**Software Test Plan**

|  |  |
| --- | --- |
| **Document name** | **Software Test Plan** |
| **Project Name** | **Teach-Me Application** |
| **Version no.** | **2.0** |
| **Release date** | **2020-03-15** |
| **Classification** | **Internal** |

**1. Introduction**

This document is a detailed guide for listing the testing activities that should be carried out for the Teach-Me App. It describes the software test environment for testing, identifies the tests to be performed, and provides the schedule for test activities.

## **1.1 Purpose of the Document**

The Purpose and objective of STP is to outline the test strategy and overall test approach for Teach-Me App. This includes scope of STP, scope testing, testing criteria, test methodologies, traceability, resources required, assumptions considered, and dependencies and estimated schedule.

## **1.2 Scope of STP**

The scope of testing is explained in the document, it is to test the operating characteristics of the TeachMe application that runs on browsers as mentioned in the document. The tests are organized by requirement categories such as usability, functionality, security, etc. The procedure for carrying out testing in terms of preparation of test cases, test environment setup, defects logging and reporting are explained.

## **1.3 Acronyms and Definitions**

This sub-section provides the description of acronyms required to interpret the STP.

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Acronyms** | **Definitions** |
| 1. | STP | Software Test Plan |
| 2. | TC | Test Case |
| 3 | API | Application Programming Interface |

# 

# **2. Scope of Testing**

## **2.1 Features to be tested**

|  |  |
| --- | --- |
| **Sr. No.** | **Features / Functions to be tested** |
| 1. | Registration Page |
| 2. | Login Page |
| 4. | User Dashboard |
| 5. | Topics Page |
| 6 | Test Page |
| 7 | Reports/Statistics Page |
| 8 | APIs for User, Test, Subject, Service, and Response controllers. |

# **3. Test Entrance/Exit Criteria**

This section refers to identifying what must be present to effectively begin system testing (entrance criteria) and also determining the condition the system will be in when system test completes (exit criteria).

## **3.1 Entry Criteria**

* All Code should be unit tested.
* Test cases reviewed and approved.
* Environment setup should be ready before testing.

## **3.2 Exit Criteria**

* All planned tests have been executed.
* All bugs have been opened in the Defect tracking system.
* All critical bugs have been retested and resolved.
* Documenting system test results.

# 

# **4. Test Environment**

This section refers to the testing environment and Client supplied Hardware / Software.

## **4.1 Hardware**

This section states all the hardware resources required for the project.

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name of the Hardware** | **Hardware Configuration** |
| 1. | Client | Any computer with internet access |
| 2. | Web server | Apache |
| 3. | Database Server | MySQL Server |

# 

# **5. Test Strategy**

## **5.1 Testing Process**

The objective of testing is to ensure

* Software reliability
* Software quality
* System Assurance
* Optimum performance and capacity utilization.

**Usability**

* Avoid Poor site navigation

**System Integration**

* Data Interface format
* Updates
* Integrated performance

**Login**

* Login capability

## **5.3 Type of Testing**

The different types of testing that may be carried out in the project are as follows:

* Smoke Testing
* Integration Testing
* System Testing
* Regression Testing
* API Testing

***The different types of testing that may be carried out in the project are as follows:***

**Smoke testing:**

Smoke test is an initial set of tests that determine if a new software build is performing well enough to accept it for a major testing effort. It verifies the major functionalities at a high level. The Smoke test scenarios emphasize more on breadth than depth. If the test fails, the build is returned to developers without testing.

Test strategy for smoke testing will be manual.

**Integration testing:**

Integration testing is performed to establish whether the components correctly interact with each other according to the specifications.

Test strategy for integration testing for the project Integration will be performed using top–down.

**System Testing:**

It tests the system as a whole. It is a functional testing, performed to validate that the application meets requirement specifications.

Test strategy for system testing for the project Testing will be carried out once integration and functional testing is over to check end-to end flow

**Regression Testing:**

It is re-testing a program after doing the modification. It helps in ensuring that faults have not been introduced or uncovered as a result of the changes made, and that the modified system still meets its requirements. It is performed whenever the software or its environment is changed.

Test Strategy for Regression testing for the project explicit regression test cases will be prepared to carry out Regression Testing will be done once or twice during the release.

**API Testing**

It refers to the kind of testing that validates the business logic of APIs based on request and response. The goal of this testing type is ensure that system is reliable, secure, and performs well for the requested user input.

Test Strategy for API testing will be carried out every time when the API is modified and it will be done through unit tests and manually through Postman.

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Type of Testing** | **Approach of testing: Manual, Automation, Automation+ manual)** |
| Registration Page | Regression and Smoke Testing | Manual |
| Login Page | Regression and Smoke Testing | Manual |
| User Dashboard | Regression and Smoke Testing | Manual |
| Topics Page | Regression and Smoke Testing | Manual |
| Test Page | Regression and Smoke Testing | Manual |
| Reports/Statistics Page | Regression and Smoke Testing | Manual |
| APIs | API Testing | Automation(Using Unit Test) and Manual |

## 

## **5.4 Test Schedule**

|  |  |  |
| --- | --- | --- |
| **Build Version no** | **Start Date** | **End Date** |
| 1.0 | 15-02-2019 | 29-02-2019 |
| 1.1 | 01-03-2019 | 15-03-2019 |
| 1.2 | 16-03-2019 | 30-03-2019 |

# 

# **6. Defect Tracking**

***Defect Classification:***

***The following are the defect priorities defined according to their precedence:***

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Defect Priority** | **Description** |
| 1 | Critical | The critical priority bug represents that the system could not proceed further or will result in inappropriate behavior. Also, the bug needs to be resolved immediately. |
| 2 | Major | Bug with high priority states that the software may crash. Some functionalities work as expected, however. |
| 3 | Average | The bug has some undesirable outcomes, but most of the functional modules are working as expected. |
| 4 | Low | Low priority bug determines that there is no major harm caused to the system and the bug can be resolved after the other higher priority bugs are resolved. |

***The following are the defect severity defined according to their precedence:***

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Defect Severity** | **Description** |
| 1. | Causes Crash | Defect leads to crashing of the system. |
| 2. | Critical | System cannot work if the defect is not fixed. |
| 3. | Major | Defect has a large impact on the working of the system. |
| 4. | Minor | Defect does not largely affect the working of the system. |
| 5. | Enhancement | These are the features requested for improving the current system. |

# 

# **7. Assumptions**

* All bugs should receive immediate attention from the development team.
* All bugs found in a version of the software will be fixed and unit tested by the development team before the next build is released
* In case of lack of required equipment or changes in the feature requirements, the test schedules may need to be reviewed.

**8. Critical Dependencies and Constraints**

The following may impact the test cycle:

* Device availability
* Any new feature addition/modification to the application which is not communicated in advance.
* Any delay in the software delivery schedule including bug fixes. Any changes in the functional requirements since the requirements were signed-off.

# **9. Software Test Plan Maintenance**

STP shall be updated in any of the following or more circumstances:

* Slippage in schedule
* Change in the scope of work and other commitments