bewakoof\_004 | Checking for product page availablilty |
| 5 | TS\_bewakoof\_005 | Checking the product page,that user can select the desired attribute of the product. |
| 6 | TS\_bewakoof\_006 | Searching for the product by their names |
| 7 | TS\_bewakoof\_007 | Check the product availability,product price and price sorting |
| 8 | TS\_bewakoof\_008 | Selecting the product and adding product to the cart. |
| 9 | TS\_bewakoof\_009 | Selecting the Payment option for buying the product |
| 10 | TS\_bewakoof\_010 | After Payment transaction receipt will be received |
| 11 | TS\_bewakoof\_011 | Check the modules are present while opening the browser |

**9.2 Test cases Data:**

|  |  |
| --- | --- |
| **Test module** | **Test case Id with test case name** |
| 1.1 - Login | TS\_Login\_001- Validate the user name |
|  |  |
|  | TS\_Login\_002- Validate the password |
|  |  |
|  | TS\_Login\_003- Check with empty user name column |
|  |  |
|  | TS\_Login\_004-Forgot password |
|  | TS\_Login\_005-Verify & remember me after entering password |
|  | TS\_Login\_006-Changing password after giving forget password |
|  | TS\_Login\_07-User able to login after changing password |
|  |  |
|  |  |
|  |  |
| 1.2 - Register | TS\_Register\_001 - Check whether it is accepting valid email address |
|  |  |
|  |  |
|  | TS\_Register\_002 - Check whether it is a valid PW |
|  |  |
|  | TS\_Register\_003 - Whether DOB is filled or not |
|  | TS\_Register\_004 - Checking Check box for exclusive email update |
|  |  |
|  | TS\_Register\_005 - Check whether member button is working or not |
|  |  |
|  | TS\_Register\_006 - Check whether Password strength and conditions are visible |
|  |  |
|  | TS\_Register\_007 - Check whether mandratory fields has star symbol |
|  | TS\_Register\_008 - Check whether error shows while hitting submit without Email |
|  | TS\_Register\_009 - Check whether error shows while hitting submit without Password |
|  | TS\_Register\_010 - Check whether error shows while hitting submit without DOB |
|  | TS\_Register\_011 - Check whether error shows when the Password doesn't fulfill the conditions |
|  | TS\_Register\_012 - Check whether back to login works fine |
|  |  |
|  |  |
| 2.1 Categeorie | TS\_Cat\_004- Select the products for men/women |
|  | TS\_Cat\_006- Select the products for children |
|  |  |
|  | TS\_Cat\_007- Select the product from the brands |
|  |  |
| 2.2 Cart | TS\_Cat\_001- Adding the product to the cart |
|  |  |
| 3.1 Range | TS\_Ran\_001- Selecting the products based on the size and price |
|  |  |
|  | TS\_Ran\_002- Selected the products based on the medium price |
|  |  |
|  | TS\_Ran\_003- Selected the Brands based on the size and price |
|  |  |
|  | TS\_Ran\_004- Selected the Brands based on the High price |
|  |  |
| 4.1 Options | TS\_Opt\_001- Check whether the UPI Available or not |
|  |  |
|  | TS\_Opt\_002- Check whether the NetBanking is Available or not |
|  |  |
|  | TS\_Opt\_003- Check whether the Credit/debit card is Available or not |
|  |  |
|  | TS\_Opt\_004- Check whether the COD is Available or not |
|  |  |
| 5.1 Status | TS\_Status\_001- Check whether the Payment is done successfully by NetBanking |
|  |  |
|  | TS\_Status\_002- Check whether the Payment is done successfully by Credit/Debit card |
|  | TS\_Status\_003- Check whether the Payment is done successfully by UPI |

**9.3 Testing Metrics:**

Total number of test cases=37

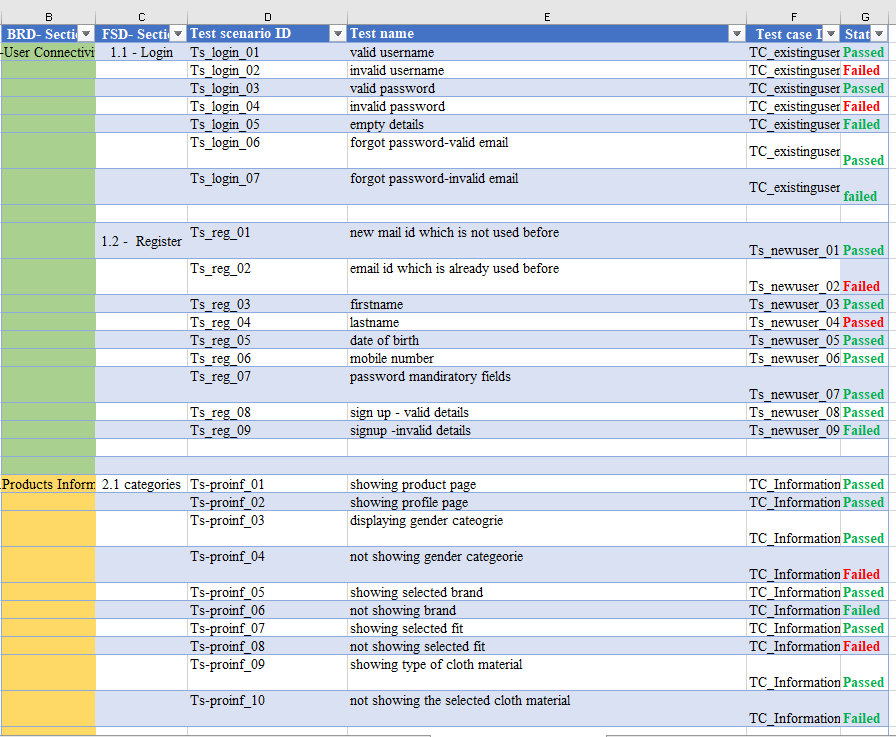
Number of test cases passed=33

Number of test cases failed=03

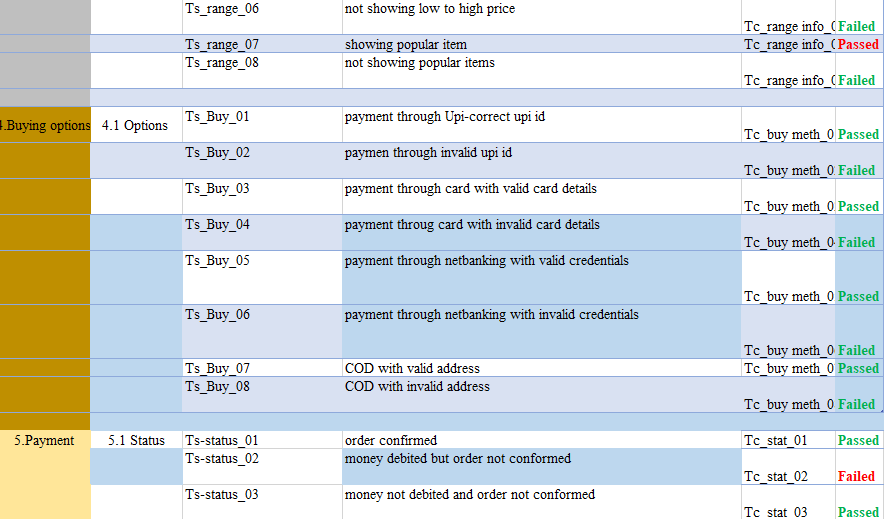
Number of test cases rejected=01

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team member | Number of test cases assigned | Number testcases passed | Number of test cases failed | status |
| Joshna | 8 | 8 | 0 | Done |
| Nandhana | 10 | 8 | 2 | Not done |
| Jagadeesh | 8 | 8 | 0 | Done |
| Yogiramu | 11 | 9 | 2 | Not done |

**9.4 Requirement Traceability Matrix:**







**10.RISK AND MITIGATION**

* Broken Access control
* Payment security
* Use secure passwords.

**11.REPORTING TOOL**

JIRA is the reporting tool for the Project.

**12.TEST SUMMARY**

A Test Strategy document is created for sshopping web application as per the content. It needs to be reviewing for sign-off by all entities involved in project management, business team, development team, and system administration Team.

**13.APPROVALS**

|  |  |
| --- | --- |
| **Approved by role** | **Approved by Name** |
| Project Coordinator | Anu meha |
|  |  |