**EGERTON**  **UNIVERSITY**

**ONLINE LIBRARY SYSTEM PROJECT**

**TEST PLAN**

**S13/02367/19**

**MOURICE ADERO**

**Table of content**

Table of Contents

[**Chapter 1** 1](#_Toc130939064)

[**Introduction** 1](#_Toc130939065)

[**Goals/Objectives** 1](#_Toc130939066)

[**Scope** 2](#_Toc130939067)

[**Chapter 2** 3](#_Toc130939068)

[**Test Plan** 3](#_Toc130939069)

[**Interfaces** 4](#_Toc130939070)

[**Chapter 3** 5](#_Toc130939071)

[**Test Procedures** 5](#_Toc130939072)

[**Test Criteria** 5](#_Toc130939073)

[**Unit Test** 6](#_Toc130939074)

[**Integration Test** 8](#_Toc130939075)

[**System Test** 9](#_Toc130939076)

[**Chapter 4** 10](#_Toc130939077)

[**Testing Resource Planning** 10](#_Toc130939078)

[**Human** 10](#_Toc130939079)

[**System** 10](#_Toc130939080)

[**Chapter 5** 11](#_Toc130939081)

[**Test Environment** 11](#_Toc130939082)

[**Chapter 6** 12](#_Toc130939083)

[**Test Schedule and Estimation** 12](#_Toc130939084)

# **Chapter 1**

## **Introduction**

### **Goals/Objectives**

The goals and objectives of this test plan are to ensure that the online library system is tested thoroughly, to verify that it meets the functional and non-functional requirements specified in the software requirements specification, and to identify and report any defects or issues found during testing. The test plan aims to achieve the following goals and objectives:

1. To verify the functional requirements of the online library system, including but not limited to:

* User authentication and authorization
* Search and retrieval of books and other resources
* Requests for items
* User profile management
* Reporting and analytics

1. To verify the non-functional requirements of the online library system, including but not limited to:

* Performance and scalability
* Usability and accessibility
* Security and privacy
* Compatibility and interoperability
* Reliability and availability

1. To ensure that the online library system interfaces with other systems and components correctly and efficiently, including but not limited to:

* The database management system
* The web server and browser
* The network infrastructure
* The hardware and software platforms

1. To apply appropriate testing techniques and methods, such as unit testing, integration testing, system testing and acceptance testing, to ensure that the online library system is tested comprehensively and effectively.

By achieving these goals and objectives, the online library system can be tested thoroughly and accurately, and any defects or issues found during testing can be addressed and resolved before the system is deployed to production.

### **Scope**

The scope of the testing process includes:

1. Functional Testing: This type of testing focuses on verifying whether the system meets its intended functions as specified in the requirements document. It covers testing of features such as search and filter functionality, user registration, login and logout.
2. Performance Testing: This type of testing measures the system's performance with regard to responsiveness, throughput, and scalability. It covers testing of system response times, concurrent user handling, and database performance.
3. Security Testing: This type of testing verifies that the system is secure and protected against unauthorized access, theft, and data manipulation. It covers testing of login security, data encryption, and data access controls.
4. Compatibility Testing: This type of testing verifies that the system works correctly on different browsers, platforms, and operating systems.
5. Usability Testing: This type of testing measures how easy the system is to use and understand. It covers testing of the user interface, navigation, and user experience.

The scope of the testing process also includes identifying the risks involved in testing, determining the test environment, defining the testing objectives, and outlining the expected outcomes of the testing process.

# **Chapter 2**

## **Test Plan**

The Test Strategy outlines the approach that are used to perform testing on the Online Library System. It describes the types of tests that are conducted, the methods that are used to execute the tests, and the tools that are used to manage and track the testing process.

Test strategy

**Types of tests:** The testing approach for the Online Library System involves three main types of tests: Unit Testing, Integration Testing, and System Testing.

* Unit Testing: This type of testing is performed on individual software components to ensure that each component works correctly in isolation. This include the user signup and login modules, requesting items modules and item management modules.
* Integration Testing: This type of testing is performed to ensure that the different software components work together correctly. This includes navigations, search feature that gets different items and other combined features like item view page and downloading.
* System Testing: This type of testing is performed to validate that the entire system functions correctly as a whole. The system is tested from the point the user gets to the landing page, all through to account creation, profile management and all the other roles including administrators’ roles.

Methods of testing: The testing approach involve both manual and automated testing. Manual testing is used for certain types of testing, such as exploratory testing and usability testing. Automated testing is used for repetitive testing, such as regression testing.

Test data management: Test data management is used to ensure that the test environment is set up correctly and that the test data used in testing is accurate and relevant.

Test coverage: Test coverage is measured to ensure that all requirements are adequately tested. Test coverage metrics is used to determine the percentage of requirements that have been tested and the percentage of defects found during testing.

### **Interfaces**

The interface section describes the interfaces that is tested as part of the testing effort. These interfaces include:

* User Interfaces: This includes all the web pages, forms, and other UI components that users interact with when using the online library system. Testing of the user interface ensures that it is easy to use, user-friendly, and consistent across all pages.
* Application Interfaces: These are the interfaces between different modules of the application. Testing of these interfaces ensure that data is passed correctly between different modules of the system. Modules like search and filter that directly manipulates data are tested to ensure relevant results are received by the users.
* Database Interfaces: These are the interfaces between the application and the database. Testing of these interfaces ensure that data is correctly inserted, updated, and deleted from the database. This makes use of SQL queries that are executed through various related modules.

# **Chapter 3**

## **Test Procedures**

Test procedures refer to a set of detailed instructions and steps that need to be followed in order to execute specific tests for the system. These test procedures outline the specific actions that need to be taken to verify that the system meets the desired functionality, requirements and specifications.

### **Test Criteria**

Test criteria are derived from the requirements specification and are measurable, objective, and verifiable.

* Functional requirements: The test criteria for functional requirements is based on the expected behaviour of the system. For example, a requirement specifies that the system should allow users to search for books by author, the test criteria verifies that the search function works correctly and returns accurate results.
* Performance requirements: Performance requirements define the expected response time, throughput, and resource utilization of the system. The test criteria verify that the system meets these requirements under expected and peak load conditions.
* Security requirements: Security requirements specify the measures that the system should take to protect user data and prevent unauthorized access. The test criteria verify that the system enforces security measures and that it is not vulnerable to common security threats.
* Usability requirements: Usability requirements define the ease of use of the system, including navigation, interface design, and user assistance. The test criteria verify that the system is intuitive, user-friendly, and meets the needs of the target users.
* Compatibility requirements: Compatibility requirements define the system's ability to work with other software, hardware, and network environments. The test criteria verify that the system works correctly with the specified hardware and software.
* Regulatory requirements: Regulatory requirements define the legal and industry standards that the system must comply with. The test criteria verify that the system meets these requirements and is certified to operate in the intended jurisdiction. Integration for third party API like Google Books API is limited in the system as to the requirements of Google such that any item searched gives only the results that are allowed to be shown by third party.

### **Unit Test**

Unit testing is a type of software testing where individual units or components of the system are tested in isolation from the rest of the system. The purpose of unit testing is to validate that each unit/component of the system is working as expected and to identify and fix any defects early in the development cycle.

**Test Cases:** A set of test cases is developed to test each unit/component. Test cases is designed to test the functionality of each unit/component and should cover all possible use cases.

**Examples**

Test Case 1: User Registration

Test Steps:

* Navigate to the registration page.
* Enter valid user information in all the required fields.
* Click on the "Submit" button.
* Verify that the user is redirected to the login page.
* Verify that the user account is created successfully by checking the database.

Expected Results:

* The registration page should load successfully.
* All required fields should accept valid user information.
* The user is able to submit the form successfully.
* The user is redirected to the login page.
* The user's account is created successfully and stored in the database.

Test Case 2: User Login

Test Steps:

* Navigate to the login page.
* Enter the correct username and password.
* Click on the "Login" button.
* Verify that the user is redirected to the homepage.
* Verify that the user's login details are stored in the database.

Expected Results:

* The login page should load successfully.
* The user should enter the correct login credentials.
* The user is able to log in successfully.
* The user is redirected to the homepage.
* The user's login details are stored in the database.

Test Case 3: Search Book

Test Steps:

* Navigate to the search page.
* Enter a valid book title or author name in the search field.
* Click on the "Search" button.
* Verify that the search results are displayed correctly.
* Click on the book title to view the book details.
* Verify that the book details are displayed correctly.

Expected Results:

* The search page should load successfully.
* The user should enter a valid book title or author name.
* The search should return correct results.
* The search results are displayed correctly.
* The user is able to view the book details.
* The book details are displayed correctly.

**Test Data:** Test data is created to test the unit/component. The test data includes both valid and invalid data to test the robustness of the unit/component.

**Example**

Valid data:

First Name: John

Last Name: Doe

Email: [johndoe@email.com](mailto:johndoe@email.com)

Password: abcDEF123!

Invalid data:

First Name: 123

Last Name:!$%

Email: johndoecom

Password: abcd (less than 8 characters)

**Test Environment:** A test environment is set up to conduct the unit testing. The environment is similar to the production environment to ensure that the test results are accurate. This is achieved by use of local server (XAMPP) and web browser.

**Test Execution:** The unit tests are executed using the test cases and test data in the test environment.

**Defect Resolution:** Any defects identified during the unit testing is resolved. The defects are logged in a defect tracking tool and prioritized based on severity.

### **Integration Test**

Integration testing focuses on verifying that individual software modules work together correctly as a larger integrated system. The main objective of integration testing is to uncover defects in the interaction between different software modules and ensure that the system functions as expected.

* Integration Test Objective: This section provides an overview of the integration testing objectives, including the software components to be tested, the type of integration testing to be conducted, and the expected results.
* Integration Test Approach: This section describes the approach that is taken to carry out integration testing. The approach includes a detailed description of the sequence in which the software components is integrated, and the criteria for determining when to move from one integration phase to the next.
* Integration Test Risks: This section identifies potential risks associated with integration testing, such as conflicts between different software components or unforeseen compatibility issues and steps to mitigate the issues.

### **System Test**

System Test: The System Test verifies the overall functionality, usability, reliability, and performance of the Online Library System. The following tests is performed during the System Test:

* Functionality Test: This test verifies that all the functions and features of the system work as intended. The test covers all the use cases and scenarios defined in the Use Case document.
* Usability Test: This test evaluates the ease of use and user-friendliness of the system. The test involves a group of representative end-users who perform typical tasks using the system. The goal is to identify any issues related to the user interface, navigation, or any other usability concerns.
* Reliability Test: This test evaluates the system's ability to perform consistently and reliably over time. The test involves running the system continuously for an extended period to identify any potential stability or reliability issues.
* Performance Test: This test evaluates the system's ability to handle the expected workload and traffic. The test involves simulating the expected user traffic and monitoring the system's response time, resource utilization, and throughput.
* Security Test: This test evaluates the system's ability to maintain the confidentiality, integrity, and availability of the data. The test involves attempting to bypass security measures, perform unauthorized actions, and exploit vulnerabilities in the system.

# **Chapter 4**

## **Testing Resource Planning**

### **Human**

Human resources refer to the individuals involved in the testing process.

* Roles and responsibilities: The tester’s role is clearly defined. This help ensure that all aspects of testing are covered, and that there is no duplication of effort or gaps in the testing process.
* Skills and expertise: Testing require someone skilled in the area of application for instance, PHP and bootstrap 5.
* Availability: Testing is done throughout the development of the system thus the tester is present during the entire time.

### **System**

System testing requires a range of hardware and software resources to ensure that the system functions as intended. The following resources is required for system testing:

* Hardware: This includes laptop computers, mobile devices, and network infrastructure such as routers and switches.
* Software: This includes operating systems, databases, web servers, and other middleware.
* Test Data: The system testing process require a large volume of test data to simulate different scenarios and ensure that the system can handle different types of inputs.

All of these resources need to be in place before the system testing process begins to ensure that the testing team can execute the test cases effectively and efficiently.

# **Chapter 5**

## **Test Environment**

This includes hardware, software, tools, and other resources necessary to execute tests effectively and efficiently.

Hardware Requirements: This outlines the hardware resources needed for testing the online library system. The hardware requirements are an active access point for internet connection, a computer such as laptop or desktop and a mobile phone.

Software Requirements: This outlines the software resources required for testing the online library system. The system works on every operating system. The other software required is only a web browser. The application server can be hosted on a remote location that is configured using Linux operating system.

Test Data: This include lists of the data sources, volumes, and types of data needed for various testing scenarios. Data required such as valid emails, pdf files for items and images of different types used to populate the items in the database.

Test Environment Setup: The test environment is set up using a local server of XAMPP that uses Apache server and MySQL database server. It also includes several testing browsers like Google Chrome, Ms Edge and Firefox.

Test Environment Risks and Mitigation: Test environment is set up through a local git repository that has a backup on GitHub. This allows for references, backup and recovery of data and system files.

# **Chapter 6**

## **Test Schedule and Estimation**

**Testing Phases:** The testing phases included in this Test Plan are Unit Testing, Integration Testing, and System Testing. Each testing phase have a specific timeline and estimated completion date.

**Test Schedule:** The following table outlines the estimated start and end dates for each testing phase:

| **Testing Phase** | **Estimated Start Date** | **Estimated End Date** |
| --- | --- | --- |
| Unit Testing | 10/02/2023 | 15/02/2023 |
| Integration Testing | 20/02/2023 | 26/02/2023 |
| System Testing | 01/03/2023 | 10/02/2023 |

**Testing Effort:** The following table outlines the estimated testing effort required for each testing phase:

| **Testing Phase** | **Estimated Effort** |
| --- | --- |
| Unit Testing | 5 days |
| Integration Testing | 5 days |
| System Testing | 10 days |

**Risks and Contingency Plans:** The following risks have been identified:

* Defects found during testing that require significant rework
* Limited availability of testing resources

Contingency plans have been developed to address each risk:

* Prioritize defect fixing and adjust testing timelines accordingly
* Re-allocate resources from other projects or departments as needed to ensure testing is completed on time