**Table of Contents**

1. INTRODUCTION
2. SCOPE
3. TEST APPROACH
4. ROLES AND RESPONSIBILITIES
5. ENTRY AND EXIT CRITERIA

5.1 Entry Criteria

5.2 Exit Criteria

1. TEST STRATEGY

6.1 QA role in the test process

6.2 Bug life cycle

6.3 Testing types

6.4 Bug Severity and Priority Definition

1. RESOURCE AND ENVIRONMENT NEEDS

7.1 Testing Tools

7.2 Configuration Management

7.3 Test Environment x Support level

1. TEST SCHEDULE
2. APPROVALS
3. TERMS/ACRONYMS

**Test Plan**

**Project “**CALIFORNIA MARKETING**”**

1. INTRODUCTION

Customer wants a perfect website, which passes the full cycle of manual and automation testing. Given the specificity of the site it is very important to have the same quality and the site.

The Test Plan has been created to facilitate communication within the team members. This document describes approaches and methodologies that will apply to the unit, integration and system testing of the <https://qasvus.wixsite.com/ca-marketing>. It includes the objectives, test responsibilities, entry and exit criteria, scope, schedule major milestones, entry and exit criteria and approach. This document has clearly identified what the test deliverables will be, and what is deemed in and out of scope.

1. SCOPE

The document mainly targets the GUI testing and validating data in report output as per Requirements Specifications provided by Client.

2.1 Functions to be tested:

* Manual functional
* Automation
* Performance
* Security
* API

2.2 Functions not to be tested.

2.3. Not other than mentioned above in section 2.1

1. TEST APPROACH

The approach that is used is Analytical therefore, in accordance with requirements-based strategy, where an analysis of the requirements specification forms the basis for planning, estimating and designing tests.

The project is using an agile approach, with weekly iterations. At the end of each week, the requirements identified for that iteration will be delivered to the team and will be tested.

1. ROLES AND RESPONSIBILITIES

| **Role** | **Staff Member** | **Responsibilities** | |
| --- | --- | --- | --- |
|  |  |  | |
| Project  Manager | \*\*\*\* | 1. Acts as primary contact for development and QA  team.  2. Responsible for Project schedule and the overall  success of the project. | |
|
|
|
|  |  |  | |
| QA Lead | \*\*\*\* | 1. Participation in the project plan creation/update process. | |
|  |  | 2. Planning and organization of the test process for the release. | |
|  |  | 3. Coordinate with QA analysts/engineers on any | |
|  |  | issues/problems encountered during testing. | |
|  |  | 4. Report progress on work assignments to the PM. | |
|  |  |  | |
| QA | Juli Diva | 1. Understand requirements. | |
|  | \*\*\*\* | 2. | Writing and executing Test cases. |
|  |  | 3. | Preparing RTM. |
|  |  | 4. | Reviewing Test cases, RTM. |
|  |  | 5. | Defect reporting and tracking. |
|  |  | 6. | Retesting and regression testing. |
|  |  | 7. | Bug Review meeting. |
|  |  | 8. | Preparation of Test Data. |
|  |  | 9. | Coordinate with QA Lead for any issues or problems |
|  |  | encountered during test preparation/execution/defect | |
|  |  | handling. | |
|  |  |  |  |

1. ENTRY AND EXIT CRITERIA

5.1 Entry Criteria

* + All test hardware platforms must have been successfully installed, configured, and functioning properly.
  + All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
  + All the standard software tools including the testing tools must have been successfully installed and functioning properly.
  + Proper test data is available.
  + The test environment such as lab, hardware, software, and system administration support should be ready.
  + QA resources have completely understood the requirements.
  + QA resources have sound knowledge of functionality.
  + Reviewed test scenarios, test cases and RTM.

5.2 Exit Criteria

* A certain level of requirements coverage has been achieved.
* No high-priority or severe bugs are left outstanding.
* All high-risk areas have been fully tested, with only minor residual risks left outstanding.
* Cost – when the budget has been spent.
* The schedule has been achieved.

1. TEST STRATEGY

6.1 QA role in test process:

* Understanding Requirements.
* Requirement specifications will be sent by client.
* Understanding of requirements will be done by QA.
* Preparing Test Cases:

QA will be preparing test cases based on exploratory testing. This will cover all scenarios for requirements.

- Preparing Test Matrix:

QA will be preparing a test matrix which maps test cases to respective requirements. This will ensure the coverage for requirements.

- Reviewing test cases and matrix:

* Peer review will be conducted for test cases and test matrix by QA Lead.
* Any comments or suggestions on test cases and test coverage will be provided by the reviewer respective Author of Test Case and Test Matrix.
* Suggestions or improvements will be re-worked by the author and will be send for approval.
* Re-worked improvements will be reviewed and approved by the reviewer.

- Creating Test Data:

Test data will be created by respective QA on client's developments/test site

based on scenarios and Test cases.

- Executing Test Cases:

* Test cases will be executed by respective QA on the client's development/test site based on designed scenarios, test cases and Test data.
* Test result (Actual Result, Pass/Fail) will be updated in test case document Defect Logging and Reporting: QA will be logging the defect/bugs in Word document and JIRA, found during execution of test cases. After this, QA will inform the respective developer about the defect/bugs.

- Retesting and Regression Testing:

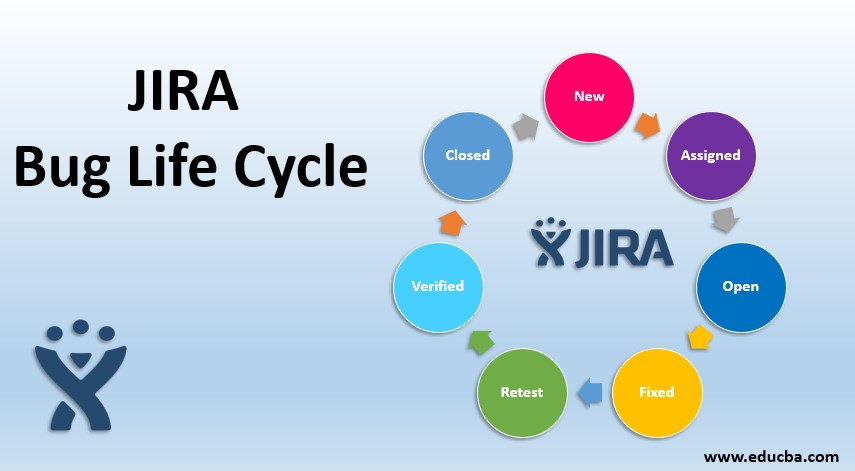
Retesting for fixed bugs will be done by respective QA once it is resolved by the respective developer and bug/defect status will be updated accordingly. In certain cases, regression testing will be done if required.

- Deployment/Delivery:

* Once all bugs/defects reported after complete testing are fixed and no other bugs are found,the report will be deployed to the client's test site.
* Once a round of testing will be done by QA on the client's test site if required Report will be delivered along with sample output by email to the respective lead and Report group.

6.2 Bug life cycle:

All the issues found while testing will be logged into JIRA.



6.3 Testing types

**Manual Functional Testing:**

**PURPOSE**: Functional testing will be performed to check the functions of an application. The functional testing is carried out by feeding the input and validating the output from the application

**SCOPE:** The below table details the scope of Functional test

| Modules | Test Scripts |
| --- | --- |
| Log In | Verify that the user can log into the site using an existing mail and password |
| Add new address | Verify that the user can click on Add New Address button |
| Fill out the form | Verify that the user can enter data in the field |
| Submit | Verify that the user can submit the entered data by clicking on Add address button |
| A new address is present | Verify that the user can see a newly added address |

**TESTERS**: Testing Team

**METHOD**: The test will be performed according to Functional scripts

**ENVIRONMENT:** OS: Windows 11/64, Browsers: Chrome latest, Mozilla

Firefox latest, Microsoft Edge latest

**TIMING:** after Exploratory test is completed

**Automation Testing:**

**PURPOSE:** This test focuses on creating automation scripts based on

manual test cases

**TESTERS:** Testing Team

**SCOPE:** The same as in Manual Functional Testing

**TOOLS:** The testing team should use PyCharm as a main IDE, Python as a main

language, Selenium WebDriver as a main framework for test scripts

automation, and BrowserStack to run tests in various environments

**TIMING:** After manual testing is done and all critical issues are resolved

**Performance testing:**

**PURPOSE:** This test is performed to measure the speed, responsiveness, stability

of the site and also how well the page is built for optimal performance

**TESTERS:** Testing Team

**SCOPE**: https://qasvus.wixsite.com/ca-marketing

**TOOLS:** The testing team should use Google Lighthouse, GTMetrix

**TIMING:** After manual testing is done and all critical issues are resolved

**Security testing:**

**PURPOSE**: This test is performed to reveal current or potential security

vulnerabilities

**TESTERS**: Testing Team

**SCOPE**: <https://qasvus.wixsite.com/ca-marketing>

**TOOLS**: The testing team should use Snyk and Mozilla Observatory

**TIMING**: After manual testing is done and all critical issues are resolved

**API testing:**

**PURPOSE:** API tests are performed to determine if the developed APIs meet the

expectations when it comes to the functionality, performance,

reliability and security of the website

**TESTERS:** Testing Team

**SCOPE:** https://qasvus.wixsite.com/ca-marketing

**TOOLS**: The testing team should use Postman API as a platform for building

and using APIs and JavaScript as a language for test scripts

**TIMING:** After manual testing is done and all critical issues are resolved

6.4 Bug Severity and Priority Definition

Bug Severity and Priority fields are both very important for categorizing bugs and prioritizing if and when the bugs will be fixed. The bug Severity and Priority levels will be defined as outlined in the following tables below. Testing will assign a severity level to all bugs. The Test Lead will be responsible to see that a correct severity level is assigned to each bug.

The QA Lead, Development Lead, and Project Manager will participate in bug review meetings to assign priority to all currently active bugs. This meeting will be known as “Bug Triage Meetings”. The QA Lead is responsible for setting up these meetings on a routine basis to address the current set of new and existing but unresolved bugs.

**Severity List**

| Severity ID | Severity | Severity Description | |
| --- | --- | --- | --- |
| 1 | Highest | The module/product crashes or the bug causes non-  recoverable conditions. System crashes or  database or file corruption, or potential data loss, program  hangs requiring reboot are all examples of a Severity 1 bug. | |
|
|
|
|
|  |  |  | |
| 2 | High | Major system component unusable due to failure or incorrect  functionality. Severity 2 bugs cause serious problems such as a lack of functionality, or insufficient or unclear error messages  that can have a major impact to the user, prevents other areas  of the app from being tested, etc. Severity 2 bugs can have a  work around, but the work around is inconvenient or difficult. | |
|
|
|
|
|
|  |  |  | |
| 3 | Medium | Incorrect functionality of component or process. There is a  simple work around for the bug if it is Severity 3. | |
|
|  |  |  | |
| 4 | Low | Documentation errors or signed off Severity 3 bugs. | |
|  |  |  | |

**Priority List**

| Priority | Priority Level | Priority Description |
| --- | --- | --- |
| 1 | Highest | This bug must be fixed immediately; the product cannot  ship with this bug. |
|
|  |  |  |
| 2 | High | These are important problems that should be fixed as soon  as possible. It would be an embarrassment to the company if this bug shipped. |
|  |  |  |

| 3 | Medium | The problem should be fixed within the time available. If  the bug does not delay shipping date, then fix it. | | |
| --- | --- | --- | --- | --- |
|
|
|  |  |  | | |
| 4 | Low | It is not important (at this time) that these bugs be  addressed. Fix these bugs after all other bugs have been  fixed. Enhancements/ Good to have features incorporated-  just are out of the current scope. | | |
|
|
|  |  |  | | |
| 5 | Lowest | Documentation errors or signed off Low 4 bugs. | | |
|  |  |  | | |

1. RESOURCE AND ENVIRONMENT NEEDS

7.1 Testing Tools

| **Process** | **Tool** |
| --- | --- |
|  |  |
| Test case creation | Microsoft Word, Microsoft Excel, JIRA |
|  |  |
| Test case tracking | JIRA, Confluence |
|  |  |
| Test case execution | Manual, Selenium \*\*\*\* |
|  |  |
| Test case management | Microsoft Excel, JIRA, Confluence |
|  |  |
| Defect management | Microsoft Word, JIRA, Confluence |
|  |  |
| Test reporting | JIRA |
|  |  |
| Checklist creating | Microsoft Excel, JIRA |
|  |  |
|  |  |
|  |  |

7.2 Configuration Management

* Documents CM: SVN
* Code CM: Git

7.3 Test Environment x Support level 1 (browsers):

* Windows 10 : Edge, Chrome (latest), Firefox (latest), Safari (latest)
* Mac OS : Chrome (latest), Firefox (latest), Safari (latest)
* Android: Chrome (latest), Firefox (latest)

Support level 1 (devices):

iPhone 6/6s, \*\*\*\*\*

Support level 2:

* Windows 7: IE 9+, Chrome (latest), Firefox (latest), Safari (latest)
* Windows XP: IE 8, Chrome (latest), Firefox (latest), Safari (latest)
* Support level 3: x anything else

8. TEST SCHEDULE

| **Task Name** | **Start** | **Finish** | **Effort** | **Comments** |
| --- | --- | --- | --- | --- |
| Test Planning | 02.04.2023 | 02.11.2023 |  | documentation |
| Review Requirements documents | 02.12.2023 | 02.13.2023 |  |  |
| Create test basis | 02.14.2023 |  | with team | flex |
| Staff and train new test resources | 02.15.2023 |  |  | flex as needed |
| Black box testing | 02.15.2023 | 02.29.2023 |  |  |
| Exploratory testing | 02.15.2023 | 02.29.2023 |  |  |
| ADHOC testing | 02.15.2023 | 02.29.2023 |  |  |
| Positive testing | 02.15.2023 | 02.29.2023 |  |  |
| Negative testing | 02.15.2023 | 02.29.2023 |  |  |
| Functional testing – Iteration 1 | 02.15.2023 | 02.29.2023 |  |  |
| Iteration 1 deploy to QA test environment | 02.23.2023 | 02.26.2023 |  | re-test |
| Regression testing | 06.26.2023 | 11.03.2023 |  |  |
| Functional testing – Iteration 2 | 02.26.2023 | 11.03.2023 |  | re-test |
| System testing | 02.26.2023 | 11.03.2023 |  |  |
| Regression testing | 02.26.2023 | 11.03.2023 |  |  |
| UAT | 02.26.2023 | 11.03.2023 |  |  |
| Resolution of final defects and final build testing | 12.03.2023 | 03.20.2023 |  |  |
| Deploy to Staging environment | 03.23.2023 |  |  |  |
| Performance testing | 03.24.2023 | 04.07.2023 |  |  |
| Release to Production | 04.08.2023 |  |  |  |

9. APPROVALS:

|  | Project Manager | QA Lead |
| --- | --- | --- |
| Name | \*\*\*\* | \*\*\*\*\* |
| Signature |  |  |

10. TERMS/ACRONYMS

The below terms are used as examples, please add/remove any terms relevant to the document.

| TERM/ACRONYM | DEFINITION |
| --- | --- |
|  |  |
| API | Application Program Interface |
|  |  |
| GUI | Graphical user interface |
|  |  |
| PM | Project manager |
|  |  |
| UAT | User acceptance testing |
|  |  |
| CM | Configuration Management |
|  |  |
| QA | Quality Assurance |
|  |  |
| RTM | Requirements Traceability Matrix |
|  |  |