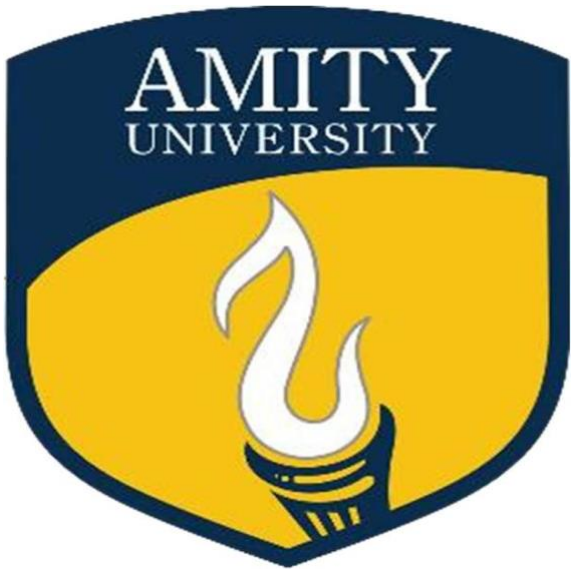
AMITY UNIVERISTY UTTAR PRADESH NOIDA

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



Topic: University Course Registration System

Software Project Management

[CSE432]

Project File

Submitted To:
Dr Misha Kakkar

Submitted By:

- Dhruv Varshney
  A2305221157
- Boddu Asmitha Bhavya
  A2305221386
- Pratyaksha Sharma
  A2305221652
- V Sri Hrudya
  A2305221030

7CSE6-X

TABLE OF CONTENTS (TOC)

1.0 INTRODUCTION

This section introduces the purpose and importance of the testing document. It outlines the primary objectives of testing the Course Registration System, provides an overview of the system, and explains how testing will ensure the delivery of a robust, functional, and high-performance platform for students, instructors, and administrators.

2.0 OBJECTIVES AND TASKS

2.1 Objectives

- Ensure all modules of the Course Registration System meet the functional requirements.
- Identify and resolve bugs or inconsistencies in early stages.
- Ensure the system is reliable, secure, and user-friendly.
- Validate performance under load, ensuring smooth operation during peak usage (e.g., course registration periods).

2.2 Tasks

- Design detailed test cases for each feature and module.
- Conduct tests across different environments (e.g., development, production).
- Monitor performance metrics and identify bottlenecks.
- Log bugs and follow up with developers for quick resolutions.
- Perform regression testing for any new code deployment or updates.

3.0 SCOPE

Defines the boundaries of testing for the Course Registration System, including functional testing (user login, course registration, attendance tracking), integration testing (interaction between modules), performance testing (system behavior under heavy load), and security testing. The scope also clarifies what modules and features are within and outside the scope of testing.

4.0 Testing Strategy

This section outlines various testing methodologies and their specific focus areas.

4.1 Alpha Testing (Unit Testing)

- Unit testing of individual components to verify that each module functions independently.
- Example: Testing the registration form to ensure valid data entries are accepted and invalid ones are rejected.

4.2 System and Integration Testing

- Verifies that all modules work together seamlessly (e.g., the login, course selection, and payment modules interact correctly).

- Example: Ensure the login system passes credentials to the course selection module without errors.

4.3 Performance and Stress Testing

- Tests the system's response under high loads to identify performance bottlenecks.
- Example: Simulate multiple users registering for courses at the same time to measure system responsiveness.

4.4 User Acceptance Testing (UAT)

- Involves real users testing the system to validate that it meets their expectations and usability standards.
- Example: A group of students performs course registrations to provide feedback on the system's usability.

4.5 Batch Testing

- Validates batch processes such as generating attendance reports and grade summaries.
- Example: Test the automated generation of attendance reports at the end of the semester.

4.6 Automated Regression Testing

- Use of automated tools to ensure that changes or updates do not affect the existing functionality.
- Example: Automate testing of the login and course modules to detect any unintended issues after updates.

4.7 Beta Testing

- Allows a limited group of users to test the system in a live environment to provide real-world feedback before full deployment.

5.0 Hardware Requirements

Specifies the hardware setup required to perform the tests effectively.

- **Server**: Intel Xeon Processor, 32 GB RAM, 1 TB HDD
- **Client Machines**: Core i5 or i7 processors, 8 GB RAM, minimum 256 GB SSD

6.0 Environment Requirements

Details the software environment for executing tests, including servers, databases, and operating systems.

6.1 Main Frame

- **Operating System**: Linux/Windows Server
- **Database**: MySQL
- **Network**: LAN/WAN access for testing interactions across systems

6.2 Workstation

- **Operating System**: Windows/Mac/Linux

- **Browsers**: Chrome, Firefox, Edge
- **Development Tools**: Visual Studio Code, MySQL Workbench

7.0 Test Schedule

A detailed timeline to ensure that testing phases are completed on time.

Phase	**Start Date**	**End Date**	**Activity**
Alpha Testing	21-Aug-2024	25-Aug-2024	Unit tests of individual components
Integration Testing	26-Sept-2024	31-Sept-2024	Verify module interactions
Performance Testing	01-Oct-2024	02-Oct-2024	Test under peak load
User Acceptance Testing	03-Oct-2024	05-Oct-2024	UAT with students and staff

Here's the Traceability Test Matrix (TTM) for all test cases:

Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution	Defects?	Defect ID	Defect Status
4.1.3-REQ-1	The system must allow students to browse courses by department, level, and availability status.	TC-01	Verify that students can browse courses by department, level, and availability.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.1.3-REQ-2	The system must display course details, including prerequisites, credit hours, and schedule.	TC-02	Verify that course details are displayed correctly with all required information.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A

Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution	Defects?	Defect ID	Defect Status
4.1.3-REQ-3	The system must allow students to register for open courses and provide immediate confirmation of successful registration.	TC-03	Verify that students can register for open courses and receive confirmation.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.1.3-REQ-4	If a course is full, the system must provide the option to join a waitlist.	TC-04	Verify that the waitlist option is available when a course is full.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.1.3-REQ-5	The system must allow students to drop courses before a specified deadline.	TC-05	Verify that students can drop courses before the specified deadline.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.2.3-REQ-1	The system must allow students to view their personal	TC-06	Verify that students can view their course schedule in a user-	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A

Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution	Defects?	Defect ID	Defect Status
	course schedule in a user-friendly format.		friendly format.							
4.2.3-REQ-2	The system must support exporting the schedule to external calendar systems (e.g., Google Calendar, iCal).	TC-07	Verify that students can export their schedule to external calendar systems.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.2.3-REQ-3	The system must automatically update the schedule when courses are added or dropped.	TC-08	Verify that the schedule updates automatically when courses are added or dropped.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.2.3-REQ-4	The system must send reminders for upcoming classes and exams.	TC-09	Verify that the system sends reminders for upcoming classes and exams.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A

Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution	Defects?	Defect ID	Defect Status
4.3.3-REQ-1	The system must allow faculty to view and manage the list of enrolled students.	TC-10	Verify that faculty can view and manage the list of enrolled students.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.3.3-REQ-2	The system must support the entry of student grades for assignments and exams.	TC-11	Verify that faculty can enter student grades for assignments and exams.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.3.3-REQ-3	The system must automatically calculate final grades based on weights assigned to each assessment.	TC-12	Verify that the system calculates final grades correctly based on the assigned weights.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.3.3-REQ-4	The system must provide attendance tracking and alert faculty for students	TC-13	Verify that the system tracks attendance and alerts faculty for	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A

Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution	Defects?	Defect ID	Defect Status
	with low attendance rates.		students with low attendance.							
4.4.3-REQ-1	The system must allow administrators to create new user accounts and assign roles.	TC-14	Verify that administrators can create new user accounts and assign roles.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.4.3-REQ-2	The system must allow administrators to modify user roles and permissions.	TC-15	Verify that administrators can modify user roles and permissions.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.4.3-REQ-3	The system must allow administrators to delete user accounts and associated data securely.	TC-16	Verify that administrators can delete user accounts and associated data securely.	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A
4.4.3-	The system must	TC-17	Verify that the system	Manual	[Designer Name]	Yes	Completed	No	N/A	N/A

Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution	Defects?	Defect ID	Defect Status
REQ-4	provide audit logs for all user account modifications made by administrators.		provides audit logs for user account modifications.							

Test Cases

Course Registration System

1. **TC-01**: Browse Courses
    - **Description**: Verify that students can browse courses by department, level, and availability.
    - **Preconditions**: Student is logged in.
    - **Steps**:
        1. Navigate to the course browsing section.
        2. Select a department.
        3. Select a level (e.g., undergraduate, graduate).
        4. Check the availability status.
    - **Expected Result**: A list of courses matching the selected criteria is displayed.
2. **TC-02**: Display Course Details
    - **Description**: Verify that course details, including prerequisites, credit hours, and schedule, are displayed correctly.
    - **Preconditions**: Course is available.
    - **Steps**:
        1. Select a specific course from the course list.
    - **Expected Result**: The course details page displays prerequisites, credit hours, and the schedule accurately.
3. **TC-03**: Register for Course
    - **Description**: Verify that students can register for open courses and receive immediate confirmation.

- **Preconditions**: Course is open for registration.
- **Steps**:
    1. Select an open course.
    2. Click on the register button.
- **Expected Result**: A confirmation message is displayed, confirming successful registration.

4. **TC-04**: Join Waitlist for Full Course
    - **Description**: Verify that the waitlist option is available when a course is full.
    - **Preconditions**: Course is full.
    - **Steps**:
        1. Select a full course.
        2. Click on the waitlist button.
    - **Expected Result**: A confirmation message is displayed indicating that the student has been added to the waitlist.
5. **TC-05**: Drop Course
    - **Description**: Verify that students can drop courses before the specified deadline.
    - **Preconditions**: Student is registered in a course.
    - **Steps**:
        1. Navigate to the enrolled courses section.
        2. Select a course to drop.
        3. Click on the drop button.
    - **Expected Result**: A confirmation message is displayed, and the course is removed from the student's schedule.

Course Schedule Management

6. **TC-06**: View Personal Course Schedule
    - **Description**: Verify that students can view their personal course schedule in a user-friendly format.
    - **Preconditions**: Student is logged in and has courses enrolled.
    - **Steps**:
        1. Navigate to the personal schedule section.
    - **Expected Result**: The schedule is displayed in a clear and user-friendly format.
7. **TC-07**: Export Schedule to External Calendar
    - **Description**: Verify that students can export their schedule to external calendar systems (e.g., Google Calendar, iCal).

- **Preconditions**: Student has a schedule available.
- **Steps**:
    1. Navigate to the schedule section.
    2. Click on the export button.
    3. Select an external calendar system.
- **Expected Result**: The schedule is exported successfully to the selected calendar system.

8. **TC-08**: Automatic Schedule Update
    - **Description**: Verify that the schedule updates automatically when courses are added or dropped.
    - **Preconditions**: Student has courses enrolled.
    - **Steps**:
        1. Drop a course or add a new course.
        2. Navigate to the schedule section.
    - **Expected Result**: The schedule reflects the changes made immediately.
9. **TC-09**: Send Class and Exam Reminders
    - **Description**: Verify that the system sends reminders for upcoming classes and exams.
    - **Preconditions**: Student has enrolled courses with upcoming classes/exams.
    - **Steps**:
        1. Wait for the reminder notification period.
    - **Expected Result**: A reminder notification is received for the upcoming class/exam.

Faculty Management

10. **TC-10**: View and Manage Enrolled Students
    - **Description**: Verify that faculty can view and manage the list of enrolled students.
    - **Preconditions**: Faculty is logged in.
    - **Steps**:
        1. Navigate to the enrolled students section.

- **Expected Result**: The list of enrolled students is displayed.

11. **TC-11**: Enter Student Grades
    - **Description**: Verify that faculty can enter student grades for assignments and exams.

- **Preconditions**: Faculty is logged in and has access to grade entries.
- **Steps**:
    1. Select a student.
    2. Enter grades for assignments/exams.

- **Expected Result**: The grades are saved successfully.

12. **TC-12**: Calculate Final Grades
    - **Description**: Verify that the system calculates final grades based on weights assigned to each assessment.
    - **Preconditions**: Grades for assessments are entered.
    - **Steps**:
        1. Navigate to the final grades section.

- **Expected Result**: The final grades are calculated and displayed correctly.

13. **TC-13**: Attendance Tracking
    - **Description**: Verify that the system provides attendance tracking and alerts faculty for students with low attendance rates.
    - **Preconditions**: Faculty is logged in and attendance is recorded.
    - **Steps**:
        1. Navigate to the attendance tracking section.

- **Expected Result**: Attendance records are displayed, and alerts for low attendance are shown.

Administrator Management

14. **TC-14**: Create User Accounts
    - **Description**: Verify that administrators can create new user accounts and assign roles.
    - **Preconditions**: Administrator is logged in.
    - **Steps**:
        1. Navigate to the user account management section.
        2. Create a new user account.
        3. Assign a role to the new account.

- **Expected Result**: The user account is created successfully with the assigned role.

15. **TC-15**: Modify User Roles and Permissions
    - **Description**: Verify that administrators can modify user roles and permissions.
    - **Preconditions**: Administrator is logged in.

- **Steps**:
    1. Navigate to the user account management section.
    2. Select a user account to modify.
    3. Change the user's role or permissions.

- **Expected Result**: The user's role or permissions are updated successfully.

16. **TC-16**: Delete User Accounts
    - **Description**: Verify that administrators can delete user accounts and associated data securely.
    - **Preconditions**: Administrator is logged in.
    - **Steps**:
        1. Navigate to the user account management section.
        2. Select a user account to delete.

- **Expected Result**: The user account is deleted securely, and confirmation is received.

17. **TC-17**: Audit Logs for User Account Modifications
    - **Description**: Verify that the system provides audit logs for all user account modifications made by administrators.
    - **Preconditions**: Administrator is logged in.
    - **Steps**:
        1. Navigate to the audit log section.

- **Expected Result**: A detailed log of user account modifications is displayed.

8.0 Control Procedures

Defines procedures to manage the testing process, including test case execution, defect logging, and reporting mechanisms.

- **Defect Tracking**: Use JIRA to log and monitor bugs.
- **Daily Status Reports**: Track testing progress and issues.
- **Review Meetings**: Weekly reviews to ensure alignment with the schedule.

9.0 Features to Be Tested

- **User Authentication**: Verify login and session handling.
- **Course Registration**: Test course enrollment processes, including pre-requisites checks.
- **Attendance and Grades Management**: Validate attendance tracking and grade submission modules.
- **Notifications and Alerts**: Ensure reminders and notifications are generated correctly.

10.0 Features Not to Be Tested

- **Third-party Payment Systems**: Only integration testing will be performed for payment modules.
- **Database Maintenance Scripts**: Internal maintenance jobs will not be part of the testing scope.

11.0 Resources/Roles & Responsibilities

Defines the key resources and their responsibilities in the testing process.

- **Test Manager**: Plan, coordinate, and manage testing activities.
- **Test Engineers**: Execute test cases and document results.
- **Developers**: Analyze defects and implement fixes.
- **Business Analysts**: Review UAT results and provide feedback.

12.0 Schedules

Includes detailed milestones and deadlines for each testing phase.

Task	Deadline
Test Case Design Completion	20-Sept-2024
Alpha Testing Completion	25-Sept-2024
UAT Completion	05-Oct-2024
Final Report Submission	22-Oct-2024

13.0 Significantly Impacted Departments (SIDs)

Lists the departments most affected by the Course Registration System.

- **Academic Department**: Impacted by course registrations and attendance.
- **IT Support**: Responsible for system maintenance and troubleshooting.
- **Student Affairs**: Impacted by notifications and student interactions.

14.0 Dependencies

Identifies factors that may impact the testing process.

- **Development Team Availability**: Critical for defect resolution.
- **Test Data**: Must be ready before the testing phase begins.

15.0 Risks/Assumptions

Lists the risks and assumptions associated with the testing process.

- **Risk**: Server downtime during testing could delay completion.

- **Assumption**: All modules will be stable by the testing start date.

16.0 Tools

Specifies the tools required to execute the testing process.

- **Defect Tracking**: JIRA
- **Automation Testing**: Selenium WebDriver
- **Performance Testing**: JMeter

17.0 Approvals

This section records approvals from key stakeholders for the testing plan.

Role	Name	Signature	Date
Test Manager	[Name]		[Date]
Project Manager	[Name]		[Date]
Client Representative	[Name]		[Date]

This testing document ensures a structured approach to validating the Course Registration System, addressing functional, performance, and user experience aspects to deliver a reliable and efficient platform.