*<qwallity web page>*

Test Plan

**Release *<1>***

***<15/05/2024> - <16/05/2024>***

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Version #** | **Prepared**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 1 | *Zaruhi Khachatryan* | *< 15/05/2024>* | *Gohar Khachatryan* | *<16/05/2024>* | 1 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

TABLE OF CONTENTS

[1 Introduction 4](#_Toc486519897)

[1.1 Purpose of The Test Plan Document 4](#_Toc486519898)

[2 Test ITEM 4](#_Toc486519899)

[2.1 Project description 4](#_Toc486519900)

[2.2 Items to be Tested / Not Tested 4](#_Toc486519901)

[2.3 Items to be excluded 4](#_Toc486519902)

[2.4 Test Approach(s) 4](#_Toc486519903)

[2.5 Test Deliverables 4](#_Toc486519906)

[2.6 Staffing / Training Needs 5](#_Toc486519908)

[3 Risk and mitigation 5](#_Toc486519909)

[3.1 Test Risks / Issues 5](#_Toc486519910)

[4 Test Environment and infrastructure 5](#_Toc486519911)

[4 Required Infrastructure 5](#_Toc486519912)

[5 Roles and responsibilities 5](#_Toc486519914)

[5.1 Roles and assigned responsibilities 5](#_Toc486519915)

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

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

[6 Test Schedule 8](#_Toc486519916)

[6.1 Milestones and schedule 9](#_Toc486519917)

# 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 <Product and Version> 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.

This document describes the methods and procedures that will be used by the Qwallity Manual testing team in the functional testing process of the web

# Test ITEM

## Project description

This test plan focuses on the QWALLITY web application, designed to help students specialize in quality assurance (QA). Through various sections of the application, students can engage in practical tasks and develop their professional skills. QWALLITY serves as a course platform where admin users can add, edit, or delete courses, while non-admin users can purchase and access these courses. The application includes the following sections: Home, About Us, Courses (Fundamental and Advanced), Exercises (Calculator, Black Box, White Box, Upload), Register, Login, My Courses, Profile Page, User Actions, and Weather.

## Items to be Tested / Not to be Tested

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to Test** | **Test Description** | **Test Date** | **Estimation** |
| Registration | To verify that the user registration functionality on the web page works correctly, ensuring new users can successfully create accounts and receive confirmation emails. This includes error messages, data storage | <05/15/24> | 3pt |
| Log in | To ensure the user login functionality on web page works correctly, allowing users to securely access their accounts. This includes error messages, authentication processes, security measures, and user experience. | <05/15/24> | 1pt |
| Save and edit the user profile | To ensure the save and edit user profile functionality on web page works correctly, allowing users to update and maintain their personal information securely and accurately. | <05/15/24> | 3pt |
| Account Balance | To ensure the account balance functionality on the web page works correctly, allowing users to view, and manage their account balances securely and accurately. | <05/15/24> | 3pt |
| Log in as admin | To ensure the admin login functionality on web page works correctly, allowing admin to securely access the admin dashboard. | <05/15/24> | 1pt |
| Edit info of the course as Admin | To ensure the admin editing functionality for course information on web page works correctly, allowing administrators to update and maintain course details securely and accurately. | <05/15/24> | 3pt |
|  |  |  |  |

## Items to Not be tested

|  |  |
| --- | --- |
| **Item Not to Test** | **Comment** |
| weather |  |
|  |  |
|  |  |

## Test Approach(s)

The testing strategy for the QWALLITY web application involves a combination of manual and automated testing approaches. This hybrid approach ensures comprehensive coverage of all functionalities, performance, and security aspects of the application.

* **Functional Testing:** Both manual and automated approaches will be used to verify that all features and functionalities work as intended.
* **Performance Testing:** Automated tools will be used to assess the application's performance under various load conditions.
* **Security Testing:** A combination of manual and automated methods will be employed to identify and address security vulnerabilities.

### Testing and Traceability

* **Systematic Testing Approach:**
  + A traceability matrix will be maintained to ensure that all requirements are covered by test cases.
* **Traceable Documents:**
  + Test cases will be linked to requirements and design documents to ensure complete coverage and traceability.

### Scope of GUI Testing

* **GUI Testing:**
  + All parts of the GUI will be tested for functionality, usability, and compliance with design specifications.

### Integration Testing

* **Integration Components:**
  + All integrated components of the application will be tested to ensure they work together seamlessly.
* **Challenges:**
  + Ensuring data consistency and proper communication between different modules.
  + Handling third-party integrations and ensuring they do not disrupt the application's functionality.

### Regular Bug Triages

* **Bug Prioritization:**
  + Regular bug triage meetings will be held to prioritize and address bugs.
* **Participants:**
  + QA Lead
  + Development Lead
  + Business Analyst (BA)
  + Documentation Writer

### Testing Execution and Bug Tracking

* **Tracking Progress:**
  + The testing team will use a bug tracking tool like Jira to log and track the progress of all testing activities.
  + Test execution reports will be generated regularly to monitor progress and identify any bottlenecks.

### Bug Severity and Priority Setting

* **Definitions:**
  + Severity and priority levels will be clearly defined in the test plan to ensure a common understanding among all team members.
  + Criteria for severity (e.g., Critical, High, Blocking, Low) and priority (e.g., High, Medium, Low) will be documented and communicated.

### Description of the Types of Testing and Testing Period

* **Types of Testing:**
  + Unit Testing: Conducted during the development phase by developers.
  + Integration Testing: Performed after unit testing to ensure different modules work together.
  + System Testing: Conducted by the QA team to validate the complete and integrated application.
  + Acceptance Testing: Performed by end-users to ensure the application meets their requirements.
  + Regression Testing: Conducted after any changes to ensure existing functionality is not affected.
* **Testing Period:**
  + Unit Testing: Ongoing throughout development.
  + Integration Testing: 2 weeks.
  + System Testing: 4 weeks.
  + Acceptance Testing: 1 week.
  + Regression Testing: Ongoing with each release cycle.

## Test Deliverables

This QA testing schedule is largely based on the development and technical publications schedules. All dates are subject to change if the development or documentation milestones are moved. Here are the key dates and testing periods.

| Milestone/Project | Completion/Execution Dates |
| --- | --- |
| Requirements Review/Estimation | <05/15/24> |
| *Test Case preparation* | <05/16/24> |
| *Test Case review* | <05/17/24> |
| Manual Testing | <05/18/24> |
| Automation script preparation | <05/19/24> |
| Automation code review | <05/120/24> |
| Regression testing(manual+automation) | <05/21/24> |

## 

## Staffing / Training Needs

Manual testing skills

Automated testing skills

# Risk and mitigation

## Test Risks / Issues

## **potental delays due to technical errors**

# Test Environment and infrastructure

## Required Infrastructure

Operating systems(Windows, macOS, Linux for desktops.)

Chrome, Firefox, Safari, Edge, and Internet Explorer (if required).

Jira

Zephyr

# Roles and responsibilities

## Roles and assigned responsibilities

[Describe various roles and responsibilities given to them. E.g. Junior Tester, Senior Tester, Project Manager etc.]

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Zaruhi Khachatryan | QA Engineer |
|  |  |
|  |  |

## Test Team Leader/Manager

The QA Team Leader/Manager is responsible for the following:

**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

The software tester reports to the QA Team Leader/Manager and is responsible for writing and executing manual and automated tests. The Software Tester's responsibilities include:

**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

[Describe the describe key milestones, deliverables, efforts, start date and end date]

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

# 