**Project Name**:**Festicket**

**Project ID**: XYZ

**Author**: Dilip Patel

**Date of Creation**: 14/04/2019

**Project Overview**: The aim of the project is to enable a digital experience to our customer. The high-level feature that will be implemented is:

* Search Product from a responsive UI

The purpose of this document is to define:

* Test approach (High level) for Exploratory testing, UI testing & Functional testing
* Entry & Exit criteria for the tests under scope.

The objective of the UI testing is to determine if the links, pages and database connectivity is appropriately established and there are no broken links, distorted pages and disconnection with the database while extracting the content.

Whereas Functional testing focuses on if all the intended objectives of the application are met and there are no issues with the outcome. It also helps identifying any dependencies, ensures data integrity is maintained across different components.

**In Scope:**

* Exploratory testing
* UI Testing
* System Integration testing/Functional testing of the end to end business processes.

**Out of Scope:**

* Performance testing
* Operational Acceptance testing.

**Test Design:**

* The use cases in the Structural requirement specifications will be used to design the test scenarios.
* The scenarios will be reviewed by respective stakeholders
* Scenarios will be prioritised for execution based on their business importance and severity.
* Scenarios will be signed off by the reviewers, agile team and SME’s before execution.

**Test Execution:**

* Execution Method: Manual & Automation
* All the scripts that are signed off by the SME’s will be executed.

**UI testing/Usability testing**:

* Home page, product page, special offers & Search Functionality
* Product page, category/Sub category Navigation
* Product description including quantity
* Shopping cart – add and remove product from list, payment option, delivery option

**Functional Testing:**

* Login and Signup option
* Search functionality
* Product review
* Add/remove functionality in the shopping cart
* Checkout process
* Payment option

**Automation/Manual scope**:

* All the Search Product, add to cart, payment scenarios with Random selection will be automated
* The scenarios which require specific test data will be performed manually.
* The basic UI checks such as Links, Forms, Page displays, Page navigation and Logo display will be automated.
* The URL Launch will be automated
* The customer registration module (if being tested separately under Module testing) can be automated to detect any errors while registration.
* Support multiple browsers

**High priority scenarios:**

1. Sign up and Login
2. Add/remove product in the shopping Cart
3. Checkout process
4. Payment gateway and payment processing
5. Discount product
6. Search functionality

**Step to run test on different browser:**

This project is written in Intellij IDEA

Open runcukes file

Go to Run configuration file

Go to VM Option and change browser =chrome, firefox or ie

Issues and/or challenges you’ve come across and how did you overcome them?

When I click on login button after filling all required information, I was getting test failed. To overcome above issue I have to use implicitly wait.

If you had more time what would you do differently?

If I had more time I would like to add more scenario to verify login function like compare with url, text.

What other test case would you automate and why?

On basis of high priority I would automate following test case

All the Search Product, add to cart, payment option, delivery option, checkout, sign-in functionality.

Above case are more time and money consuming for manual testing and also I will automate smoke and sanity test when new functionality added.

What dependencies or 3rd party framework/libs/modules used and why?

To automate above task I have to use following dependencies:

Selenium Webdriver manager, cucumber Junit, selenium java, cucumber java, gherkin and core java as programming language.

Resources used

My study material, previous project