



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Fall 2023-24

Section: B

Software Quality And Testing

STUDENT MANAGEMENT SYSTEM

Report submitted By

SN	Student Name	Student ID
1	Thithy Banik	19-41557-3
2	Shithi Roy	19-41545-3
3	Sakib Shadman	19-39373-1
4	Shakhawat Amin	18-38312-2

Under the supervision of

Abhijit Bhowmik

Associate Professor

Department of Computer Science

Faculty of Science and Technology American International University-
Bangladesh (AIUB)

Software Test Plan

for

Student Management System

Version 1.0 approved

Prepared by Thithy,
Shithi,
Shakhawat,
Sakib

American International University- Bangladesh

23/12/23

Table of Contents

Revision History	3
1. TEST PLAN IDENTIFIER: STUDENT MANAGEMENT SYSTEM- MTP01.3	4
2. REFERENCES	4
3. INTRODUCTION	4
Background to the Problem	4
Solution to the Problem	4
4. REQUIREMENT SPECIFICATION.....	5
4.1 System Features.....	4
4.2 System Quality Attributes	7
4.3 System Interface	7
4.4 Project Requirements	11
5. FEATURES NOT TO BE TESTED.....	11
6. TESTING APPROACH	12
6.1 Testing Levels	12
6.2 Test Tools.....	13
6.3 Meetings	13
6.4 TEST CASES/TEST ITEMS.....	16
7. ITEM PASS/FAIL CRITERIA	19
8. TEST DELIVERABLES	20
9. STAFFING AND TRAINING NEEDS.....	20
10. RESPONSIBILITIES	21
11. TESTING SCHEDULE	22
12. PLANNING RISKS AND CONTINGENCIES	23
13. APPROVALS.....	23

Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.11.29	Thithy	Version 1
0.2	2023.12.02	Shithi	Version 2
0.3	2023.12.05	Sakib	Version 3
0.4	2023.12.08	Shakhawat	Version 4
0.5	2023.12.13	Shithi	Version 5
0.6	2023.12.16	Thithy	Version 6
0.7	2023.12.21	Shakhawat	Version 7

1. TEST PLAN IDENTIFIER: [Student Management System - MTP01.3](#)

2. REFERENCES

<https://itsourcecode.com/free-projects/php-project/student-management-system-in-php-with-source-code/>

3. INTRODUCTION

➤ Background to the Problem

The Student Management System (SMS) project aims to efficiently organize and manage student-related information within an educational institution. In the context of software quality and testing, ensuring the reliability and functionality of the system is crucial.

The introduction should highlight the significance of the SMS in streamlining administrative tasks, enhancing communication, and improving overall efficiency. Emphasize the importance of a robust software quality assurance process to identify and rectify potential issues before deployment.

Discuss the key objectives of the system, such as accurate data storage, user-friendly interfaces, and secure access controls. The introduction should set the stage for the subsequent sections, where the focus will shift to the methodologies employed for software quality assurance and testing throughout the project lifecycle.

➤ Solution to the Problem

To ensure a high-quality Student Management System (SMS) project, a comprehensive approach to software quality and testing is essential. Here's a solution framework:

Requirement Analysis:

- Thoroughly understand and document functional and non-functional requirements.
- Involve stakeholders to validate and refine requirements.

Test Planning:

- Develop a detailed test plan outlining testing objectives, scope, resources, and schedules.
- Define test cases and scenarios based on identified requirements.

Testing Levels:

- Conduct unit testing to validate individual components.
- Perform integration testing to ensure seamless interaction between modules.
- Execute system testing to verify the entire system's functionality.

Testing Types:

- Implement functional testing to validate specific features and functions.

- Conduct performance testing to assess system responsiveness and scalability.
- Execute security testing to identify and address potential vulnerabilities.

Automation Testing:

- Employ automation tools to expedite repetitive and regression testing.
- Prioritize test cases for automation based on criticality and frequency of use.

User Acceptance Testing (UAT):

- Collaborate with end-users to conduct UAT for real-world validation.
- Address and resolve feedback from users during this phase.

Bug Tracking and Resolution:

- Utilize a robust bug tracking system to log and prioritize identified issues.
- Implement a systematic approach to resolve and retest reported bugs.

Performance Monitoring:

- Implement continuous monitoring tools to identify and address performance bottlenecks.
- Analyze system behavior under various load conditions.

Documentation:

- Maintain comprehensive documentation for test cases, test results, and issue resolution.
- Ensure that documentation is accessible and understandable to all stakeholders.

Continuous Improvement:

- Conduct regular reviews of the testing process to identify areas for improvement.
- Encourage a culture of continuous learning and adaptation.
- By implementing this solution framework, the Student Management System project can achieve a high level of software quality and effectively address potential issues throughout the development lifecycle.

4. REQUIREMENT SPECIFICATION

3.1 System Feature

4.1.1 User Signup:

- New student will require valid personal information such as phone number, name, email and password.
- If the registration process is successful, the User will see a popup confirming such and will be redirected to Login page.
- Priority level: High
- Precondition: User must have valid information's.

4.1.2 Student Login:

- Student will log into the system with their own Username and password.
- If the login is successful, the Patient will be redirected to the website homepage. Otherwise, it will load the login page again with wrong credentials message.
- Priority Level: High
- Precondition: Student must have valid Username and password.

4.1.3 View Result:

- A student can view their result.

- Student have to first select a specific subject and then they can view their result.
- Priority Level: Medium
- Precondition: Student must have valid Username and password.

4.1.4 Notice:

- A student can check notices.
- Priority Level: Medium
- Precondition: Student must have valid Username and password.

4.1.5 Select Group/Department:

- Students can select their group or department.
- Priority Level: Medium
- Precondition: Student must have valid Username and password.

4.1.6 View Classmates

- Students can view their classmates.
- Priority Level: Medium
- Precondition: Student must have valid Username and password.

4.1.7 User Logout

- A user will be able to log out of the system from his valid account.
- Logout will be successful if after pressing logout it redirects to the login page.
- Priority Level: High
- Precondition: User needs to successfully log in first

4.2 System Quality Attributes

System quality attributes in a Student Management System project related to software quality and testing typically include:

Reliability: Ensuring the system consistently performs accurately, without errors or failures, to maintain trust and dependability.

Usability: Focusing on the user interface and experience to make the system intuitive and easy to use for both administrators and students.

Scalability: Addressing the system's ability to handle an increasing amount of data, users, or transactions without compromising performance.

Performance: Evaluating the speed and responsiveness of the system under different loads and conditions to meet the performance expectations.

Security: Implementing measures to protect sensitive student data, user authentication, and system integrity to prevent unauthorized access or data breaches.

Maintainability: Ensuring that the system is easy to maintain and update, with clear documentation and modular design to facilitate future enhancements.

Compatibility: Verifying that the system works seamlessly with various platforms, browsers, and devices to accommodate diverse user environments.

Testability: Designing the system to facilitate effective testing, with well-defined test cases and scenarios to ensure comprehensive coverage.

Flexibility: Allowing the system to adapt to changing requirements, such as updates in educational policies or additional features requested by users.

Interoperability: Ensuring the system can integrate smoothly with other existing educational tools or systems, promoting a cohesive and interconnected environment.

Considering and addressing these quality attributes during the development and testing phases contributes to building a robust and reliable Student Management System

4.1 System Interface

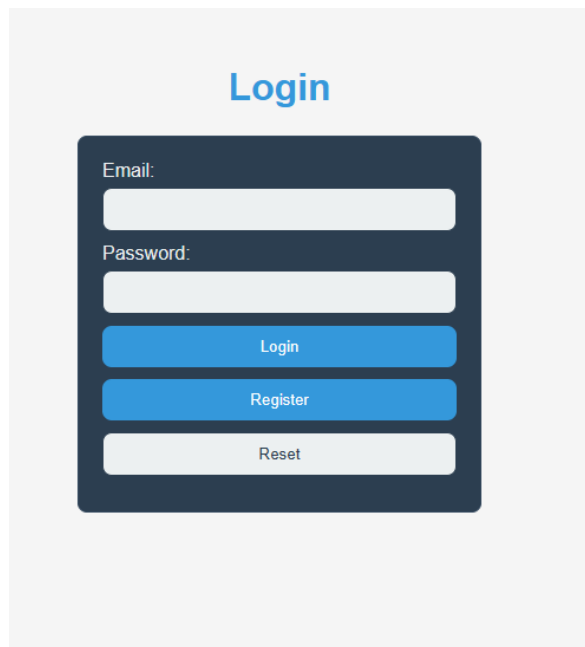
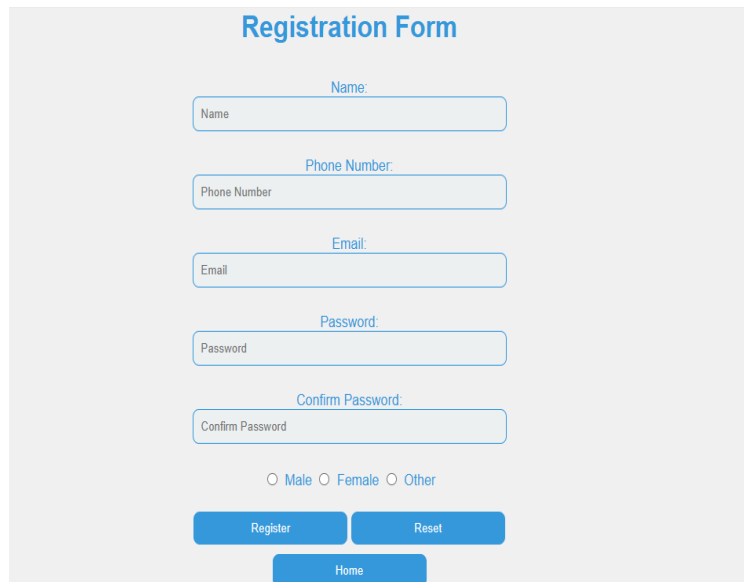
The image shows a login page interface. At the top, the word "Login" is displayed in a blue font. Below it, there is a dark blue rectangular box containing four input fields and three buttons. The first two fields are labeled "Email:" and "Password:" in white text. The third button is blue with the word "Login" in white. The fourth button is blue with the word "Register" in white. The fifth button is white with the word "Reset" in dark blue. The entire interface is set against a light gray background.

Figure1: Login Page

User have to login with email and password.



The registration form is titled "Registration Form" in blue. It contains several input fields: "Name", "Phone Number", "Email", "Password", and "Confirm Password". Below these fields are three radio buttons for "Male", "Female", and "Other". At the bottom, there are three buttons: "Register", "Reset", and "Home".

Registration Form

Name:

Phone Number:

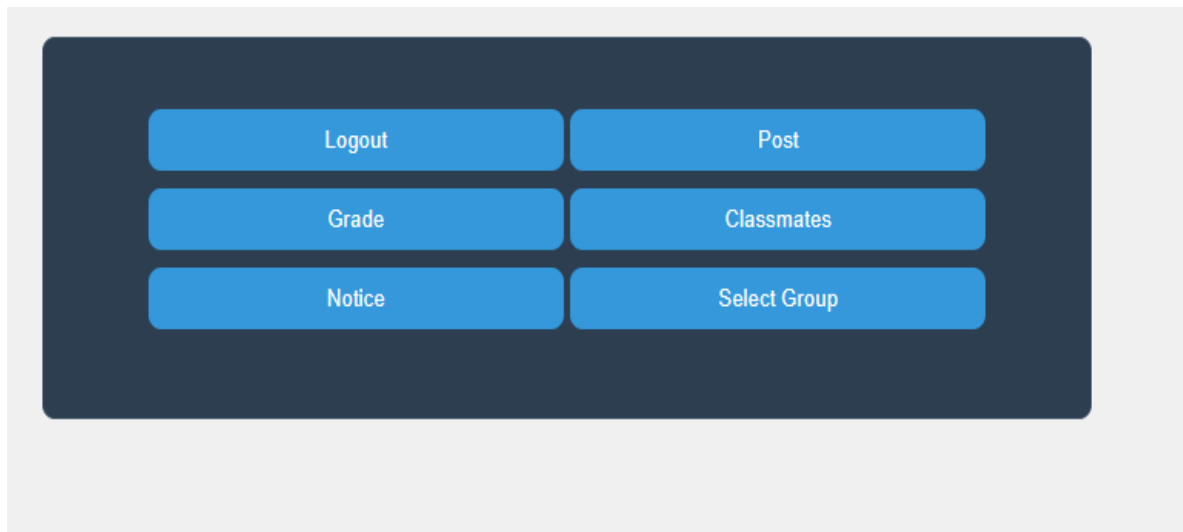
Email:

Password:

Confirm Password:

☐ Male ☐ Female ☐ Other

Figure2: Registration Page



The home page features a dark blue background with a grid of six blue buttons. The buttons are arranged in three rows and two columns. The first row contains "Logout" and "Post". The second row contains "Grade" and "Classmates". The third row contains "Notice" and "Select Group".

Logout Post

Grade Classmates

Notice Select Group

Figure3: Home Page

This is Home page. After successfully login into the system, student will redirect this page.

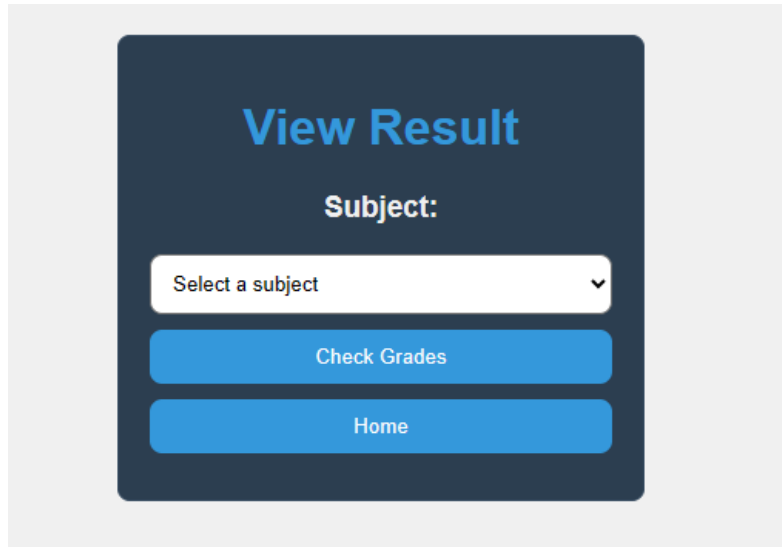


Figure4: Grade Page

This is Grade page. Student can check their result from this page.

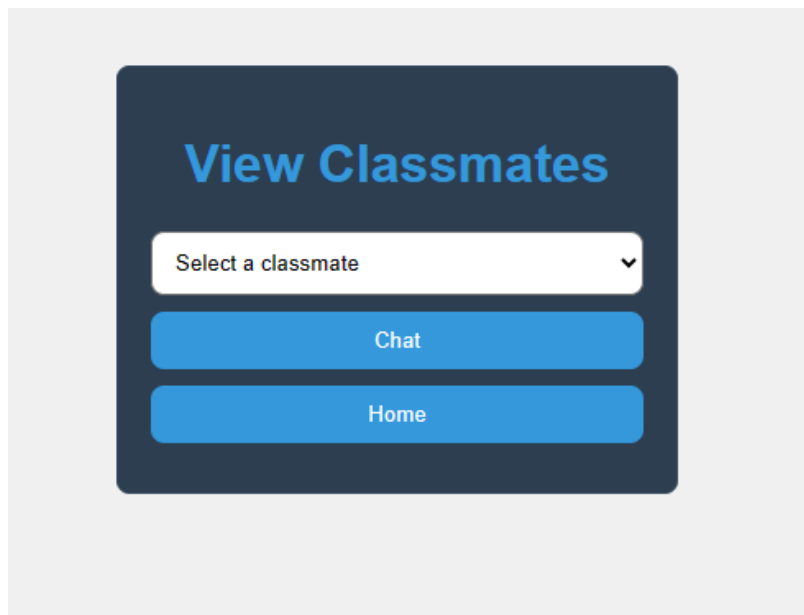


Figure5: Classmates

Students can view their classmates from this page.

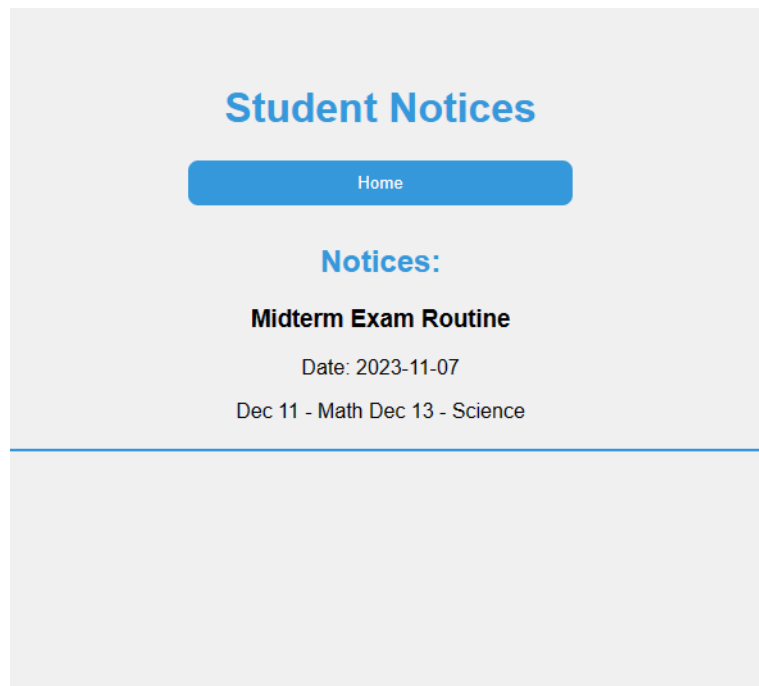


Figure6: Notice Page

From here, student can check notices.

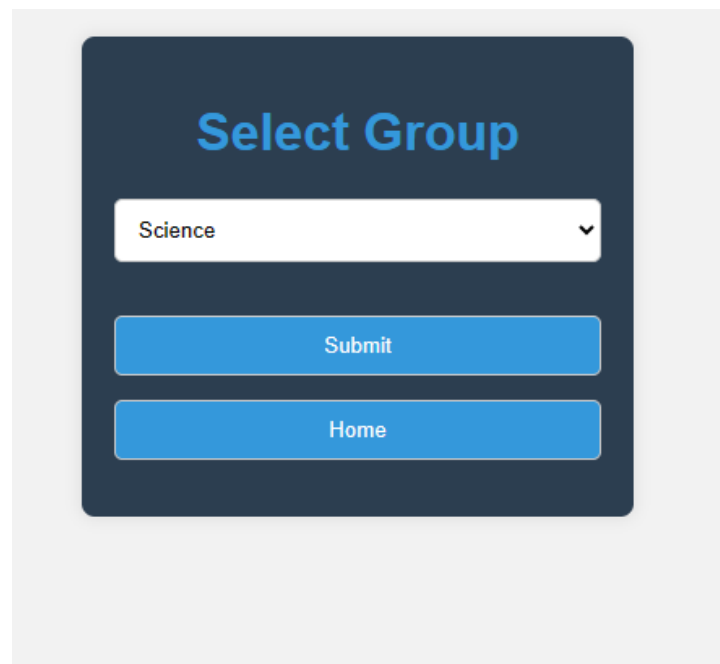


Figure7: Select Group

From here. students can select their group or department.

4.3 Project Requirements

- Time: This web-based application may take about 20 days to complete.
- Budget: 4,00,000 BDT
- Size: The final size of this web-based application will not be more than 350 MB.
- HTML, CSS, PHP, JavaScript, JQuery and Ajax will be used to build this web-based application.

5. FEATURES NOT TO BE TESTED

- Out-of-Scope Functionalities:
 - Features beyond project scope or defined requirements.
- Third-Party Integrations:
 - External systems assumed to have their own testing.
- Unsupported Browsers or Devices:
 - Testing limited to specified compatibility matrix.
- Non-Functional Requirements:
 - Performance characteristics, unless specified.
- Deprecated Functionality:
 - Features set for removal in future releases.
- Security Vulnerabilities:
 - Known vulnerabilities with documented measures.

6. TESTING APPROACH

6.1 Testing Levels

❖ Unit Testing:

1. Identifying Units:

- Break down the student management system into smaller units or components, such as student registration, grade calculation, attendance tracking, and user authentication.

2. Set Up Testing Environment:

- Create a dedicated testing environment mirroring the production setup, including databases, libraries, and configurations.

3. Testing Framework:

- Choose an appropriate testing framework (e.g., JUnit for Java) for unit testing.

4. Write Test Cases:

- Develop a comprehensive set of test cases for each unit, covering positive and negative scenarios. For example, test cases for student registration may include successful registration, duplicate entries, and invalid inputs.

5. Use Mocking and Stubbing:

- Implement mocking and stubbing techniques to isolate units under test from external dependencies.

6. Write Test Code:

- Implement test cases using the chosen testing framework, following a structured format: setup, execution, and assertion.

7. Automate Testing:

- Automate the execution of test cases to facilitate easy testing with code changes and prevent regressions.

8. Test Coverage Analysis:

- Regularly analyze test coverage to ensure that a significant portion of the codebase is exercised by unit tests.

9. Maintain Test Suite:

- Update and maintain the test suite as the student management system evolves. Ensure that new features and changes come with corresponding test cases.

10. Regression Testing:

- Perform regression testing after significant changes to ensure that existing functionality remains intact.

❖ System Testing:

1. Define Test Scenarios:

- Identify various scenarios encountered by users, such as student registration, grade updates, and attendance tracking. Create detailed test cases for these scenarios.

2. Testing Environment Setup:

- Set up a testing environment that closely resembles the production setup, including databases, servers, network configurations, and third-party integrations.

3.Integrate Components:

- Ensure seamless integration of individual components into the system, verifying effective communication and accurate data sharing.

4.End-to-End Scenarios:

- Execute end-to-end test scenarios covering the entire student management process, including student enrollment, grade updates, and generating reports.

5.User Interactions:

- Thoroughly test user interfaces to verify smooth navigation, error-free data input, and appropriate feedback.

6.Performance Testing:

- Test the system's performance under different conditions, including load testing and stress testing. Measure its ability to handle increased load gracefully.

7.Security Testing:

- Evaluate the system's security features, including penetration testing to identify potential vulnerabilities. Test authentication mechanisms, data encryption, and authorization controls.

8.Compatibility Testing:

- Ensure the system works correctly across various browsers, devices, and operating systems. Test responsiveness and functionality on mobile devices if applicable.

9.Usability Testing:

- Conduct usability testing with actual users or representatives, gathering feedback on ease of use, intuitiveness, and overall user experience. Make necessary improvements based on feedback.

10.Edge Cases and Boundary Testing:

- Explore edge cases and boundary conditions that might lead to unexpected behavior. Test with extreme values, invalid inputs, and scenarios that could potentially break the system.

11.Recovery and Backup Testing:

- Simulate system failures and recovery scenarios. Test backup and data recovery processes to ensure data restoration in case of failure.

12.Documentation Review:

- Review system documentation, including user manuals and technical guides, to ensure accuracy and completeness.

13. Regression Testing:

- Perform regression testing after making fixes or enhancements based on earlier testing phases to ensure that new changes do not impact previously tested functionality.

14. Reporting and Issue Tracking:

- Document all identified issues and defects using a dedicated issue tracking system. Prioritize, assign, and monitor the progress of issue resolution.

15. Final Review and Sign-Off:

- Conduct a final review of testing results and the system's overall performance. Obtain approval from relevant stakeholders once all issues are resolved, and the system meets the desired criteria.

❖ Acceptance Testing :

1. Objective:

- Confirm that the Student Management System meets the specified requirements and is ready for deployment by satisfying user expectations and business needs.

2. Testing Levels:

➤ User Acceptance Testing (UAT):

Conducted by actual users or representatives to ensure the system aligns with their needs.

➤ Business Acceptance Testing (BAT):

Focused on validating business requirements, ensuring that the system meets organizational objectives.

3. Test Scenarios:

- Scenario 1 - Student Enrollment:
 - Confirm that the system allows for efficient and accurate student enrollment processes.
- Scenario 2 - Grade Submission and Calculation:
 - Validate the accuracy of grade submission and calculation functionalities.
- Scenario 3 - Attendance Tracking:
 - Verify that the system effectively tracks and records student attendance.

4. Usability Testing:

Evaluate the system's ease of use, navigation, and overall user experience.

5. Performance Testing:

Ensure the system performs well under expected user loads and usage scenarios.

6. Security Testing:

Validate security features, including authentication mechanisms, data encryption, and authorization controls.

7. Compatibility Testing:

Confirm the system's compatibility across various browsers, devices, and operating systems.

8. Acceptance Criteria:

Define clear acceptance criteria that the system must meet to be considered acceptable.

9. Feedback Collection:

Gather feedback from users and stakeholders to identify areas for improvement and user satisfaction.

10. Regression Testing:

Perform regression testing to ensure that new changes or fixes do not impact previously accepted functionality.

11. Approval Process:

Obtain formal approval from stakeholders and users once the system meets the defined acceptance criteria.

12. Documentation:

Document acceptance test cases, results, and any issues identified during the testing process for future reference and continuous improvement.

Test Tools

Tools Utilized:

➤ **AS/400 Provided Utilities and Commands:**

Standard utilities and commands provided by AS/400 will be the primary tools for testing purposes. These tools will be leveraged for various testing activities.

➤ **Program Development Manager (PDM):**

PDM will serve as the source version configuration management tool. It will be used in conjunction with an in-house check-in/check-out control utility, which is integrated into each developer's standard AS/400 access menu.

➤ **AS/400 Screen Design Aid (SDA):**

For the development of initial prototypes, AS/400 Screen Design Aid (SDA) will be employed. It will be used to create the layout and general content of new screens. The prototypes will be shared with the sales administration staff for feedback before proceeding with further testing and development.

Rationale:

- Utilizing standard AS/400 tools ensures consistency and compatibility within the development and testing environment.
- PDM and the check-in/check-out control utility facilitate version control, allowing effective management of source code.

Meetings

➤ **Meeting Schedule:**

➤ **Weekly Test Team Meetings:**

- The test team will convene once a week to assess the progress made, discuss ongoing testing activities, and identify any emerging error trends. These regular meetings aim to address issues promptly and ensure effective collaboration within the team.

➤ **Bi-weekly Test Team Leader Meetings:**

- The test team leader will meet with the development team and the project manager once every two weeks. These meetings will be scheduled on alternate weeks to the weekly team meetings. The purpose is to provide updates, discuss overarching progress, and align testing efforts with the overall project goals.

➤ **Additional Meetings for Emergencies:**

- Additional meetings can be called as needed for addressing emergency situations. These ad-hoc meetings will be organized to promptly resolve critical issues that require immediate attention.
- Meeting Objectives:

➤ **Weekly Test Team Meetings:**

- Evaluate the progress of ongoing testing activities.
- Identify and address error trends and problems early in the testing phase.
- Enhance communication and collaboration within the test team.

➤ **Bi-weekly Test Team Leader Meetings:**

- Provide a higher-level overview of testing progress to development and project management.
- Discuss any challenges or roadblocks hindering testing effectiveness.
- Align testing efforts with the overall project timeline and goals.

➤ **Additional Meetings for Emergencies:**

- Address critical issues that require immediate attention.
- Coordinate rapid responses and solutions for urgent problems.

➤ **Communication:**

- Communication channels such as video conferencing, email, and instant messaging will be utilized for both scheduled and ad-hoc meetings.
- Meeting minutes and action items will be documented and shared with relevant stakeholders after each meeting to ensure transparency and accountability.

7. TEST CASES/TEST ITEMS

Test Case 1:

Project Name: Student Management System		Test Designed by: Tithy Banik		
Test Case ID: FR_1		Test Designed date: 29-11-23		
Test Priority (Low, Medium, High): Medium		Test Executed by: Shithi Roy		
Module Name: Student Registration		Test Execution date: 2-12-23		
Test Title: Registration With valid Information				
Description: Check If Student registration works perfectly with valid information				
Precondition (If any): N/A				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the website 2. Put valid Information 3. Click Login	Put valid Information and fill up all input level	Registration must be successful	Registration Successful	Pass
Post Condition: Redirected to Login page.				

Test Case 2:

Project Name: Student Management System		Test Designed by: Shithi Roy		
Test Case ID: FR_2		Test Designed date: 2-12-23		
Test Priority (Low, Medium, High): High		Test Executed by: Shakhawat Amin		
Module Name: Student Login		Test Execution date: 3-12-23		
Test Title: Student login with valid Username and password				
Description: Check If Student login works perfectly with valid Username and password.				
Precondition (If any): Student must be registered into the system				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the website 2. Put valid Username and password 3. Click Login button	Username: Shithi Roy Password: 1234	Login must be successful	Log in Successful	Pass
Post Condition: Redirected to Home page				

Test Case 3:

Project Name: Student Management System		Test Designed by: : Sakib Shadman		
Test Case ID: FR_3		Test Designed date: 05-12-23		
Test Priority (Low, Medium, High): Medium		Test Executed by: Tithy Banik		
Module Name: View Result		Test Execution date: 06-12-23		
Test Title: Select specific course and view result				
Description: Check If Student can select course and view result perfectly.				
Precondition (If any): Student Must be registered into the system				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the website 2. Then Click Grade 3. Select a subject 4. Click Check Grades	N/A	Result must be successfully viewed	Viewed Result successful	Pass
Post Condition:				

Test Case 4:

Project Name: Student Management System		Test Designed by : Shakhawat Amin		
Test Case ID: FR_4		Test Designed date: 08-12-23		
Test Priority (Low, Medium, High): Medium		Test Executed by: Sakib Shadman		
Module Name: Notice		Test Execution date: 08-12-23		
Test Title: Check Notices				
Description: Check If Student can check notices perfectly				
Precondition (If any): Student Must be registered into the system				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the website 2. Then Notice	N/A	Check notice must be successful	Successful	Pass
Post Condition:				

Taste Case 5:

Project Name: Student Management System		Test Designed by: Shithi Roy		
Test Case ID: FR_5		Test Designed date: 13-12-23		
Test Priority (Low, Medium, High): Medium		Test Executed by: Thithy Banik		
Module Name: Select Group		Test Execution date: 16-12-23		
Test Title:Select Group				
Description: Check If student can select and choose their group				
Precondition (If any): Student Must be registered into the system				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the website 2. Then Click Select Group 3. Select a group 4. Click Submit	Group Name: Science	Select group must be successful	Successfully selected group	Pass
Post Condition:				

Taste Case 6:

Project Name: Student Management System			Test Designed by: Thithy Banik		
Test Case ID: FR_7			Test Designed date: 16-12-23		
Test Priority (Low, Medium, High): High			Test Executed by: Shakhawat Amin		
Module Name: Classmates			Test Execution date: 21-12-23		
Test Title: View Classmate name					
Description: Check If student can view classmates’ name					
Precondition (If any):					
Test Steps		Test Data	Expected Results	Actual Results	Status
1. Go to the website 2. Then Classmates		N/A	Viewed classmates’ name must be successful	successful	Pass
Post Condition:					

08.ITEM PASS/FAIL CRITERIA

The main objective of this section is to describe the PASS/FAIL criteria for the tests that are a part of this project. Any system or unit receiving a score of less than 90% will be subject to the failure criteria, and any component, unit, system, or integrated test item receiving a score of 90% to 95% will be considered to meet the pass criterion

09.TEST DELIVERABLES

Test Deliverables are documents that are given to the stakeholders when the software is being developed. It contains a list of documents, tools, and other equipment that must be created, provided, and maintained to support testing activities in a project.

- Unit testing findings and results will be properly documented. To stay on track, a continuous progress report is required.
- Audience for acceptance tests will be carefully selected, as wrong users can lead to incorrect results and feedback. It is similar to a contract for development team release and software delivery.
- During the time of integration testing, new modules are integrated into the system. And

these records needed to be kept for further checking.

- Project management tools such as Jira, Trello, and others can be used to keep track of the progress report.
- After completing the each of the testing phase the details report will be generated containing the test results.

10. STAFFING AND TRAINING NEEDS

The staffing and training needs for a student learning platform would depend on the specific goals and objectives of the platform, as well as the scope and complexity of the platform itself. Here are some potential staffing and training needs for a student learning platform:

Admin: The platform would likely need one or more administrators who are responsible for managing and maintaining the platform, including managing user accounts, monitoring system performance, and troubleshooting issues. These administrators would need to be trained on the platform's features and tools, as well as any relevant technical skills and protocols.

Content developers: The platform would likely need content developers who are responsible for creating and curating the learning materials, such as videos, readings, and assessments. These content developers would need to be knowledgeable in the subject matter they are developing materials for, and they would need to be trained on the platform's content management tools and processes.

Support staff: The platform would likely need customer support staff who are responsible for assisting users with any issues or questions they have about the platform. These staff would need to be trained on the platform's features and tools, as well as any relevant customer service skills and protocols.

Instructional designers: The platform could also benefit from instructional designers who are responsible for designing and implementing effective learning experiences on the platform. These instrumental designers would need to be knowledgeable in instructional design principles and techniques, and they would need to be trained on the platform's features and tools for creating and delivering learning content.

Banking Domain Knowledge: Provide domain-specific training to familiarize the team with banking regulations, compliance standards, and financial processes (payments, deposits, dues, installments).

Security Standards: In-depth training on security standards relevant to management systems, including data encryption, access controls, and authentication protocols.

11.RESponsibilities

	TM	PM	Dev. Team	Test Team	Client
Acceptance test Documentation & Execution	X	X		X	X
System/Integration test Documentation & Exec.	X		X	X	
Unit test documentation & execution	X		X	X	
System Design Reviews	X	X	X	X	X
Detail Design Reviews	X	X	X	X	
Test Procedures and rules	X	X	X	X	
Screen & Report Prototype reviews			X	X	X
Change control and Regression testing	X	X	X	X	X

12. TESTING SCHEDULE

Dashboard

Tasks

Gantt & Reports

Documents

Milestones

Timesheet

Issues

Users

...

Switch to new version

Welcome back!

All Open

Classic

Add Task

...

TASK

OWNER

STATUS

TAGS

START DATE

DUE DATE

DURATION

PRIORITY

CREATED BY

% COMPLE...

Testing

SM1-T5

Unit Testing

Unassigned

Open

-

12-23-2023

12-25-2023

3 days

None

Dark ANGEL

0%

SM1-T7

System Testing

Unassigned

Open

-

12-26-2023

12-28-2023

3 days

None

Dark ANGEL

0%

SM1-T8

Acceptance Testing

Unassigned

Open

-

12-29-2023

12-31-2023

3 days

None

Dark ANGEL

0%

Test planning

SM1-T2

Design

Unassigned

Open

-

12-15-2023

12-17-2023

3 days

None

Dark ANGEL

0%

SM1-T3

Defining Test Object

Unassigned

Open

-

12-18-2023

12-20-2023

3 days

None

Dark ANGEL

0%

SM1-T4

Test Planning

Unassigned

Open

-

12-21-2023

12-22-2023

2 days

None

Dark ANGEL

0%

General

SM1-T1

Documentation

Dark ANGEL

Open

-

12-12-2023

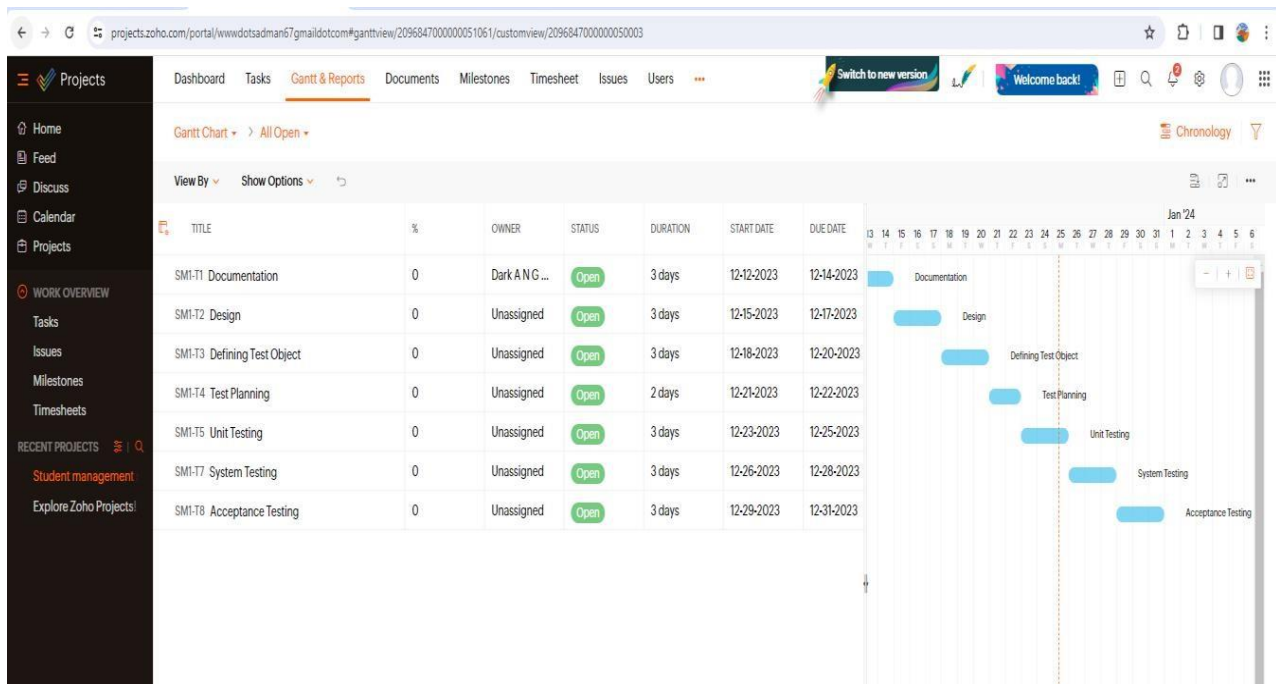
12-14-2023

3 days

High

Dark ANGEL

0%



13.PLANNING RISKS AND CONTINGENCIES

- Limited Reassigned Maintenance staff: The Reassigned Maintenance staff currently has two positions unfilled. As a result of this staff shortage there may be delays in getting staff to review appropriate documents and to participate in the Acceptance test process. Should client staff become a problem, the appropriate dates for reviews and acceptance testing will slip accordingly. No attempt will be made to bypass any part of the review and testing processes.
- Illness or Injury: Regular medical checkups are arranged for the employees.
- Software Failure: Failure of Main or Back-up Hardware: We will maintain a main and Back-up hardware system, printers and workstations should be serviced and maintained regularly.
- Internal Issue: Chain of command will be conducted strictly.

14.APROVALS

Project Sponsor	N/A
Development Management	Sakib Shadman
EDI Project Manager	Shithi Roy
RS Test Manager	Shakhawat Amin
RS Development Team Manager	Thithy Banik
Reassigned Sales	Shithi Roy
Order Entry EDI Team Manager	Shakhawat Amin