**Final Assignment: Test Plan**

**SD6503 Testing and Secure Coding**

Dave Sushames, Sachith Samarasinghe, Shontelle Steeg

Table of Contents

[Test Plan (draft) 3](#_Toc209348927)

[1. Test Plan Identifier 3](#_Toc209348928)

[2. Introduction 3](#_Toc209348929)

[3. Test Items 3](#_Toc209348930)

[4. Features to be tested 4](#_Toc209348931)

[5. Features not to be tested 4](#_Toc209348932)

[6. Approach 5](#_Toc209348933)

[7. Item Pass/Fail Criteria 5](#_Toc209348934)

[7.1 Items Pass/Fail Criteria 5](#_Toc209348935)

[7.2 Entry Criteria 6](#_Toc209348936)

[7.3 Exit Criteria 7](#_Toc209348937)

[8. Suspension and Resumption Requirements 7](#_Toc209348938)

[9. Test Deliverables 7](#_Toc209348939)

[9.1 Before Testing Phase 7](#_Toc209348940)

[9.2 During Testing 7](#_Toc209348941)

[9.3 After Testing 7](#_Toc209348942)

[10. Testing Tasks 7](#_Toc209348943)

[11. Environmental Needs 8](#_Toc209348944)

[12. Responsibilities 8](#_Toc209348945)

[13. Staffing and Training needs 9](#_Toc209348946)

[13. Schedule 9](#_Toc209348947)

[14. Risk and Contingencies 10](#_Toc209348948)

[16. Approvals 10](#_Toc209348949)

[Test Summary Report (Placeholder) 11](#_Toc209348950)

# Test Plan (draft)

## Test Plan Identifier

The Test Plain Identifier for this Plan is FullMetalLibrary\_Testing\_v1

## Introduction

This Test Plan outlines the testing activities that will be undertaken for the FullMetalLibrary ASP.NET Core 8.0 MVC web application, that has been developed with EF Core and SQL Server.

The tasks to be completed include validation of CRUD functionality for three tables created Books, Authors, Admin, session-based authentication with login, logout, registration, and the added advanced functions search, sort, and filter, as required in the final assignment descriptor.

This Test Plan follows IEEE 829 format alongside the assignment descriptor. The scope of the testing includes black box, white box, unit, automation with Selenium, static code analysis, and load testing with JMeter.

## Test Items

The following are the test items with their descriptions are the functional requirements for the FullMetalLibrary application.

Table 1: Test Items with description and requirement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Item ID** | **Test Items** | **Roles** | **Description** | **Requirement** |
| **TI-01** | Login and Logout Feature | Admin | Accuracy of the login and logout feature will be tested. | The admin should be able to login with the correct credentials, deny if the details are invalid, and logout clears the current session. |
| **TI-02** | Register Feature | Admin | Accuracy of the register feature will be tested. | The admin should be able to register a user, creating a new admin row, a strong password, and reject duplicate addressing. |
| **TI-03** | Admin Management | Admin | Accuracy of the admin CRUD will be tested. | The admin should be about to create, view, edit, activate/deactivate, and delete accounts. |
| **TI-04** | Author Management | Admin | Accuracy of the author CRUD will be tested. | The admin should be able to create, view, edit, delete authors with validation. |
| **TI-05** | Book Management | Admin | Accuracy of the book CRUD will be tested. | The admin should be able to create, view, edit, and delete books with validation. |
| **TI-06** | Search Feature | Admin | Accuracy of the search feature will be tested. | The admin should be able to search books by title/genre and authors by first/last name. |
| **TI-07** | Sort Feature | Admin | Accuracy of the sort feature will be tested. | The admin should be able to sort books and authors by A-Z and Z-A for titles and author names. |
| **TI-08** | Navigation Feature | Admin | Accuracy of the navigation feature will be tested. | The admin should be able to navigate links for Home, Books, Authors, Admins, and that the login/Register vs logout works with session. |
| **TI-09** | Validation and messages feature | Admin | Accuracy of the validation will be tested. | When the admin enters invalid or missing input, email format, publish date, names, and genres, it should trigger validation errors and display messages. |

## Features to be tested

* The CRUD for the three tables Admin, Author, Book
* Login, Logout, Register session-based authentication.
* Search in Books and Author.
* Sort A-Z and Z-A for Books and Authors.
* Validation for password strength, email, publish date, names, and by genre.
* Navigation layout behavior based on the session.

## Features not to be tested

* Mobile responsiveness
* ASP.NET Core Identity advanced features such as email confirmation and resetting passwords.
* Addition of third-part integrations.

## Approach

* Unit Testing
* Black-Box Testing
* White-Box Testing
* Automation Testing with Selenium
* Static Code Analysis
* Load Testing with JMeter

## Item Pass/Fail Criteria

The Item Pass/Fail Criteria outlines the decision rules to determine whether a test item has passed or failed.

### Items Pass/Fail Criteria

The below table shows the Pass/Fail criteria related for each feature for the FullMetalLibrary application.

Table 2: Test Item Pass/Fail Criteria.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Item ID** | **Test Condition** | **Pass Criteria** | **Fail Criteria** |
| **TI-01** | Accuracy of the login and logout feature. | The admin should be able to login with the correct credentials, denied if the details are invalid, and logout clears the current session. | The admin can bypass the login, accessing with no session, or when logging out there is a failure to clear the session. |
| **TI-02** | Accuracy of the register feature will be tested. | The admin should be able to register a new admin with a strong hash password, and reject duplicate addressing. | The admin can store a plain text password, weak password can be accepted, and a duplicate email can be accepted, or failure to register with valid credentials. |
| **TI-03** | Accuracy of the admin CRUD. | The authorized admin should be about to create, view, edit, activate/deactivate, and delete accounts. | The admin can access without authorization or session, can create, view, edit, activate/deactivate, and delete accounts with incorrect data, or fail with valid data. |
| **TI-04** | Accuracy of the author CRUD. | The admin should be able to create, view, edit, delete authors with validation, and list updates. | The admin can input invalid data accepted, CRUD fails to persist, and the list does not update. |
| **TI-05** | Accuracy of the book CRUD. | The admin should be able to create, view, edit, and delete books with validation. Future dates are rejected, and the list updates. | The admin can enter future date data; an invalid genre is accepted, or there are persistence errors. |
| **TI-06** | Accuracy of the search feature. | The admin should be able to search books by title/genre and authors by first/last name. Correct matches are returned; no results show empty without error. | The admin using the search feature but it returns irrelevant results or errors for valid queries. |
| **TI-07** | Accuracy of the sort feature. | The admin should be able to sort books and authors by A-Z and Z-A for titles and author names. Ordered matches are returned. | The admin sorts but it incorrectly orders or returns inconsistent results. |
| **TI-08** | Accuracy of the navigation feature. | The admin should be able to navigate links for Home, Books, Authors, Admins, and that the login/Register vs logout works with session. | The admin navigates the site but the links are broken or shows incorrect visibility or the pages. |
| **TI-09** | Accuracy of the validation. | When the admin enters invalid or missing input, email format, publish date, names, and genres, it should trigger validation errors and display messages. | The admin receives missing or incorrect messages, it accepts invalid input, or valid inputs are rejected. |

### Entry Criteria

The following conditions must be met before testing on the FullMetalLibrary web application can begin:

* The testing environment needs are met and accessible.
* The FullMetalLibrary code build completes without any errors.
* Database migrations for the FullMetalLibrary have been applied and complied.
* The test case documentation is available.
* The test cases are designed before test execution begins.

### Exit Criteria

The following conditions must be met before testing on the FullMetalLibrary web application can be closed:

* All major defects must be fixed, once achieved, retesting must be completed.
* No major defects are left unresolved.
* All planned test cases have been executed.

## Suspension and Resumption Requirements

Testing shall be suspended if the database connection fails or if the migrations break schema, preventing CRUD operations. The testing will resume after the issues with database integrity has been restored.

Testing shall be suspended if testing in at least 50% of test cases for a single test condition fail during its execution of the test cases, further testing must be suspended. The resumption of the testing will be dependent on defects being fixed before continuation.

## Test Deliverables

The documents required before, during, and after testing has been conducted for the FullMetalLibrary web application.

### Before Testing Phase

* Test Plan Document
* Test data

### During Testing

* Test case spreadsheet
* Unit test results
* Bug report
* Automation scripts and screenshots

### 9.3 After Testing

* Static code analysis report
* JMeter results
* Test summary report

## 10. Testing Tasks

For testing to be completed on the FullMetalLibrary web application the following testing tasks need to be executed:

* Develop and execute unit tests for all models and methods.
* Black box and white box test cases completed.
* Selenium automation scripts.
* Run static code analysis and log issues.
* Run JMeter and document load tests.
* Write bug report.
* Generate test summary report.

## Environmental Needs

To carry out the testing requirements, the below environmental needs must be met.

Table 3: Environmental Needs for testing.

|  |  |  |
| --- | --- | --- |
| **No.** | **Resources** | **Description** |
| **1** | Computer | A computer is required to access resources involving FullMetalLibrary web application. |
| **2** | Network | Internet access is required to test FullMetalLibrary web application. |
| **3** | Visual Studio 2022 (ASP.NET Core 8.0) | Correct version of Visual Studio to access and test FullMetalLibrary web application. |
| **4** | SQL Server. | Local test database to access and test FullMetalLibrary web application. |
| **5** | Microsoft Tools | Testers need access to Microsoft for test cases and documentation. |
| **6** | GitHub | Team members will need to use GitHub for the repository and migration. |
| **7** | Selenium WebDriver | To complete automation testing, the team members must use Selenium. |
| **8** | Browser | Google Chrome is needed for testing FullMetalLibrary web application. |
| **9** | JMeter | To complete load testing, team members must use JMeter. |

## Responsibilities

For testing of the FullMetalLibrary web application the below responsibilities need to be assigned for the Test Manager and Test Team.

Table 4: Roles and Responsibilities for testing.

|  |  |  |
| --- | --- | --- |
| **No.** | **Resources** | **Descriptions** |
| **1** | Test Manager  (Dave Sushames) | * Assign tasks * Track progress * Approval * Coordinates schedule |
| **2** | Test Team  (Sachith Samarasinghe, Dave Sushames, Shontelle Steeg) | * Unit Testing * Black box testing * White box testing * Static code analysis * Selenium automation testing * Load testing with JMeter * Documentation |

## 13. Staffing and Training needs

The following has been identified as the necessary staff and training needs required for the FullMetalLibrary web application:

* Team members are to be familiar with ASP.NET MVC and Entity Framework that has been introduced in SD6503 Testing and Secure Coding.
* Guidance may be needed for Git workflow.
* Training for static analysis interpretation.
* Training will be required for JMeter load testing.
* Training for automation testing using Selenium may be required.

## Schedule

To complete testing for the FullMetalLibrary web application, the following schedule has been created.

Table 5: Schedule for testing.

|  |  |  |
| --- | --- | --- |
| **Task** | **Team Members** | **Required Time** |
| Prepare the Test Plan | Team | 4-5 hours |
| Unit Testing | Team | 6-8 hours |
| Black-Box Testing | Team | 4-6 hours |
| White-Box Testing | Team | 4-6 hours |
| Automation Testing (Selenium) | Team | 4-5 hours |
| Static Analysis | Team | 4-6 hours |
| Load Testing (JMeter) | Team | 2-3 hours |
| Bug report | Team | 2-3 hours |
| Test Summary Report | Team | 3-4 hours |
| **Total** |  | **34-47 hours** |

## Risk and Contingencies

Testing the FullMetalLibrary web application, the following risks and mitigations need to be outlined:

Table 6: Risks and Mitigations for testing.

|  |  |
| --- | --- |
| **Risk and Contingencies** | **Mitigation** |
| Database migrations conflicts may arise. | To mitigate this, reset the local database, and then reapply the migrations. |
| Selenium test may fail due to the browser version being incorrect. | To mitigate this, update the WebDriver before execution. |
| Schedule overrun, testing all features not complete. | To mitigate this, prioritize CRUD and login features that are required in the assignment descriptor. |

## 16. Approvals

Table 7: Approvals for Test Plan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Position** | **Date** | **Signature** |
| Richa Panjabi | Product Owner |  |  |
| Dave Sushames | Program Manager |  |  |
| Sachith Samarasinghe | Team Member |  |  |
| Shontelle Steeg | Team Member |  |  |

# Test Summary Report (Placeholder)