Mekelle University

EIT-M

DEPARTMENT: SOFTWARE ENGINEERING

COURSE NAME: SOFTWARE TESTING AND QUALITY ASSURANCE

ASSESEMENT TYPE: INDIVIDUAL

PREPARED BY: KAPITAL GEBREHIWET

ID NUMBER: EITM/UR178681/12

SUBMITTED TO: INSTRUCTOR MESELE

SUBMISSION DATE:08/05/2017

Test plan

Table of Contents

1. Introduction

The purpose of this test plan is to outline the strategy, resources, and schedule for testing the software system that incorporates sign-in and sign-up using Clerk, course addition and viewing, payment processing via Stripe, and data management with MongoDB.

2. Scope

This test plan covers functional, non-functional, performance, and security testing of the software system. The focus will be on:

*User authentication and authorization

*Course management

*Payment processing

*Database interactions

Features to be tested

*Course Management:

Adding new courses

Viewing available courses

Course details and descriptions

 *Payment Processing:

Payment workflow using Stripe

Handling payment errors and exceptions

*Database Interactions:

Storing and retrieving user data

Course data management

Features not to be tested

Third-Party Integrations:

Assumed to be working as per their individual specifications

Non-Critical UI Elements:

Elements that do not impact core functionality

3. Quality Objectives

3.1 Primary Objectives

*Ensure the system functions as intended without defects.

*Verify that the system meets user requirements and expectations.

3.2 Secondary Objectives

*Ensure the system is secure and data privacy is maintained.

*Validate system performance under expected load conditions.

4. Test Approach

A combination of manual and automated testing will be used to cover all aspects of the system.

4.1 Test Automation

Automation will be implemented for repetitive and regression test cases to enhance efficiency and reduce manual effort.

5. Roles and Responsibilities

Testers: Execute test cases, report defects, and verify fixes.

Developers: Resolve reported defects and support testers with technical issues.

6. Entry and Exit Criteria

6.1 Entry Criteria

*Development of all features is complete.

*Test environment is set up.

*Test data is prepared.

*All necessary test documentation is reviewed and approved.

6.2 Exit Criteria

*All planned test cases are executed.

*All critical and high-priority defects are resolved.

*Test reports are reviewed and approved by stakeholders.

7. Suspension Criteria and Resumption Requirements

7.1 Suspension Criteria

Testing will be suspended if:

*Critical defects block further testing.

*Test environment becomes unstable or unavailable.

7.2 Resumption Criteria

*Testing will resume once:

*Blocking defects are resolved.

*Test environment is restored and stable.

8. Test Strategy

8.1 QA Role in Test Process

QA will be involved from the requirements phase through to final delivery, ensuring quality at every stage.

8.2 Bug Life Cycle

Identification: Testers identify and report defects.

Triaging: Defects are reviewed, prioritized, and assigned.

Resolution: Developers fix the defects.

Verification: Testers verify the fixes.

Closure: Verified defects are closed.

8.3 Testing Types

Unit Testing: Verify individual components or units of code for correctness.

Integration Testing: Test the interaction between integrated units or components.

System Testing: Validate the complete and integrated software product to ensure compliance with requirements.

Acceptance Testing: Confirm the system meets the acceptance criteria and is ready for deployment.

Functional Testing: Verify that each function of the software operates in conformance with the requirement specification.

Non-Functional Testing: Assess system performance, usability, and reliability.

Performance Testing: Measure system response times and stability under load.

Security Testing: Ensure the system is secure from vulnerabilities and threats.

8.4 Bug Severity and Priority Definition

Severity List:

Critical: System crash, data loss, or severe security issue.

High: Major functionality impacted but with a workaround.

Medium: Minor functionality impacted.

Low: Cosmetic issues or minor inconsistencies.

Priority List:

High: Must be fixed immediately.

Medium: Should be fixed in the next release cycle.

Low: Fix when time permits.

9. Resource and Environment Needs

9.1 Testing Tools

*Test Management Tool: Jira

*Automation Tool: Cypress

*Bug Tracking Tool: Jira

9.2 Configuration Management

Configuration management tools will be used to maintain consistency in the test environment and manage version control.

9.3 Test Environment

A dedicated test environment that mirrors the production environment will be used for testing.

10. Test Schedule

Planning Phase: 1 day

Test Design Phase: 2 days

Test Execution Phase: 1 day

Defect Resolution Phase: 2 days

Final Review and Sign-off: 1 day

11. Test Deliverables

Test Plan: Document outlining the testing strategy and scope.

Test Cases: Detailed test cases for each functionality.

Test Scripts: Automated test scripts for regression testing.

Test Data: Data sets required for executing test cases.

Test Reports: Summaries of test execution and results.

Defect Reports: Detailed reports on identified defects.

Final Test Summary Report: Comprehensive summary of all testing activities and outcomes.

12 Approvals:

- **The test should be approved by the testers**

13 Terms/Acronyms

QA: Quality Assurance

UAT: User Acceptance Testing