*<Qwallity App>*

Test Plan

**Release *<Version 1.1>***

***<01/05/2023> - <22/05/2023>***

VERSION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID & Version #** | **Prepared**  **By** | **Revision**  **Date** | **Approved**  **By** | **Approval**  **Date** | **Reason** |
| 001 | *Anahit Zakaryan* | *< 01/05/2023>* | *Maria Manukyan* | *<22/05/2023>* | Version 1.1 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

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 <QWALLITY Version 1.1.1> 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

QWALLITY designed as a platform for hosting and selling courses. Application have Registration, Login, Home (Add course, Course (Edit / Delete / Buy)) pages. Application have admin and non admin user roles. Admin user can add, edit, delete, and non admin user can buy courses.

**Registration screen** contains**:** **Name** – field up to 25 characters, should accept only letters. **Email** - format - xxx@domain, system should display following validation message - ‘Please use valid email format.’ in case of invalid format. It should be unique entry. **Username** - alphanumeric field up to 50 characters, should be unique entry, if entered duplicate username, system should display message – ‘Username already exists.’ **Password** - 8-14 alphanumeric characters. **Confirm Password** – the value should be the same as password. **Submit button** – After submit user should see following message “Your account has been successfully created.” **Note**: In case of invalid data no need to mention in red fields. All fields are required. After incorrect fields and submit all fields have validation messages.

**Login screen** contains**:** **Username and Password inputs, Login and Cancel buttons,** **Forgot your password link**, **Toggle password visibility icon.** After successful login user redirects to Home page with following message **“Welcome {registered name}”.** After login with not existing user or not valid pass, system shows following message **“Username or password is incorrect”.** Clicking on **Toggle**, you can see input password. System should block account after user 6 attempts. **Cancel** button should reset all entered data - both fields.

**Home screen** contains**:** **"You are now logged in"** popup with "X" icon. User should be able to close this popup. **Welcome** <Username>. **Add Course** Green button. User should be able to add courses from home screen. **Table of existing courses** (Table columns-Id, Title, Price, Date) Courses are ordered by Date.

**Add Course** screen should have 4 fields: **Title, Price, Type, Description** and **Submit** button (User can add course from Home screen). **Title -** System should allow alphabetical, up to 50 characters. **Price** - System should allow only numbers (int/float/decimal). **Type -** Drop down should contain 2 types: Fundamental and Advanced. **Description** - Any characters, up to 250. **Submit button** - Should be in orange color. After adding new course, user redirects to Home page, where all courses are visible. This users can see separated courses in Course page: Advanced and Fundamental. Selecting any of **Advanced** and **Fundamental**, user redirects to corresponding course section This users can see separated courses in Course page: **Advanced** and **Fundamental.. Author,** **Title** and **Buy** columns should be visible in screen. After clicking on Buy, user redirects to separate screen where can buy course. Screen contains the following fields: **Course title-price, course description, Buy button** (orange color).

**Non Admin -** After clicking on **Buy** button. **Account Balance is insufficient** (in case of not enough account balance). User redirects to **My Courses** screen (in case of enough account balance). User can see his/her bought courses.

**Admin user -** Buy button is not visible. Title is link. After clicking on Title, admin redirects to Edit/delete screen. After clicking on Edit button, user redirects to Edit Course page. Page contains **Title** and **Description** fields, **Submit** button. After submission, User get 'Your changes are done' message and redirects to Courses page. After clicking on Delete button, admin redirects to Courses page, "**Course is deleted**" message displays.

## Items to be Tested

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to Test** | **Test Description** | **Test Date** | **Estimation** |
| User Story # 1  Registration page | Check the compliance and functionality of Name, Email, Username, Password, Confirm Password inputs and Submit Button with requirements. | <06/05/23> | 3pt |
| User Story # 2  Login page | Check the compliance and functionality of Username and Password inputs, Login and Cancel buttons, Forgot your password link and Toggle password visibility icon with requirements. | <07/05/23> | 3pt |
| User Story # 3  Home page | Check the compliance and functionality of "You are now logged in" popup, Welcome <Username> Title, Add Course button and availability of courses table (Id, Title, Price, Date ) with requirements. | <09/05/23> | 2pt |
| User Story # 4  Add courses  (for admin) | Check the compliance and functionality Title, Price, Type, Description inputs and Submit button with requirements. | <11/05/23> | 2pt |
| User Story # 5  Delete/Edit courses (for admin) | Check the compliance and functionality of Courses page’s Advanced and Fundamental buttons, Author and Title link columns, Edit/delete buttons, Edit Course page’s Title, Description fields and Submit button, 'Your changes are done' and ‘Course is deleted’ messages with requirements. | <12/05/23> | 2pt |
| User Story # 6  Buy courses  (for non admin) | Check the compliance and functionality of Buy button visibility for admin/non admin, Courses page’s-Advanced and Fundamental buttons-Author, Title link and Buy button columns, ‘Account Balance is insufficient’ message or My Courses screen where are the bought course/s. | <13/05/23> | 1pt |
| Total estimation 1pt=5hour | | | 13pt |

## Items to Not be tested

## -

## Test Approach(s)

This section describes the methodology used by the QA team. Some examples of subsections are as follows:

**Automated Testing Approach** – What types of automated tests will be done on this product, what is the automated testing tool to be used, where will the automated tests be documented, etc.

**Testing and Traceability** – How will you ensure that you have implemented a systematic testing approach, what testing documents are traceable back to development documents, etc.

**Scope of GUI Testing** – What parts of the GUI will be tested, will it be automated, etc.

**Integration Testing** - Is there an integration component to the product in question, what are some of the integration challenges that need to be overcome.

**Regular Bug Triages** – Who is responsible for ensure that bugs are being prioritized on a regular basis, which QA, DEV, BA and Document Writer representatives are taking part in the regular bug reviews.

**Testing Execution and Bug Tracking** – How will the testing team keep track of their testing progress?

**Bug Severity and Priority Setting** – where are 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** - This shows what a standard testing process looks like.

## 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.  
QWALLITY APP testing will be manual and automation. Testing processes are include Integration, Performance, Security, Usability, Smoke, Sanity, Regression, System testing. Will be used Asana, TestRail, Jira, MySQL, JMeter, Postman and e.t.c apps and tools.

| Milestone/Project | Completion/Execution Dates |
| --- | --- |
| Requirements Review/Estimation | <01/05/2023> 1 days |
| *Test Case preparation* | <03/05/2023> 2 days |
| *Test Case review* | <05/05/2023> 2 days |
| Manual Testing | <12/05/2023> 8 days |
| Automation script preparation | <04/05/2023> 3 days (after Req. Review) |
| Automation code review | <05/05/2023> 1 day |
| Automation Testing | <15/05/2023> 3 days (after Manual Testing) |
| Regression testing (manual + automation) | ~<22/05/2023> 2 days (after BUG fix ~ 5 days ) |

## 

## Staffing / Training Needs

We need DB testing trainings.

# Risk and mitigation

## Test Risks / Issues

Include in this section any areas that may impede the progress of testing. Basically this section is an outstanding items list for QA.

# Test Environment and infrastructure

## Required Infrastructure

*https://qwallity-prod.onrender.com/*

# Roles and responsibilities

## Roles and assigned responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Anahit Zakaryan | Junior QA / Manual and partly SQL Testing |
| Maria Manukyan | Senior QA / Automation, API and SQL-DB testing |

## 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Milestone** | **Deliverable** | **Effort(Person Hour)** | **Start Date** | **End Date** |
| Registration page | Test cases and bugs | 15 | 06/05/2023 | 07/05/2023 |
| Login page | Test cases and bugs | 15 | 07/05/2023 | 09/05/2023 |
| Home page | Test cases and bugs | 10 | 09/05/2023 | 10/05/2023 |
| Add course (for admin user) | Test cases and bugs | 10 | 11/05/2023 | 12/05/2023 |
| Delete/Edit course (for admin user) | Test cases and bugs | 10 | 12/05/2023 | 13/05/2023 |
| Buy course (for non admin user) | Test cases and bugs | 5 | 13/05/2023 | 13/05/2023 |

# 