American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science &Technology (FST)

Summer- 22-23

Section: C

Software Quality Assurance and Testing

Student Management System

A Report submitted
By

SN	Student Name	Student ID
1	SHARANNAYA DEY SITHI	19-40811-2
2	A.K.M SHAHRIYAR RAHMAN	19-40787-2
3	TAHMID MOSTAQUE TALHA	18-37162-1
4	HASIBUR RAHAMAN	19-40823-2

Checked By Industry Personnel

Name:

Designation:

Company:

Sign:

Date:

Software Test Plan

for

< Student Management System>

Version 1.0 approved

Prepared by <authors>

<American International University - Bangladesh>

< 26/08-2023 >

Checked By Industry Personnel

Name:

Designation:

Company:

Sign:

Date:

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	26-08-2023	SHARANNAYA DEY SITHI	First Draft
0.2	26-08-2023	A.K.M SHAHRIYAR RAHMAN	Second Draft
0.3	26-08-2023	TAHMID MOSTAQUE TALHA	Third Draft
0.4	26-08-2023	HASIBUR RAHAMAN	Fourth Draft

1. TEST PLAN IDENTIFIER: **SMSV1.0**

2. REFERENCES

- We took the idea from AIUB portal. https://portal.aiub.edu

3. INTRODUCTION

Background to the Problem

A university enrolls a large number of students there they have to do many works like signing up for courses, checking their class schedules, seeing their grades, and looking for course materials. Doing these tasks in the real world would be hard for students, so a better way is to use a student portal system that does these things automatically.

Students encounter challenges when trying to access information about available courses and prerequisites. Manually, it becomes inconvenient to ascertain the necessary prerequisites. Additionally, students frequently require counseling regarding course selection, which necessitates physical visits. In the absence of an online portal, determining faculty availability poses difficulties. Moreover, unexpected class cancellations by faculty members present issues. Through this portal, faculty can promptly communicate cancellations. Absent an online platform, students face genuine predicaments. University notifications demand students to consistently inspect physical notice boards; however, with the online portal, students can access university notices at their convenience.

Solution to the Problem

To address this issue, we've developed a student portal management system that enables students to log in using their unique identification and password. Within the portal, there's a dedicated profile section where students can access their personal information, track their earned credits, and view their academic performance reports.

Moreover, the system incorporates a registration segment allowing students to seamlessly enroll in their preferred courses. An academics section is also available, granting students the ability to review their course outcomes and curriculum details.

Furthermore, the portal encompasses a feature for grade reports, enabling students to examine their semester-based grades and performance within the context of the overall curriculum.

We have added some functions to solve mentioned problems.

Profile section: Student can check their profile and others information.

Notification: Student can check notification of faculty members.

Course registration: Students can enroll them for their courses for upcoming semester.

Course and results: Student can check their course results and grade reports in this section.

REQUEIREMNT SPECIFICATION

3.1 System Features

1. System Login

1.1 The software shall allow students to login with their given username and password

1.2 If the username and/or password has been inserted wrong for more than three times, the random verification code will be generated by the system to retry login.

Priority Level: High
Precondition: user have valid user name and password

2. Dashboard

2.1 User can see their weekly class schedule

Priority Level: Medium
Precondition: user must login to the portal.

3. Course Registration

3.1 User able to see student id, name, cgpa and credit completed

3.2 By Clicking 'Next' should allow the user to select and unselect courses.

3.3 User see registered sections, assessment and confirm registration.

Priority Level: High
Precondition: user must login in the portal

4. Academic

4.1 User shall see courses and results

4.1.1 User shall to see grade mark quizzes of both mid and final term.

4.1.2 User shall download and upload document there

4.2 User shall see registration details Like registered courses, taken credit and registration fees

4.3 User shall see their drop application and the response

4.4 User shall see offered courses

4.5 User shall see my curriculum where they can see their all curriculum of their departmental courses (pre-requisite, overall credits)

Priority Level: Medium
Precondition: user must login to the portal.

5. Grade Reports

5.1 User check current or previous grade report either by curriculum or by semester

Priority Level: Low
Precondition: user must login in the portal

6. Library

6.1 Users can check if any book is available in the university Library

Priority Level: Low
Precondition: user must login

7. Forms

7.1 Users can download essential forms and documents related to their study like transfer-credit-form, change major form, assignment coversheet etc.

Priority Level: Low
Precondition: user must login

8. Profile

8.1 User shall see their profile containing their name and other's information like ID, CGPA, credit, program, department, core, major etc.

Priority Level: High
Precondition: user must login to the portal.

9. Notification

9.1 The software will send notification when new notes or notices are uploaded by teachers.

Priority Level: Medium
Precondition: user must login

10. Change Password

10.1 Users can change their password with their given current password, new password and confirm password

Priority Level: Low
Precondition: user must login

11. Logout

11.1 Users can Logout and get back to login page

Priority Level: High

Precondition: user must login

3.2 System Quality Attributes

- **Usability:** Since usability can be measured in terms of ease of use, the application should be user-friendly. As the system features are made easy to learn as navigation is very simple. It's also Easy for a new or infrequent users to learn to use the system.

- **Maintainability:** Maintainability is the term that defines that how easily the maintenance team can perform their task. The main task of our maintenance team is to solve the bugs, adding something new or changing some features. One of our maintenance programmers shall be able to modify any features with 20 labor hours or less of development effort.

- **Efficiency:** Since efficiency is a major system quality attribute, it is measured in terms of time required to complete any task given to the system. The application has passed with a large scale in efficiency as it is easy to use and also concise.

- **Modularity**: Every system that is developed module wise. In this kiosk system there are many modules added and those are integrated to make the whole system, so that bugs can be easily detected in any module and then the defects have to be solved that particular module. We don't have to think about other modules so it makes the work easy for our tester. In addition of that we can also add new features in our system as it has been developed module wise.

- **Reliability And Correctness:** In software development process there **is** no software is free from defects or minor bugs. Our software gives the exact output what user wants and also the correctness of the kiosk system has been ensured. Our software does not crash randomly

as it has been tested and no false output is generated so it's more reliable to use for our users.

- **Testability:** Testability refers to a term that is measured on how easily the testing team can perform their task. Testability is very much depended on modularity. Since our system is developed module wise so test engineers will have to go through an easy-going process. They don't have to check every module to fix bugs.

- **Flexibility:** The effort required to modify an operational program is called flexibility. A maintenance programmer who has at least one year of experience supporting this product shall be able to make a new copy output available to the product, including code modifications and testing, with no more than 2 hours of labor work.

3.3 System Interface

The system interface where the users will interact with the system's functionality

1. Use Case Diagram:

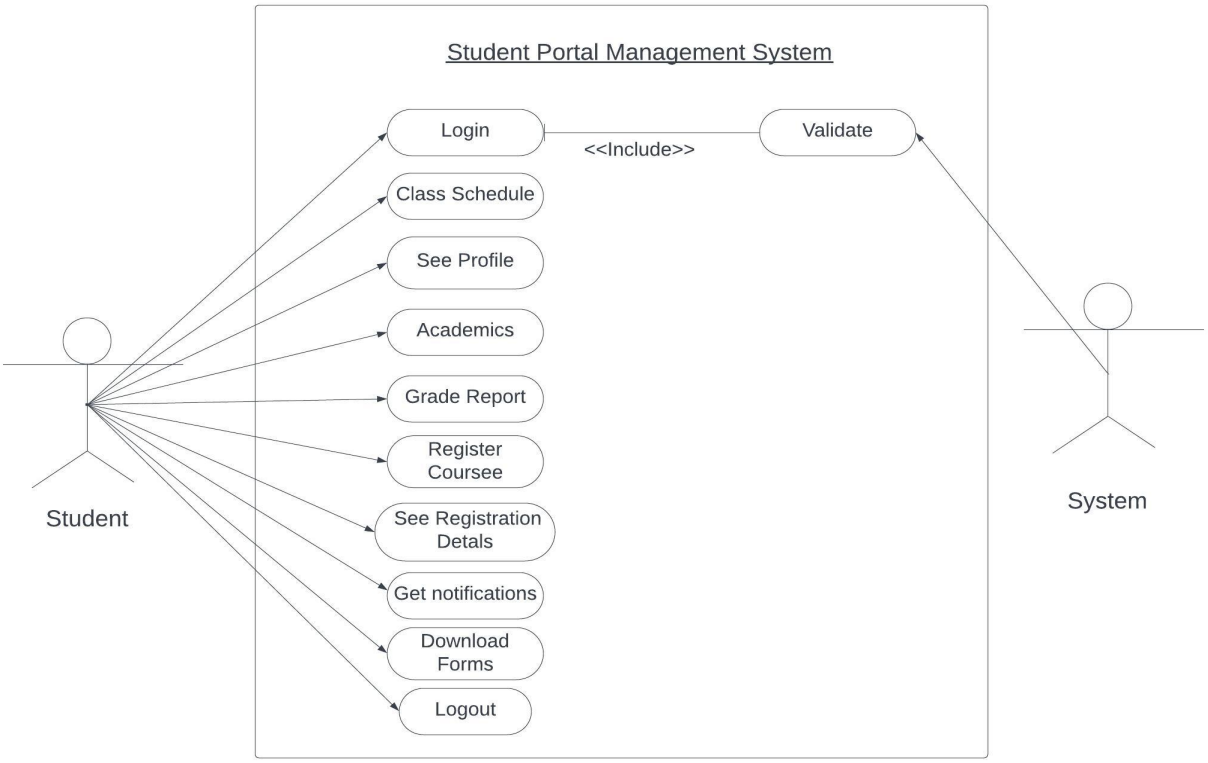
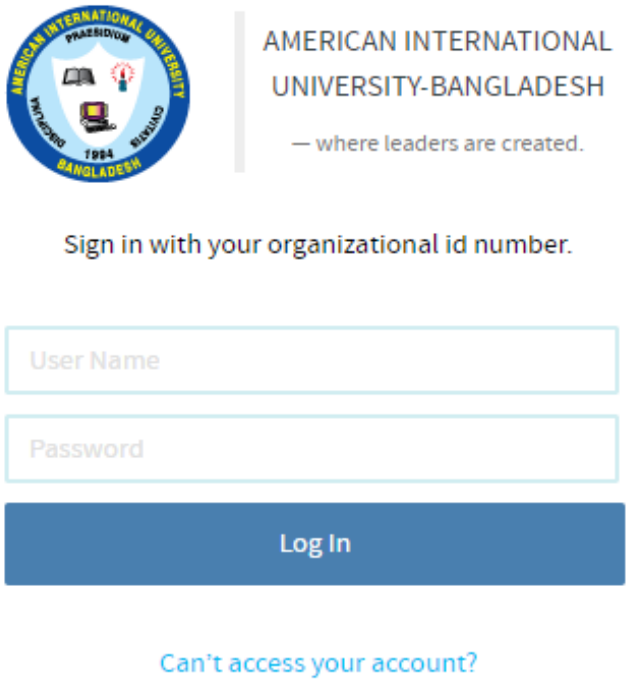
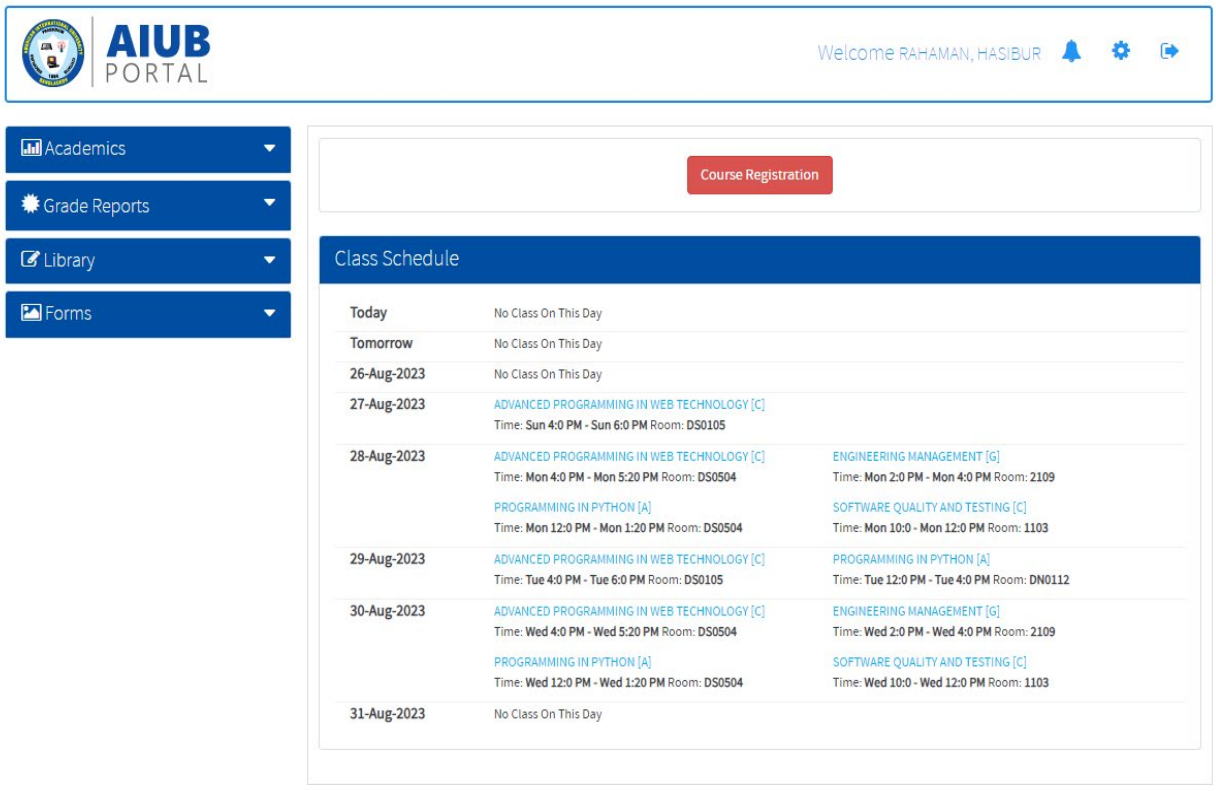


Fig: Use case Diagram

2. System Login:



3. Dashboard:



3.4 Project Requirements

Based on our request, here are some of project requirements that should be considered in project management:

Time: The project should be completed within three months.

Budget: The project should be completed within a predefined budget about 100,000 BDT only

Environmental requirements: The project comply with all environmental regulations and guidelines related to the project, including safety, health, and environmental protection.

Quality: The project meets the required quality standards including performance, reliability, and durability.

4. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts. For example:

- Inside the library feature there are some other features like current borrows, borrow history which has been implemented into the system but not been tested
- There is a messages option where users can check their mail which is implemented into the system but not be tested
- Also, inside the Academic feature there is a section called financials where users can see their financial transactions is implemented in the system but not been tested.
- Then inside Academic feature there is another section called offered courses where there is a search option which is implemented but not been tested because of time shortage.

5. TESTING APPROACH

5.1 Testing Levels

Software testing involves running a program to find any errors or bugs in the code of the software product. The software's usability, scalability, portability, dependability, and usability are all taken into account during this procedure. Software testing's primary objective is to make sure the system and all of its parts agree to the set specifications and function correctly in any situation.

UNIT TESTING

Unit testing is a technique for testing individual software application modules or components. To make sure that the software's separate components are operating as planned, it is normally carried out by developers. Unit tests are typically automated and created to test individual sections of the code, like a certain function or method. Individual pieces of code are tested separately during unit testing, which is the lowest stage of the software development process.

It helps to identify bugs early in the development process, before they become more difficult and expensive to fix. It helps to ensure that changes to the code do not introduce new bugs. It makes the code more modular and easier to understand and maintain.

Working process:

First initializes the system under test, which is just a small part of an application that is being tested, then it provides a stimulus to the system under test (often by invoking a method on it) and finally it watches the behavior that results.

- Select a Unit: Choose the specific unit that you want to test.
- Write Test Cases: Create a set of test cases that cover different scenarios and edge cases for the selected unit. Each test case should consist of input values, expected outputs, and sometimes additional conditions.
- Set Up and Tear Down: If your unit requires any setup or initialization before testing, implement a setup routine. Similarly, if there's cleanup required after testing, create a tear-down routine. These routines help maintain a consistent environment for each test case.
- Execute Tests: Run the unit tests you've created. Most programming languages provide testing frameworks execute each test case and report the results.
- Assertion: compare the actual output of the unit being tested with the expected output. If the assertion fails, it indicates a problem in the unit's implementation.
- Evaluate Results: Review the test results. A successful test case indicates that the unit is functioning correctly for that particular scenario. A failed test case indicates a defect in the unit's code that needs to be fixed.
- Debugging and Fixing: If a test case fails, analyze the code and identify the issue.
- Refactoring: This step encourages code maintenance and improvement.

- Repeat: Aim for comprehensive coverage to ensure that your units are robust and reliable.
- Automation: As the codebase grows, manual testing becomes impractical. Automate the execution of your unit tests using continuous integration tools and practices.

Example-login, Course Registration, Change Password ensure that they work correctly.

Integration Testing

Integration testing is a type of software testing that focuses on analyzing how various software application modules or components interact with one another and share data. Finding any issues or errors that develop when various components are joined and communicate with one another is the aim of integration testing. System testing usually comes after unit testing and comes before integration testing. Early detection and resolution of integration difficulties lowers the likelihood of later, more serious, and expensive concerns. It helps to identify and resolve issues that may arise when different units of the software are combined. It helps to ensure that the different units of the software work together as intended. It helps to improve the overall reliability and stability of the software. It's important to keep in mind that Integration testing is essential for complex systems where different components are integrated together.

working process

- Component Development: Before integration testing can begin, individual software components or modules need to be developed and tested in isolation. Each component is designed to perform a specific function or task.
- Test Planning: The testing team creates a detailed integration test plan. The plan outlines the scope of testing, the components to be integrated, the order of integration, the test environment, and the specific test cases to be executed.
- Integration Strategy Selection: There are several strategies for integrating components, such as top-down, bottom-up, big bang and sandwich/hybrid approaches. The chosen strategy determines the sequence in which components are integrated and tested.
- Top-Down: Integration starts with the highest-level modules, and lower-level modules are gradually added. Stubs may be used for lower-level modules that are not yet available.
- Bottom-Up: Integration starts with the lowest-level modules, and higher-level modules are gradually added. Drivers may be used for higher-level modules that are not yet available.

- Big Bang: All components are integrated simultaneously, and testing is performed on the entire system at once.
- Sandwich/Hybrid: A combination of top-down and bottom-up strategies, aiming for a middle-ground balance between the two.
- Component Integration: Based on the chosen integration strategy, components are integrated into the system incrementally. This involves combining modules and ensuring that data and control flow properly between them.
- Test Case Preparation: Test cases are designed to evaluate the interactions between integrated components. These test cases are derived from the integration test plan and focus on scenarios where different modules collaborate.
- Test Execution: The goal is to identify defects related to the interactions between components, including issues with data flow, communication, synchronization, and more.
- Defect Reporting: When defects are found during testing, they are documented, categorized, and reported to the development team. The development team then works to fix these defects, and the testing process may need to be repeated once the fixes are applied.
- Regression Testing: After defects are fixed, regression testing is performed to ensure that the fixes haven't introduced new issues and that the system still works as intended.
- Test Completion and Reporting: Once all the integration test cases have been executed and defects have been addressed, a test summary report is generated. This report includes details about the tested components, the integration strategy used, the test cases executed, any defects found, and the overall assessment of the system's integration quality.

Example-For Validation and verification used Black Box and White Box Testing.

System Testing

System testing is a type of software testing done on an entire integrated system to determine whether it complies with the necessary requirements. Integration testing successful components are used as input during system testing. Integration testing's objective is to find any discrepancies between the integrated units. System testing finds flaws in the integrated modules as well as the entire system. A component or system's observed behavior during testing is the outcome of system testing. System testing is done on the entire system under the guidance of either functional or system requirement specifications, or under the guidance of both. In general, a testing team that is separate from the development team conducts system testing in order to

objectively assess the system's quality. Functional and non-functional testing are both present. A black-box test is system testing. Prior to the acceptance testing and following the integration testing is system testing. Verifies the system's overall functionality, finds and fixes system-level issues early in the development cycle, aids in validating the requirements and making sure the system satisfies user expectations, and enhances system reliability and quality.

- System Testing Process: System Testing is performed in the following steps:
- Test Environment Setup: Establish testing conditions for better testing.
- Create Test Case: For the testing process, create test cases.
- Create Test Data: The data that will be tested should be created.
- Execute Test Case: Test cases are executed following the generation of the test case and the test data.
- Defect Reporting: Defects in the system are detected.
- Regression Testing: It is carried out to test the side effects of the testing process.
- Log Defects: Defects are fixed in this step.
- Retest: If the test is not successful then again test is performed.

Acceptance Testing

Acceptance Testing is a crucial phase in the software development life cycle where the software is evaluated to determine whether it meets the specified requirements and is ready for deployment. It ensures that the software satisfies the needs of the end-users and stakeholders before it's released into production.

The process of Acceptance Testing generally involves the following steps:

- Requirements Understanding and Preparation: Gather and review the project requirements, including functional and non-functional requirements.
- Test Case Development: Create test cases based on the requirements and use cases. Test cases should cover various scenarios, including typical, edge, and exceptional cases.
- Test Environment Setup: Set up the testing environment that closely resembles the production environment.
- Install the software, configure databases, and other necessary components.
- Test Execution: Run the test cases as per the test plan.
- Defect Reporting and Resolution: If any defects or issues are identified during testing, they are documented in a defect tracking system. Fixed defects are retested to ensure they have been resolved correctly.
- User Acceptance Testing: This phase involves involving actual end-users or stakeholders to test the software.
- Acceptance Criteria Evaluation: The software is evaluated against predetermined acceptance criteria.
- Documentation and Sign-off: Once the software passes all test cases, meets acceptance criteria, and receives approval from users, it's ready for deployment. Stakeholders provide formal sign-off, indicating their satisfaction with the software's quality.
- Deployment and Release: The deployment process should be carefully managed to ensure a smooth transition from testing to production.

In our project we will use Unit testing and Acceptance Testing.

In our projects there are some modules and functions. We can divide the whole project into some modules and pick one by one. We will use dynamic unit testing in our project. We will execute the small unit in order to finding the error. For example, we will take login section first after completing login then we can take dashboard, academics, course registration. We will test one by one.

After completing Unit Testing, we will take User Acceptance testing. It will be from end user. User will use the system; they will check all functions and modules. If they face any bugs or glitch, they will report us.

5.2 Test Tools

We have done all the software tests manually. So, no external tools have been used for this project.

5.3 Meetings

The test team will meet once in every week to evaluate progress to date and to identify error trends and problems as early as possible. The test team leader will meet with development and the project manager once every two weeks as well. These two meetings will be scheduled on different weeks. Additional meetings can be called as required for emergency situations.

Meeting	Developer Team	Testing Team
01	28-06-23	
02		04-07-23
03	11-07-23	
04		18-07-23
05	25-07-23	
06		02-07-23

6. TEST CASES/TEST ITEMS

6.1 **Test case-01**

Project Name: Student Management System			Test Designed by: HASIBUR RAHAMAN	
Test Case ID: SMSV1.0			Test Designed date: 26/08/2023	
Test Priority (Low, Medium, High): High			Test Executed by:	
Module Name: Login Session			Test Execution date:	
Test Title: Verify login with valid username and password				
Description: Test website login page				
Precondition (If any): User must have valid username and password and visit Index (aiub.edu)				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)

1. Go to the website 2. Enter username 3. Enter password 4. Click "Login"	Username: 19-40823-8 Password: 12345	Show message "Invalid username or password. Make sure you typed the user ID assigned to you by your organization. It usually looks like XX-XXXXX-X or XXXX-XXX-X. And check to make sure you typed the correct password."		
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

6.2 **Test case-02**

Project Name: Student Management System		Test Designed by: HASIBUR RAHAMAN		
Test Case ID: SMSV1_02		Test Designed date: 26/08/2023		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Login Session		Test Execution date:		
Test Title: verify login with invalid password.				
Description: Test website login page				
Precondition (If any): User must have valid username and password and visit Index (aiub.edu)				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Enter username 3. Enter password 4. Click Login	Username: 19-40823-2 Password: Random	Show message "Invalid username or password. Make sure you typed the user ID assigned to you by your organization. It usually looks like XX-XXXXX-X or XXXX-XXX-X. And check to make sure you typed the correct password."		
Post Condition: None				

6.3 **Test case-03**

Project Name: Student Management System	Test Designed by: HASIBUR RAHAMAN
Test Case ID: SMSV1_03	Test Designed date: 26/08/2023
Test Priority (Low, Medium, High): High	Test Executed by:

Module Name: Click on "Can't access your account?"			Test Execution date:	
Test Title: Clicking on "Can't access your account?"				
Description: Test the SMSV1 portal.				
Precondition (If any): User must have valid username and visit Index (aiub.edu)				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Clicking on "Can't access your account?" 2. Check the title of the page	Username: 19-40823-2	Brings to a page titled "Reset Password" with information of captcha.		
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

6.4 **Test case-04**

Project Name: Student Management System			Test Designed by: Hasibur Rahaman	
Test Case ID: ST_04			Test Designed date: 26/08/2023	
Test Priority (Low, Medium, High): High			Test Executed by:	
Module Name: Homepage Session			Test Execution date:	
Test Title: verify login with valid username and password.				
Description: Test website Homepage				
Precondition (If any): User must have valid username and password and visit Index (aiub.edu)				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the website 2. Enter username 3. Enter password 4. Click Login	Username: 19-40823-2 Password: 12345	Brings to a page titled "SMSV1 PORTAL" with information about academics, grade reports, library, others, messages, notifications, settings and class schedules		
Post Condition: User is validated with database and successfully get into the homepage. The account session details are logged in the database.				

6.5 **Test Case-05**

Project Name: Student Management System	**Test Designed by:** TAHMID MOSTAQUE TALHA

Test Case ID: SMSV1_05			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Medium			**Test Executed by:**	
Module Name: Courses & Results			**Test Execution date:**	
Test Title: Check Courses & Results in Aiub student portal				
Description: Test the Courses & Results				
Precondition: Should be logged in Homepage (https://portal.aiub.edu/Student)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Courses & Results' in Homepage	None	Should contain a title 'Grade marks Quizes' and have Courses , Semesters, information about individual course result of the user		
Post Condition: User is validated with database and provided marks which is illustrated successfully				

6.6 **Test Case-06**

Project Name: Student Management System			**Test Designed by:** TAHMID MOSTAQUE TALHA	
Test Case ID: SMSV1_06			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Medium			**Test Executed by:**	
Module Name: Registration			**Test Execution date:**	
Test Title: Check Registration in Aiub student portal				
Description: Test the Registration				
Precondition: Should be logged in Homepage (https://portal.aiub.edu/Student)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Registration' in Homepage	None	Should contain a title 'Registration' along with the information of registered courses and fees of particular semester		
Post Condition: User is validated with database and printout paper comes out with user information of courses and fees.				

6.7 **Test Case-07**

Project Name: Student Management System			**Test Designed by:** TAHMID MOSTAQUE TALHA	
Test Case ID: SMSV1_07			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Medium			**Test Executed by:**	
Module Name: Grade Report			**Test Execution date:**	
Test Title: Check Grade Report in SMSV1 student portal				
Description: Test the Grade Report				
Precondition: Should be logged in Homepage (https://portal.aiub.edu/Student)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Grade Report' in Homepage	None	Should contain a title 'Curriculum Grade Report', have a print option and information about Core and Elective Curriculum of the user		
Post Condition: None				

7.8 **Test Case-08**

Project Name: Student Management System			**Test Designed by:** TAHMID MOSTAQUE TALHA	
Test Case ID: SMSV1_08			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): High			**Test Executed by:**	
Module Name: Welcome (USER NAME)			**Test Execution date:**	
Test Title: Check Welcome (USER NAME) in SMSV1 student portal				
Description: Test the Welcome (USER NAME)				
Precondition: Should be logged in as RAHMAN, MAHBUBUR at https://portal.SMSV1.edu/Student				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Welcome RAHMAN, MAHBUBUR' in Homepage	None	Should contain a title 'RAHMAN, MAHBUBUR' and have information about logged student, a picture of logged student		
Post Condition: None				

6.9 **Test Case-09**

Project Name: Student Management System			**Test Designed by:** A.K.M SHAHRIYAR RAHMAN	
Test Case ID: SMSV1_09			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Low			**Test Executed by:**	
Module Name: Academics			**Test Execution date:**	
Test Title: Check Academics in Aiub student portal				
Description: Test Academics				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Academics'	None	Should contain a list of Courses & Results, Registration, Drop Application, Offered Courses, My Curriculum and Financials.		
Post Condition: Upon clicking any option, information of respective students are shown that are related to database.				

6.10 **Test Case-10**

Project Name: Student Management System			**Test Designed by:** A.K.M SHAHRIYAR RAHMAN	
Test Case ID: SMSV1_10			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Medium			**Test Executed by:**	
Module Name: Grade Reports			**Test Execution date:**	
Test Title: Check Grade Reports in Aiub student portal				
Description: Test Grade Reports				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Grade Reports'	None	Should contain By Curriculum and By Semester		
Post Condition: User is validated with database and the grade report of respective students is shown.				

6.11 **Test Case-11**

Project Name: Student Management System			**Test Designed by:** A.K.M SHAHRIYAR RAHMAN	
Test Case ID: SMSV1_11			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Low			**Test Executed by:**	
Module Name: Library			**Test Execution date:**	
Test Title: Check Library in Aiub student portal				
Description: Test Library				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Library'	None	Should contain Search Book, Current Borrows, Borrow History, Financial, UGC Digital Library, SAGE, Research4Life		
Post Condition: User is validated with database and current/past borrowing of books is listed.				

6.12 **Test Case-12**

Project Name: Student Management System			**Test Designed by:** A.K.M SHAHRIYAR RAHMAN	
Test Case ID: SMSV1_12			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Low			**Test Executed by:**	
Module Name: Others			**Test Execution date:**	
Test Title: Check Others in Aiub student portal				
Description: Test Others				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Others'	None	Should Applications, Parking Applications and Download Forms		
Post Condition: User is validated with database and downloaded forms, applications can be accessed.				

6.13 **Test Case-13**

Project Name: Student Management System			**Test Designed by:** Sharannaya Dey Sithi	
Test Case ID: SMSV1_13			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Low			**Test Executed by:**	
Module Name: Messages			**Test Execution date:**	
Test Title: Check Messages in Aiub student portal				
Description: Test Messages				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Messages'	None	Should contain Mail Box		
Post Condition: User is validated with database and messages in inbox, sent, draft and trash are shown.				

6.14 **Test Case-14**

Project Name: Student Management System			**Test Designed by:** Sharannaya Dey Sithi	
Test Case ID: SMSV1_14			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Medium			**Test Executed by:**	
Module Name: Notifications			**Test Execution date:**	
Test Title: Check Notifications in Aiub student portal				
Description: Test Notifications				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Notifications'	None	Should list down all the notes and notices sent by faculties		
Post Condition: User can see other notifications through 'See All'				

6.15 **Test Case-15**

Project Name: Student Management System			**Test Designed by:** Sharannaya Dey Sithi	
Test Case ID: SMSV1_15			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): Low			**Test Executed by:**	
Module Name: Settings			**Test Execution date:**	
Test Title: Check Settings in Aiub student portal				
Description: Test Settings				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Settings'	None	Should contain Change Password		
Post Condition: User can change password				

6.16 **Test Case-16**

Project Name: Student Management System			**Test Designed by:** Sharannaya Dey Sithi	
Test Case ID: SMSV1_16			**Test Designed date:** 26/08/2023	
Test Priority (Low, Medium, High): High			**Test Executed by:**	
Module Name: Logout			**Test Execution date:**	
Test Title: Check Logout in Aiub student portal				
Description: Test Logout				
Precondition: Must log in with valid user id and password. Should log in Homepage Index (aiub.edu)				
Test Steps	**Test Data**	**Expected Results**	**Actual Results**	**Status (Pass/Fail)**
1. Click on 'Logout	None	Should log out the user from the portal and bring back to login page		
Post Condition: None				

7. ITEM PASS/FAIL CRITERIA

For the Student Management System, we intend to use 16 test cases. Unit testing will involve the execution of test cases, and based on how well they go, we will label the test as passed or failed. If there are any problems, we will repair them. We will fix the problems causing it if the testing is as successful as possible given the possibility that certain test cases may fail.

8. TEST DELIVERABLES

- Acceptance test plan
- System/Integration test plan
- Unit test plans/turnover documentation
- Screen prototypes
- Report mock-ups
- Defect/Incident reports and summaries
- Test logs and turnover reports

9. STAFFING AND TRAINING NEEDS

This section shows how to staff the test jobs and prepare them for the work. Staffing is set for the duration of the project. It's realistic to assume that the vast majority of the staff will agree to do some testing. The following occupations are recognized:

Project Manager: Responsible for maturing the complete execution of the Web website. This includes creating requirements, managing the seller relationship, and overseeing the testing cycle.

Test Manager: Responsible for fostering the expert test strategy, examining the test deliverable, dealing with test cycles, collecting measurements and reporting progress to the Project Manager, and recommending when testing should be completed.

Test Engineer: Planning tests, creating test methods, creating test information, running tests, preparing occurrence reports, examining episodes, writing mechanized test strategies, and detailing measurements to the test administrator are all responsibilities of this position.

The test manager and test specialists should be familiar with the website development life cycle. Because this project is being developed in a traditional manner, this is a nonexclusive

depiction of Staffing and Training requirements. As a result, the names of conscious people for each project aren't given.

10. RESPONSIBILITIES

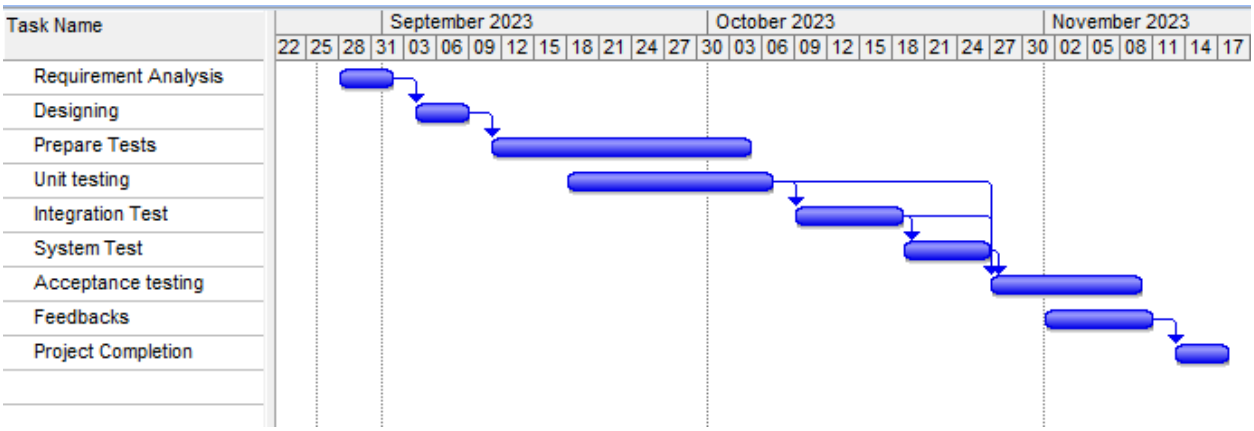
	TM	PM	Dev Team	Test Team	Client
Acceptance test Documentation & Execution	X	X		X	X
System/Integration test Documentation & Exec.	X		X	X	
Unit test documentation & execution	X		X	X	
System Design Reviews	X	X	X	X	X
Detail Design Reviews	X	X	X	X	
Test procedures and rules	X	X	X	X	
Screen & Report prototype reviews			X	X	X
Change Control and regression testing	X	X	X	X	X

11. TESTING SCHEDULE

Time has been allocated within the project plan for the following testing activities. The specific dates and times for each activity are defined in the project plan timeline. The persons required for each process are detailed in the project timeline and plan as well.

ID	Task	Duration (Days)	Responsible
1	Requirement Analysis	5 days	Project Team
2	Designing	5 days	Project Manager Designing Team
3	Prepare Tests	18 days	Developer team
4	Unit testing	15 days	Developer team
5	Integration Test	8 days	Developer team

6	System Test	6 days	Project Manager & Test Engineer
7	Acceptance testing	10 days	Test Engineer & Potential Users
8	Feedbacks	8 days	Potential Users
9	Project Completion	5 days	Project Manager



12. PLANNING RISKS AND CONTINGENCIES

Planning risks and contingencies is a crucial part of project management to ensure the successful execution of a project. The risks and contingencies of portal management systems are defining below:

- University portal might be utilized to give data about the student's unique activities and events, course information, schedules, scholarly resources and contact data.
- If a large number of users use the system at the same time, then the server can be slowdown.
- Cost is another risk for quality software, we need a stable budget
- can this software achieve the specific goal or scope.

13. APROVALS

Project Sponsor	SMSV1
Development Manager	Tahmid
EDI Project Manager	Shahariyar
RS Test Manager	Sithi
Rs Development	Hasib
Re assign Sales	Talha
Order Entry EDI Team Manager	Dip