***QWALLITY APP***

**Test Plan Template**

**Release *<1>***

***<15.05.2024> - <20.06.2024>***

**VERSION HISTORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Version #** | **Prepared**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1 | *Anna Hovhannisyan* | *14.05.2024* | *Gohar Khachatryan* | *20.06.2024* | 01 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**TABLE OF CONTENTS**

**1** **Introduction 4**

1.1 Purpose of The Test Plan Document 4

**2** **Test ITEM 4**

2.1 Project description 4

2.2 Items to be Tested / Not Tested 4

2.3 Items to be excluded 4

2.4 Test Approach(s) 4

2.5 Test Deliverables 4

2.6 Staffing / Training Needs 5

**3** **Risk and mitigation 5**

3.1 Test Risks / Issues 5

**4** **Test Environment and infrastructure 5**

4 Required Infrastructure 5

**5** **Roles and responsibilities 5**

5.1 Roles and assigned responsibilities 5

5.2 Test Team Leader………………………………………………………………...7

5.3 Software Tester…………………………………………………………………..8

**6** **Test Schedule 8**

6.1 Milestones and schedule 9

* **Introduction**
* **Purpose of The Test Plan Document**

The purpose of this document is to communicate the testing approach that the QA team will use for the <Qwallity App 01> release. This document is targeted to the following reader groups:

**The QA Team**- This document will communicate internally the process used and the scope of the testing.

**The Development/Management Teams**- This document will provide a clear understanding of the testing approach to all external teams.

* **Test ITEM**
* **Project description**

*This test plan is for Qwallity web application․ Qwallity is an application that helps novice QA professionals learn, master the profession․ They can perform various exercises through which they gain experience.*

*It is the platform for courses, where the admin user can add/edit/delete the course, and the non-admin user can buy the courses. It consists of the following sections: Home, About us, Courses (Fundamental and Advanced), Exercises (Calculator, Black box, White box, Upload), Register, Login, My courses, Profile Page, User \_ Action, Weather.*

* **Items to be Tested / Not to be Tested**

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to Test** | **Test Description** | **Test Date** | **Estimation** |
| # 1 Registration functionality | User should register to the system with valid data |  | 5pt |
| # 2 Log In functionality | User sould successfully log in to the system |  | 1pt |
| # 3 Calculator functionality | User can calculat numbers with addition, subtraction, multiplication and division |  | 3pt |
| # 4 Black Box functionality | User can check to the system with valid data |  | 1pt |
| # 5 Add Courses functionality | Admin user can add course with **Fundamenatl** or **Advanced** coure types |  | 3pt |
| # 6 My courses functionality | When a user buys a course, they should see the purchased course in this section |  | 1pt |
| # 7 Courses edit/delet functionality | Admin user can edit/delet courses |  | 5pt |
| # 8 Courses field (**Fundamental/Advanced**) buy for nonadmin user | User can buy **Fundamental** or **Advanced** courses |  | 1pt |
| # 9 User Action functionality for non-admin user | The user can add amount to his account balance |  | 3pt |

* **Items to Not be tested**

|  |  |
| --- | --- |
| **Item Not to Test** | **Comment** |
| Weather | It is integrated API |
| Home Page | This section can be tested with additional time |
|  |  |

* **Test Approach(s)**

**Manual Testing Approach**

* Boundary Value Analysis
* Equivalence partitioning
* Decision Table Testing

We follow Agile Methodology in this project.

**Scope of GUI Testing** – Home page, About Us page

**Regular Bug Triages** – QA, Dev and BA are responsible for ensure that bugs are being prioritized on a regular basis.

**Testing Execution and Bug Tracking-** we will use Zephir scale and Jira to keep track of testing progress.

**Bug Severity and Priority Setting** – In bug reports these defined clearly in order to minimize ambiguity and ensure everyone has a common understanding.

**Description of the types of testing done and the testing period for this release** - Functionality Testing, API testing, Performance testing, Security testing

* **Test Deliverables**

|  |  |
| --- | --- |
| **Milestone/Project** | **Completion/Execution Dates** |
| Requirements Review/Estimation | <16/05/24> |
| *Test Case preparation* | <20/05/24> |
| *Test Case review* | <28/05/24> |
| Manual Testing | <09/06/24> |
| Automation script preparation |  |
| Automation code review |  |
| Regression testing(manual+automation) | <19/06/24> |

* **Staffing / Training Needs**
* *API training, DB skills*
* **Risk and mitigation**
* **Test Risks / Issues**

There is no security testing.

* **Test Environment and infrastructure**
* Required Infrastructure
* [**https://qwallity-dev.onrender.com/**](https://qwallity-dev.onrender.com/)
* [**https://qwallity-prod.onrender.com/**](https://qwallity-prod.onrender.com/)
* **Roles and responsibilities**
* **Roles and assigned responsibilities**

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Junior QA Engineer | * Create test cases * Executing test cases * Bug Reports |
| Senior QA Engineer | * Create test case scenarios * Chrck test cases written by junior QA enginner |
| QA Lead | * Collect the requirements and documentation * Written test plan * Clarifies requirements with client * Reporting about the progres and found bugs |
| Project Manager | * Manage testing process * Provide all the needed resources for the testing activities |

* **Test Team Leader/Manager**

**Team Management - Planning Tasks**

* Define detailed Test schedule for team.
* Provide initial test planning for the QA team.
* Define QA Team roles and responsibilities.
* Estimate effort for the various deliverables.
* Identify training requirements.
* Identify support requirements.
* Interview candidates to fulfill the various Software Tester roles.

**Team Management - Daily Tasks**

* Define QA tasks to be performed.
* Resolve management issues involving QA and the development team.
* Track ongoing QA preparation and execution tasks in a schedule tool.
* Manage the QA Team (motivation, assessment, and orientation of new members).

**Team Management - Weekly Tasks**

* Assign tasks to various team members.
* Attend applicable management meetings for the purpose of providing QA's approval of all change requests (when applicable).
* Chairs the team status meetings.

**Team Management – Ad hoc Tasks**

* Manage testing of software fixes during the Beta, Final and Regression phases of Testing.
* Identify potential testing roadblocks.
* Write performance reviews of testers.

**Team Management - Deliverables**

* Ensure quality, timeliness of the various testing deliverables as identified in this Strategy document.
* Provide comments as the internal testing reviewers for the development deliverables (Functional Specs, design docs, etc.).
* **Software Tester**

**Test Plan/Matrices and Scripts Preparation**

* Research relevant documentation to become knowledgeable enough to understand how the application was designed for the purpose of writing Test Plans/Matrices and Scripts.
* Write test plans that can be easily reproduced.
* Write test scripts that are easy to maintain.
* Ensure test plans and scripts are Traceable to applicable requirements and functional design documents (Functional Specs, help text, Design Documents, etc.).
* Write test cases (required set-up, procedures and information).
* Attend testing overviews (if available).

**Independent Verification of Test Specs/Matrices and Scripts**

* Provide Test Plans/Matrices and Scripts for review by peers, development and marketing representatives.
* Incorporate review comments into Test Plans/Matrices and Scripts.
* Conduct peer reviews Test Plans/Matrices and Scripts.

**Test Execution**

* Execute the Test Plans and Matrices.
* Run the automated tests.
* Report problems by raising bugs in Siebel.
* Follow up on bugs previously submitted in Siebel.

**Analyze Results**

* Report on successful test spec/matrix completion.
* Report on successful automated test completion.
* Verify successful resolution of bug fixes by verifying the contents of the bug reports and rerunning the test where applicable.
* Identify issues that should be documented in the Readme/Release Notes.
* **Test Schedule**
* **Milestones and schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **Deliverable** | **Effort(Person Hour)** | **Start Date** | **End Date** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |