***SOFTWARE TEST SPECIFICATION***

***Student Course Enrollment System***

**Project Manager:**

**Design Lead:**

**Technical Lead:**

**1.0 Introduction**

The Student Course Enrollment System is a web-based application designed to simplify and streamline the course enrollment process for students and administrators in educational institutions. The system provides a centralized platform where students can view and enroll in available classes while administrators can manage class details and monitor student enrollments. The goal is to eliminate manual processes, reduce administrative workload, and provide students with a user-friendly interface to manage their course selections efficiently.

**1.1 Goals and Objectives**

The primary goals and objectives of the Student Course Enrollment System are:

* **Facilitate Enrollment:** To provide students with a streamlined process for viewing, enrolling, and tracking their classes.
* **Simplify Administration:** To enable administrators to create, update, and manage classes with predefined limits and schedules.
* **Ensure Compliance:** To ensure that students do not exceed the maximum limit of active classes, maintaining academic workload balance.
* **Centralized Management:** To provide a centralized platform where both students and administrators can view and manage enrollment statuses and class progress.
* **Improve Efficiency:** To reduce time and effort spent on manual enrollment processes and reduce human errors.

**1.1.2 Specific Objectives**

Based upon the use cases mentioned in the requirements, we have identified the testing objective for each scenario. Under each User cases we will test the specific test cases.

1. **Student Login (UC-001)**
   * Valid Login Attempt
2. Invalid Password
3. Account Lock After Multiple Failed Attempts
4. Empty Username or Password Field
5. Session Management

**2. View Available Classes (UC-002)**

1. View All Classes
2. Class Search by Keyword
3. Filter Classes by Category
4. No Available Classes

**3. Enroll in a Class (UC-003)**

1. Successful Enrollment
2. Exceeding Class Limit
3. Class Full
4. Duplicate Enrollment Attempt
5. **Create a New Class (UC-004)**
6. Create Class with Unique Name and Valid Data
7. Duplicate Class Name Error
8. Missing Field Data
9. Invalid Enrollment Limit
10. **Manage Class Enrollment (UC-005)**
11. View Enrolled Students
12. Mark Student as Complete
13. Exceeding Enrollment Capacity
14. View and Update Student Progress
15. **Additional System-Level Scenarios**
16. Concurrent Logins
17. Data Security and Encryption
18. Session Expiry After Inactivity
19. Access Permissions
20. Data Consistency on Class or Student Deletion

**1.1.3 Testing Focus and Objective**

The testing focus for the **Student Course Enrollment System** centers on verifying critical functionalities, ensuring data integrity, and maintaining security and usability across both student and admin roles. The primary objective is to validate that students can seamlessly log in, view, and enroll in available classes while adhering to enrollment limits, and that admins can efficiently manage class creation, enrollment capacities, and student progress. Key testing efforts will address common user flows, such as successful and failed login attempts, class search and filtering, enrollment actions, and session management. Security is also a major focus, with tests designed to confirm data encryption, enforce access permissions, and protect against unauthorized access. Additionally, the system’s stability under concurrent usage will be tested to ensure reliable performance, especially during peak enrollment periods. Overall, the testing aims to provide a smooth, secure, and functional user experience aligned with project requirements and institutional standards.

**2.0 Test Plan**

This section serves as a comprehensive guide outlining the testing strategy and crucial project management considerations essential for the successful execution of tests.

**2.1 Software to be Tested**

In this sub-section, the specific software components earmarked for testing are clearly identified. The primary focus is on the **Student Course Enrollment System** , encompassing key functionalities associated with user roles (students and admins), course management and progress tracking. The purpose is to narrow down the testing scope, ensuring a targeted and efficient approach to validation.

1. **Emphasis on User Roles:** The testing plan recognizes the importance of verifying the distinct experiences of both students and admins within **Student Course Enrollment System** . This includes assessing the accessibility, functionality, and security features tailored to each user role.
2. **Key Functionalities**: The identification of core functionalities, such as course management and progress tracking, ensures that testing efforts are concentrated on areas crucial to the platform's success. This targeted approach allows for a thorough examination of the features that contribute significantly to the user experience.

**2.2 Testing Tools and Environment**

This sub-section details the testing environment and tools essential for executing the testing strategy effectively.

**Test Environment:** The test environment encompasses a variety of web browsers, including Chrome, Firefox, and Safari. This diversity ensures that the E-Learning Module is compatible and functional across different platforms, replicating the real-world scenarios that users may encounter.

**Testing Tools (Manual):** As manual testing does not involve automated testing tools, the focus shifts to manual testing techniques, documentation, and communication tools.

1. **Manual Testing Techniques**: Manual testing will employ a variety of techniques, including exploratory testing, usability testing, and acceptance testing. Testers will follow predefined test cases and scenarios to systematically validate the software.
2. **Documentation Tools**: Comprehensive test documentation will be created and managed using tools such as spreadsheets or document editors. Test cases, test scenarios, and test results will be documented for traceability and future reference.
3. **Communication Tools**: Effective communication among team members is essential. Collaboration tools, such as project management platforms, messaging apps, and email, will be utilized to facilitate communication, share testing progress, and address any issues identified during testing.

**2.3 Test Schedule**

The manual testing process will be conducted over a period of 3 weeks, starting on 12th November and concluding on 2nd December. This timeline is strategically planned to ensure a thorough evaluation of the **Student Course Enrollment System**  ule while allowing sufficient time for test execution, issue resolution, and documentation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Testing Focus** | **Details** |
| **Week 1** | **Functional Testing: Core Features** | | | |
| November 12 - 13 | User Authentication Testing | Validate login functionality, handle valid/invalid credentials, and session management for students and admins. |
| November 14 - 15 | Class Viewing and Search Testing | Test viewing available classes, searching by keyword, and filtering by category. Verify display and accuracy of class details. |
| November 16 - 17 | Student Enrollment Testing | Test enrollment functionality: successful enrollments, class limits, seat availability, and duplicate enrollments. |
| 11/18/24 | Review and Documentation | Document test results for authentication, class viewing, and enrollment. Identify any issues and scenarios for retesting. |
| **Week 2** | **Admin Functions and System-Level Testing** | | | |
| November 19 - 20 | Class Creation and Management Testing | Test class creation by admins, unique names, enrollment limits, and handling of missing/invalid data. Verify updates to class lists. |
| November 21 - 22 | Manage Class Enrollment Testing | Validate admin functionality for managing enrollments, updating progress, and marking students as complete. |
| November 23 - 24 | System-Level Testing | Test concurrent user access and session expiration for system stability and security during peak usage. |
| 11/25/24 | Review and Documentation | Document test results for admin functionalities and system-level scenarios. Prepare for retesting and final integration checks. |
| **Week 3** | **Security, Data Integrity, and Regression Testing** | | | |
| November 26 - 27 | Security Testing | Test encryption of sensitive data, enforce access permissions, and ensure protection against unauthorized access. |
| November 28 - 29 | Data Consistency and Integrity Testing | Verify data integrity in scenarios like class or student deletion to ensure consistent record updates. |
| November 30 - December 1 | Regression Testing | Retest failed or corrected scenarios from previous weeks to confirm fixes and stability. |
| 12/2/24 | Final Review and Report Preparation | Compile the final test report summarizing test results, issues, and recommendations for deployment. |

**2.4 Testcases**

|  |  |
| --- | --- |
| **ID** | **TC01** |
| **Test Input** | Attempt to log in with valid credentials. |
| **Steps** | 1) Go to login page. |
| 2) Enter valid credentials. |
| 3) Click "Login." |
| **Expected Output** | Student logs in successfully and is redirected to the home page. Login time is recorded. |
| **Description** | Confirm that a student can log in with valid credentials and access the system. |
|  |  |
|  |  |
| **ID** | **TC02** |
| **Test Input** | Attempt to log in with an incorrect password. |
| **Steps** | 1) Enter valid email. |
| 2) Enter incorrect password. |
| 3) Click "Login." |
| **Expected Output** | System displays error message for incorrect credentials, login fails. |
| **Description** | Confirm that the system prevents login with incorrect credentials. |
|  |  |
|  |  |
| **ID** | **TC03** |
| **Test Input** | Attempt to log in leaving the username and/or password fields empty. |
| **Steps** | 1) Leave email and/or password fields blank. |
| 2) Click "Login." |
| **Expected Output** | System prompts user to fill in required fields. |
| **Description** | Ensure the system requires all login fields to be completed before submission. |
|  |  |
|  |  |
| **ID** | **TC04** |
| **Test Input** | Attempt to log in after multiple failed attempts (account lock). |
| **Steps** | 1) Enter incorrect credentials repeatedly (e.g., 5 times). |
| 2) Attempt to log in again. |
| **Expected Output** | Account is locked; user is prompted to contact support or wait to retry. |
| **Description** | Validate that the system locks the account after multiple failed login attempts to enhance security. |
|  |  |
|  |  |
| **ID** | **TC05** |
| **Test Input** | Session timeout due to inactivity. |
| **Steps** | 1) Log in successfully. |
| 2) Remain inactive for the specified timeout period. |
| 3) Attempt to navigate the system. |
| **Expected Output** | User is automatically logged out and prompted to log in again. |
| **Description** | Confirm that the system logs out inactive users to maintain security. |
|  |  |
|  |  |
| **ID** | **TC06** |
| **Test Input** | View all available classes as a student. |
| **Steps** | 1) Log in as a student. |
| 2) Navigate to the "Available Classes" page. |
| **Expected Output** | A list of all available classes is displayed with details such as name, description, seat availability, and schedule. |
| **Description** | Verify that students can view the list of all available classes. |
|  |  |
|  |  |
| **ID** | **TC07** |
| **Test Input** | Search for classes using a keyword. |
| **Steps** | 1) Log in as a student. |
| 2) Navigate to the "Available Classes" page. |
| 3) Enter a keyword in the search bar. |
| 4) Click "Search." |
| **Expected Output** | System displays classes matching the keyword entered. |
| **Description** | Ensure the search functionality returns relevant classes based on the keyword. |
|  |  |
|  |  |
| **ID** | **TC08** |
| **Test Input** | Filter classes by category. |
| **Steps** | 1) Log in as a student. |
| 2) Navigate to the "Available Classes" page. |
| 3) Select a specific category from the filter options. |
| **Expected Output** | System displays classes that fall under the selected category. |
| **Description** | Verify that the filtering functionality correctly displays classes based on the selected category. |
|  |  |
|  |  |
| **ID** | **TC09** |
| **Test Input** | Attempt to view available classes when none are available. |
| **Steps** | 1) Log in as a student. |
| 2) Navigate to the "Available Classes" page. |
| **Expected Output** | System displays a message indicating that no classes are currently available. |
| **Description** | Ensure the system handles scenarios where no classes are available gracefully. |
|  |  |
|  |  |
| **ID** | **TC10** |
| **Test Input** | Enroll in a class with available seats (successful enrollment). |
| **Steps** | 1) Log in as a student. |
| 2) Select a class with available seats. |
| 3) Click "Enroll." |
| **Expected Output** | Student is successfully enrolled; seat availability updates; confirmation message displayed. |
| **Description** | Confirm that students can enroll in classes with available seats. |
|  |  |
|  |  |
| **ID** | **TC11** |
| **Test Input** | Attempt to enroll in a class when already enrolled in three active classes. |
| **Steps** | 1) Log in as a student. |
| 2) Ensure enrollment in three active classes. |
| 3) Attempt to enroll in an additional class. |
| **Expected Output** | System displays a message indicating that the maximum number of active classes has been reached. |
| **Description** | Verify that the system enforces the maximum active class limit per student. |
|  |  |
|  |  |
| **ID** | **TC12** |
| **Test Input** | Attempt to enroll in a class that is full (no seats available). |
| **Steps** | 1) Log in as a student. |
| 2) Select a class with no available seats. |
| 3) Click "Enroll." |
| **Expected Output** | System displays a message indicating that the class is full. |
| **Description** | Ensure the system prevents enrollment in classes that have reached maximum capacity. |
|  |  |
|  |  |
| **ID** | **TC13** |
| **Test Input** | Attempt to enroll in the same class more than once. |
| **Steps** | 1) Log in as a student. |
| 2) Enroll in a class. |
| 3) Attempt to enroll in the same class again. |
| **Expected Output** | System prevents duplicate enrollment and displays an appropriate error message. |
| **Description** | Confirm that the system does not allow duplicate enrollments in the same class. |