UI Test & Strategy plan for platform to distribute product reviews from partner sites for Gartner

**Prepared by**

Nitin Mittal

[nitinkmital@gmail.com](mailto:nitinkmital@gmail.com)

+91-9034756975

**Dated**

December 16, 2019

**Edition 1.0**

**Table of Contents**

[1 Summary 3](#_Toc27407606)

[2 Audience 3](#_Toc27407607)

[3 Automation Scope 3](#_Toc27407608)

[1.1 In Scope 3](#_Toc27407609)

[1.1.1 Environments 3](#_Toc27407610)

[Operating system 3](#_Toc27407611)

[Stack (Test Enviroment) 4](#_Toc27407612)

[Tech Stack Used 4](#_Toc27407613)

[1.2 Out of Scope 4](#_Toc27407614)

[Stack (Test Enviroment) 4](#_Toc27407615)

[4 Automation Strategy 4](#_Toc27407616)

[4.1 Overview of Automation Testing 4](#_Toc27407617)

[4.2 Automation Framework 4](#_Toc27407618)

[4.3 Test Cases (GRMT (Gartner Review Management System)) 5](#_Toc27407619)

[4.4 Defect Management 7](#_Toc27407620)

[5 Automation Deliverables 8](#_Toc27407621)

[6 Automation Criteria 8](#_Toc27407622)

[6.1 Entry criteria 8](#_Toc27407623)

[6.2 Suspension Criteria 8](#_Toc27407624)

[6.3 Resumption Criteria 8](#_Toc27407625)

[6.4 Exit criteria 9](#_Toc27407626)

# Summary

This document is designed to communicate the implementation strategy, automation plan to successfully automate **UI to be displayed on customer site** for distribute product reviews system for Gartner.

The document would be detailing the following:

* **Automation Scope** – The overall Automation coverage and approach to the test automation will be described.
* **Automation Approach** identifies the methodology that will be used in test automation of API.
* **Automation Deliverables** outline the deliverables.
* **Assumptions, Dependencies, and Risks** describe the assumptions, dependencies, and risks of the project’s road map.

# Audience

This document is intended for the QA Analysts, Project Managers, Software Developers and Stakeholders that may be involved in developing, testing, deployment and operation of Gartner.

# Automation Scope

This automation plan presents the scope, approach, risks, issues, responsibilities, assumptions, automation environment required to create a reliable, feasible, and extensible solution to automate API’s of Platform to get, send and receive reviews.

The main objectives of the automation testing are:-

|  |  |
| --- | --- |
| **Objectives** | **Description** |
| 1 | Define a maintainable API automation framework. |
| 2 | To successfully automate **UI to be displayed on customer site** for distribute product reviews system for Gartner. |

## In Scope

* Automate the UI to be displayed on customer site for platform to store , process and distribute reviews with the help of selenium
* Framework design and development
* Validate the contents on UI using GET API

## 

## Environments

### Operating system

* Windows 7 or above

### Stack (Test Enviroment)

* Pre-production/Test of UI
* Pre-Production/Test of API endpoints

### Tech Stack Used

* JAVA
* Selenium
* Extent Report
* Log4J
* TestNG

## Out of Scope

* All the third party/external applications will be out of scope.
* Email and Telephony related functionalities.
* Maintenance of test infrastructure
* Non-functional testing

### Stack (Test Enviroment)

* Stage, Pre-Production and Production(live) for both API and UI

# Automation Strategy

## Overview of Automation Testing

Automation testing is carried out to assist QA and reduce their effort w.r.t testing the application, confirm system meets the requirements and goals or otherwise, establish baseline for future tests, support constant changes to application / framework, collate automation related information on regular basis to help Solution Architect, Infrastructure, Development teams, etc, to make informed decision related to overall fitness of the system under test.

The following are the types of functional test automation -

* **UI Testing –** This system need to automate the verification of the UI to be displayed on customer site, how the reviews are displayed over there. We have to assert the UI and then validate the data corresponding to data got from GET API on UI

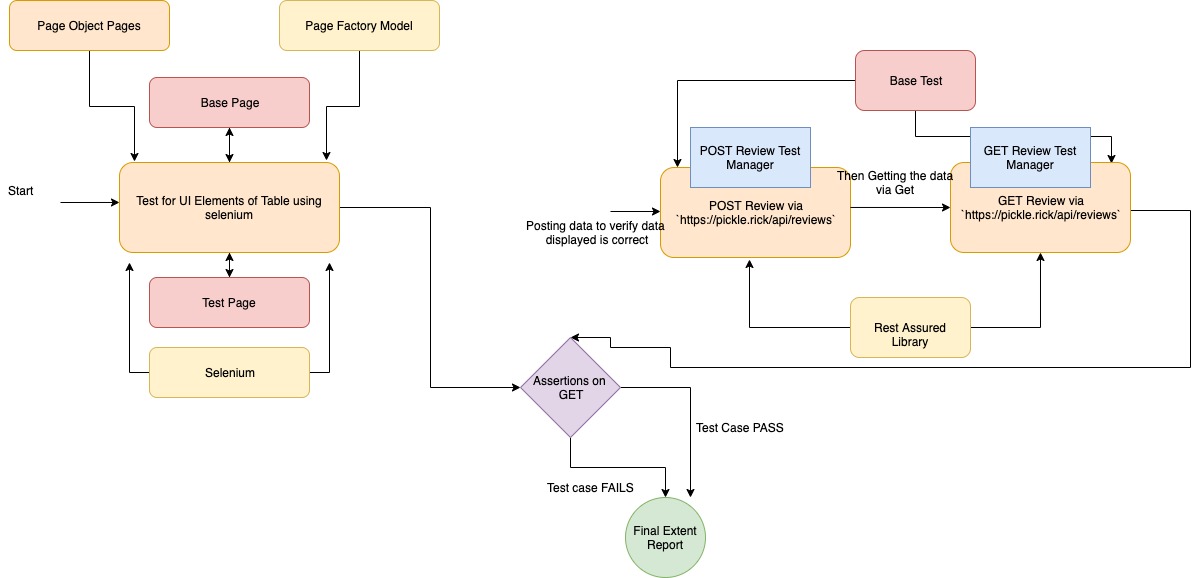
## Automation Framework

The Automation Framework should be consist of some basic test automation frameworks for JAVA like selenium , Rest Assured And TestNG. Our Automation includes to test the Table need to be tested for UI elements, The Same tabular view to be displayed on different partner sites.

Firstly we are going to test the UI elements of Table are present or not, Then we will be testing whether the data fetched from GET API is same as the data displayed in UI for different strategy and test case.

We are submitting data using POST API according to test case, verify the UI of the system and then we will be asserting the output from GET API and UI tabular display are same or not. Accordingly the result need to be main In the

P.S :- I am considering that GET API is returning all the reviews we are having in database



## Test Cases (GRMT (Gartner Review Management System))

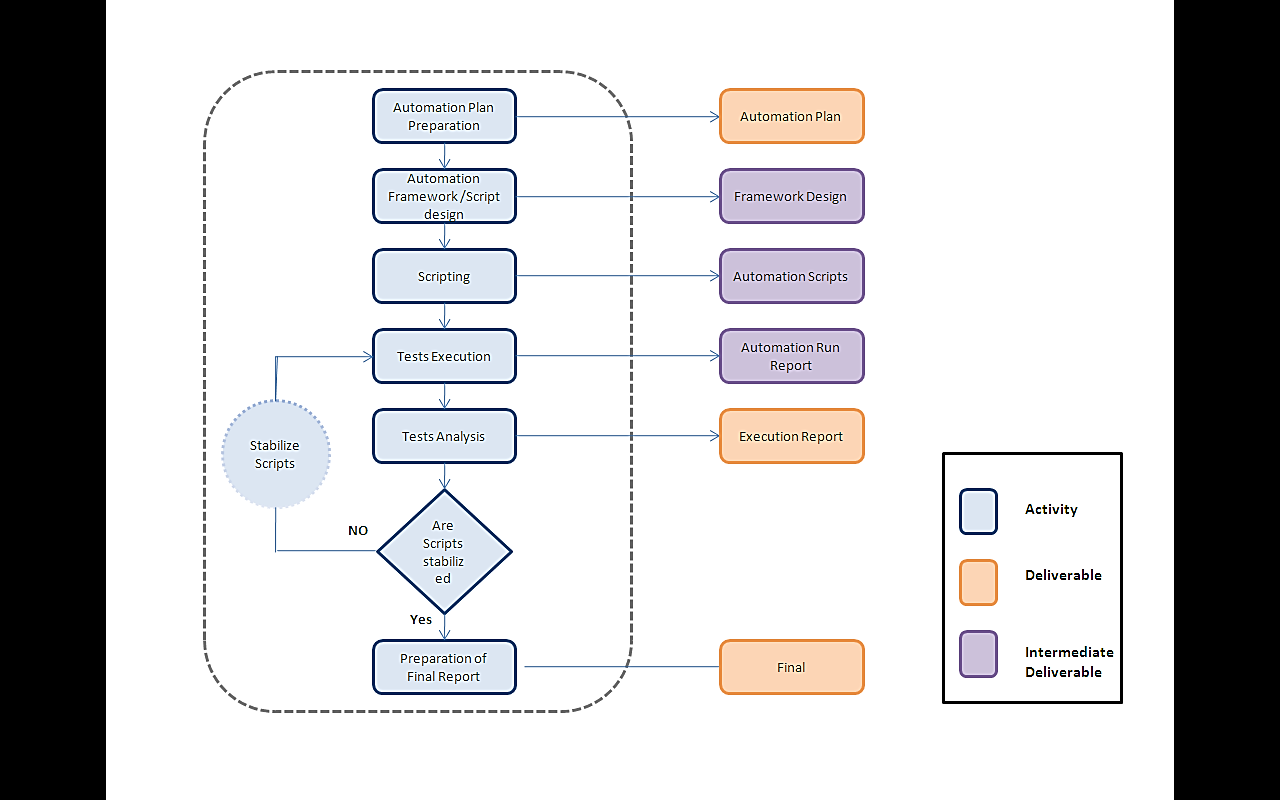
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Summary | | Expected GET API Output | | Comment |
| GRMT-UI-1 | Verify that single Record POST in EN language via POST API can be extracted via GET API and UI | | 1 row in output  Language EN | | Check the DB after POST and Purge DB after Test Case Execution |
| GRMT-UI-2 | Verify that single Record POST in ES language via POST API can be extracted via GET API and UI | | 1 row in output  Language ES | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-3 | Verify that multiple(5) different (different in text) Reviews in FR Language POST via POST API can be extracted via GET APIand on UI | | 5 rows in output  Language FR  Isduplicate is NO for all | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-4 | Verify that multiple(5) different (different in text) Reviews in EN Language POST via POST API can be extracted via GET API on UI | | Same 5 rows in output  Language EN  IsDuplicate is NO for all | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-5 | Verify that multiple(5) different (different in text) Reviews in ZH Language POST via POST API can be extracted via GET API and comparI with UI | | Same 5 rows in output  Language ZH  IsDuplicate is NO for all | | Check the DB after POST and Purge DB after Test Case Execution  (Chinese) | |
| GRMT-UI-6 | Verify that multiple(5) different (different in text) Reviews in JA Language POST via POST API can be extracted via GET API | | Same 5 rows in output  Language JA  IsDuplicate is NO for all | | Check the DB after POST and Purge DB after Test Case Execution  (Japanese) | |
| GRMT-UI-7 | Verify that 2 same records POST (Same in Text) with same rating and language EN via POST can be extracted via GET API | | Same 2 rows in output  Language EN  IsDuplicate is NO for one and YES for other | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-8 | Verify that 2 different records POST (Same in Text) with different rating and language KU via POST can be extracted via GET API | | Same 2 rows in output  Language KU  IsDuplicate is NO for all | | Check the DB after POST and Purge DB after Test Case Execution  (Korean) | |
| GRMT-UI-9 | Verify that 2 different records POST (different in Text) with same rating and language EN via POST can be extracted via GET API | | Same 2 rows in output  Language EN  IsDuplicate is NO for alUIl | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-10 | Verify that 5 records in which 2 are different records POST (different in Text) with different rating and 3 are same with same text and rating with language EN via POST can be extracted via GET API | | Same 5 rows in output  Language EN  IsDuplicate is NO for 2 Records and YES for 3v records | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-11 | Verify that 5 records in which 2 are different records POST (different in Text) with different rating and 3 are same with same text and rating with language ES via POST can be extracted via GET API | | Same 5 rows in output  Language ES  IsDuplicate is NO for 2 Records and YES for 3 records | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-12 | Verify that 5 records in which 2 are different records POST (different in Text) with different rating and 3 are same with same text and rating with language FR via POST can be extracted via GET API | | Same 5 rows in output  Language FR  IsDuplicate is NO for 2 Records and YES for 3 records | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-15 | Verify that A review Containing Multiple Language text in single POST via POST Can be extracted via GET | | 1 same rows in output  Language ? not to be sure | | Check the DB after POST and Purge DB after Test Case Execution | |
| GRMT-UI-16 | Verify that two review having same text but one is having space and other is having no space can be treated as two different reviews | | 2 same rows in output  Language EN  isDuplicate NO for both rows | | Check the DB after POST and Purge DB after Test Case Execution | |
|  | |  | |

## Defect Management

* Defect will be logged in JIRA
* Identification of defects that are found during the test execution phase in Automation will be mapped to the corresponding tests with pass, fail, defects captured etc.
* QA will define the severity of the bugs based on the Functional Specification documents. QA lead & dev lead will set the priority after triage
* These defects are to be resolved by the development team and the fix priorities and meanings are mentioned in below table -

|  |  |
| --- | --- |
| **Fix Priority** | **Definition** |
| Priority 0/1 | **Critical**  This needs Immediate fix since no further testing is possible during the test stage in question. |
| Priority 2 | **High**  Resolve as soon as possible, since it is a severe problem that seriously affects the testing stages. Other tests from the test stage can still be run. |
| Priority 3 | **Medium**  It can wait until a new build because the functionality works, but produces wrong output results. |
| Priority 4 | **Low**  Resolve only when more important defects are resolved since it is Cosmetic problem. |

# Automation Deliverables

This diagram below outlines the Automation testing process and deliverables as part of the Ladbrokes 

# Automation Criteria

## Entry criteria

* Entry criteria for this automation testing are, when system is ready to test
* The Automation plan is completed and signed off, before start of the test creation / execution.
* Availability of required resources, tools and clarifications on time.

## Suspension Criteria

* Non resolution of high priority defects identified during a test execution phase, which affect the subsequent test cycles or phases
* Changes to functionality and scenarios resulting in changes to the use cases
* Network related errors like disruption in network links / connectivity
* System Hardware/ Software are not able to suit automation requirements.
* Lack of test data to test the application system during the creation of automation scripts.

## Resumption Criteria

* Development Team shall address and fix all the High priority defects
* All the Network related errors / Database related errors to be resolved
* Required test data to test the application system and ready for scripting.

## Exit criteria

* All major functionality scenarios successfully automated and executed, no outstanding defects at severity 1 and 2, with acceptable number of lower severity defects
* Issues w.r.t framework should be fixed and retested. Any unresolved issues that will be carried over into Production environment must be documented and signed-off.
* Automation Completion Report produced, signed off by stakeholders and Test Results secured
* The test execution is said to be complete when the tests are run with various input and output parameters and successfully test the system.
* The test is said to be completed when all the test deliverables mentioned in section 4 are achieved.