Department of CIS

TITLE OF THE PROJECT

Integrated University Management System (IUMS)

Submitted By

Tajul Islam Refath

ID: 192-16-447

Supervised By

Md. Mehedi Hassan

Lecturer

Department of Computing & Information System (CIS)

Daffodil International University

Course Code: CIS499

Section: A

Spring-2023

APPROVAL

This Project titled **"Integrated University Management System (IUMS)"**, Submitted by **Tajul Islam Refath,** ID No:**192-16-447** to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on- 19-07-2023.

BOARD OF EXAMINERS

Mr. Md Sarwar Hossain Mollah — **Chairman**
Associate Professor and Head
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Mr. Md. Mehedi Hassan — **Internal Examiner**
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

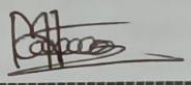
Mr. Syed Tangim Pasha — **Internal Examiner**
Lecturer
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Dr. Saifuddin Md. Tareeq — **External Examiner**
Professor & Chairman
Department of Computer Science and Engineering
University of Dhaka, Dhaka

Declaration

I hereby declare that; this project has been done by me under supervision of Md. Mehedi Hassan, Lecturer, department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

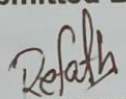
Supervised By



Md. Mehedi Hassan

Lecturer
Department of CIS
Daffodil International University

Submitted By



Name: Tajul Islam Refath
ID: 192-16-447
Department of CIS
Daffodil International University

Abstract

The Integrated University Management System (IUMS) is a comprehensive software solution designed to streamline and enhance the management and administration processes within a university or educational institution. It serves as a centralized platform that integrates various academic, administrative, and financial functions, enabling efficient information flow and data management. The primary goal of IUMS is to automate and optimize the diverse operations and workflows that occur within a university, ranging from admissions and registration to course management, grading, and financial transactions. By leveraging advanced technologies, IUMS eliminates manual paperwork, reduces redundancy, and enhances overall productivity and effectiveness. IUMS offers a wide array of modules and features that cater to different stakeholders within the university ecosystem. Students can use the system to apply for admission, register for courses, access academic resources, view grades, and communicate with faculty and staff. Faculty members can utilize IUMS for course planning, scheduling, grading, and communicating with students. Administrative staff can manage admissions, generate reports, handle financial transactions, and maintain records through the system. The system's architecture is designed to be highly scalable, adaptable, and secure. It employs a robust database management system to store and retrieve data efficiently, ensuring data integrity and availability. Additionally, IUMS incorporates role-based access control mechanisms to enforce data privacy and restrict access to sensitive information. IUMS also enables seamless integration with external systems and services, such as learning management systems, library management systems, and financial systems, allowing for a unified and cohesive technological environment within the university. The implementation of IUMS offers several benefits to the university. It simplifies and accelerates administrative processes, reduces errors and redundancies, improves communication and collaboration among stakeholders, enhances data analysis capabilities, and provides real-time access to critical information. These advantages contribute to the overall efficiency, transparency, and competitiveness of the institution.

In conclusion, the Integrated University Management System (IUMS) serves as a comprehensive solution to address the complex management and administrative challenges faced by universities. It revolutionizes the way universities operate by integrating various functions, streamlining processes, and improving communication, ultimately leading to enhanced productivity and effectiveness in the academic environment.

Table of Contents

List of Figure

List of Table

Chapter 1 – Introduction

1.1 Introduction

The Integrated University Management System (IUMS) is a comprehensive software solution designed to streamline and automate various administrative and academic processes within a university. It serves as a centralized platform that integrates multiple modules and functionalities to enhance efficiency, data accuracy, and communication across different departments and stakeholders.

IUMS is tailored specifically for universities, providing a wide range of features that cover various aspects of university management, including student information management, faculty management, course management, admissions and enrollment, examination and grading, financial management, and more. By digitizing and automating these processes, IUMS aims to optimize administrative tasks, improve decision-making, and enhance the overall educational experience for students, faculty, and staff.

The primary objective of implementing IUMS is to replace traditional paper-based systems and manual processes with a cohesive and user-friendly digital solution. This system allows for seamless data integration and real-time access to information, ensuring that accurate and up-to-date data is available at all times. Furthermore, IUMS enables effective communication and collaboration among various stakeholders, fostering a more efficient and productive university ecosystem.

1.2 Document Contents in Project Document

This book or documentation, which will track the development of the project, will include the following chapters.

Chapter 1: Introduction

An overview of the proposed project and system.

Chapter 2: Initial Phase

The key goals and objectives, problem area, potential solutions, and project history are all covered in this chapter along with the preliminary research findings for the proposed system.

Chapter 3: Literature Review

This chapter goes into great detail on the problem domain, solutions, evaluation of existing solutions, and final suggestion.

Chapter 4: Methodology

This section will cover the value of using methodology, several ways that can be used, the best methodology, and how to apply it.

Chapter 5: Planning

Project plans, including test plans, risk and change management, and others, are covered in this chapter.

Chapter 6: Feasibility

This is where you'll find the full feasibility study report and cost-benefit analysis.

Chapter 7: Foundation

The issue area identification, general need list, proposed technology, and justifications will all be covered in this chapter.

Chapter 8: Exploration

It includes fundamental UML diagrams, a set of requirements for both the new and old systems, a prototype, and more.

Chapter 9: Engineering

The logical and behavioral models of the suggested system are presented in this chapter.

Chapter 10: Deployment

Here, we'll discuss coding examples and how to divide a development problem into manageable components based on development priorities.

Chapter 11: Testing

This chapter contains a variety of test concepts and results.

Chapter 12: Implementation

The implementation approach, training model, and other relevant subjects are covered in this section.

Chapter 13: Critical Appraisal and Evaluation

The detailed examination of the initial goals that were achieved and those that weren't.

Chapter 14: Lessons Learned

Most of the project's lessons learned and challenges are included in the pre-project-closing review.

Chapter 15: Conclusion

You can see a summary of the project's objectives, accomplishments, and lessons learned here.

Chapter 2 – Initial Study

2.1 Project proposal

In today's world, the solution to all human problems is at hand. The world has become internet based. Internet or online based work is now very easy to do as compared to previous paper-based work. With time, educational institutions and online education programs should progress towards online based work. Use the Integrated University Management System (IUMS) to manage educational programs, challenges and solutions to educational program management. So, the digital and online based system of education in educational institutions or online is Integrated University Management System (IUMS). It is a completely different process for educational institutions, students, teachers, where all educational activities can be done in an automated manner. It will be usable only for teaching purposes of the educational institution.

The Integrated University Management System (IUMS) was created to help develop education in the country. Online based education across the country is designed to solve problems and streamline and automate various administrative tasks and processes within a university that will be done entirely over the Internet. That is, the Integrated University Management System (IUMS) system allows all educational programs to be completed through the Internet from anywhere in the country.

The improvement of Bangladesh's educational system is the goal of this project. Create progressively more opportunities. This project will develop a system that will connect our educational institutions, teachers, and students in order to address the challenges that teachers have when they are trying to teach. In other words, students do not need to go to a teacher and borrow resources from him in order to solve the problem. He can access this IUMS system at home to access all of his academic information, including how much his tuition is. The management team of the educational institution also doesn't need to pull out a piece of paper to examine all the student information; instead, they may access it all via the IUMS automation system.

Background Study:

Although the education program in our country has been doing modernization or automation for a long time, that modernization may not have been done across the country. So still most of the universities of our country are still not able to manage their institutions through automation systems. So when I did my internship (Daffodil Computers Limited) a university from Sylhet wanted to manage their RTM-AKTU University in the automation process. So we have developed the Integrated University Management System (IUMS).

But while developing this system we have seen some automation IUMS but we have found some weak points which we have successfully implemented in our system.

The universities I researched are:

Southeast University https://admission.seu.edu.bd/, Northern University http://182.160.97.196:8088/admission/ and Pundra University of Science and Technology https://pundrauniversity.edu.bd/admissions/student/ student_online_admission.

I have thoroughly researched all the above universities and found some weak parts or flaws in them.

Some of the weak points I noticed in their automation IUMS is in their admission process. If I talk about the admission system in Southeast University, they have all kinds of user satisfaction and form filling system for admission following proper guidelines, but their weak point is that they have a part in the system even if the student is behind in any department in his HSC. He can apply for all departments while applying. But this is a weak point for them.

Then if I talk about the automation system of Northern University then they only have some forms on the website for the admission of students for some information. They have no automation system.

If it can be noticed in the whole Pundra University of Science and Technology then they don't have automation but as much as they have done, students can apply for admission in any department after hsc in any department like Southeast University.

If all the above aspects are considered by background study. So the systems of Southeastern University, Northern University and Pundra University of Science and Technology do not have all the features required for the IUMS automation process.

Description of the proposed system

A comprehensive software solution aimed at streamlining and automating numerous administrative procedures within a university or educational institution is known as an integrated university management system automation. It strives to deliver a seamless experience for students, teachers, staff, and administrators by centralizing and integrating data, increasing efficiency, enhancing communication, and improving efficiency.

The system typically consists of modules that cover various areas of university management, including:

- Student Information System: This module manages student records including admission, enrollment, academic progress, grades, attendance and personal information. It enables students to access their profile, register for courses, view schedules and track their academic progress.
- Course Management: This module facilitates course creation, scheduling and management. It allows faculty members to publish course content, track student performance, manage assignments and assessments, and communicate with students.

- Financial Administration: This module manages financial processes, such as tuition fees, scholarships, grants and student billing. It enables online payment processing, tracks financial transactions, generates invoices and provides financial reporting capabilities.
- Human Resource Management: This module manages faculty and staff information including recruitment, contracts, payroll, and attendance and leave management. It helps track employee performance, manage benefits and generate reports for HR-related activities.
- Library Management: This module automates library functions including cataloging, circulation, and book storage and inventory management. It provides a searchable database of assets, enables online borrowing and renewal and tracks overdue items.
- Online Learning: This module supports online learning and course delivery. It has features like virtual classroom, discussion forum, assignment submission, online quizzes and grading tools.
- Reporting and Analysis: This module provides reporting and data analysis capabilities to generate insights and monitor key performance indicators. It enables administrators to track enrollment trends, student performance, financial metrics, and other relevant data for informed decision making.
- Communication and Collaboration: The system includes features to facilitate communication and collaboration among students, faculty, staff, and administrators. This may include email integration, messaging platforms, discussion boards and announcement systems.

Integrated university management system automation aims to eliminate manual processes, reduce paperwork, reduce errors and improve data accuracy and security. It enables seamless information flow between different departments and stakeholders, improving overall performance and user experience. Additionally, the system provides administrators with centralized control and real-time access to data, empowering them to make data-driven decisions and streamline university operations.

By implementing an integrated university management system automation, educational institutions can improve administrative processes, improve student services, increase productivity and ultimately create a more efficient and effective learning environment.

Prioritized Features according to MoSCoW

During the requirement analysis Stage, the MoSCoW technique is utilized to prioritize requirements. It facilitates understanding the significance of the requirements.

Serial No.	**Requirement for IUMS**	**Priority of IUMS**
01.	User Authentication and Access Control	Must-have
02.	Student Management	Must-have
03.	Course Management	Must-have

04.	Financial Management	Must-have
05.	Faculty and Staff Management	Must-have
06.	Academic Advising	Should-have
07.	Online Registration and Enrollment	Should-have
08.	Grade Management	Should-have
09.	Reporting and Analytics	Should-have
10.	Communication and Notifications	Should-have
11.	Learning Management System Integration	Could-have
12.	Alumni Management	Could-have
13.	Research Administration	Could-have
14.	Resource Booking	Could-have
15.	Campus Facilities Management	Won't-have
16.	Human Resources (HR) Management	Won't-have
18.	E-commerce and Online Payments	Won't-have

Table 1: Prioritized Features

Exploration & Engineering

The final criteria are based on the Prioritized Requirements List (PRL).:

The Exploration & Engineering Integrated University Management System (IUMS) is a hypothetical system that could be used by universities to streamline and manage various aspects of their operations. While there is no specific existing system with this name, I can provide a general overview of what such a system might entail based on common functionalities found in university management systems.

- Student Information Management: IUMS would include modules to manage student information, such as admissions, registration, course selection, grades, attendance, and personal details. It would provide a centralized database for storing and retrieving student data.
- Course Management: The system would facilitate the management of courses offered by the university. It would include features like course scheduling, assignment and assessment management, and tracking of course-related resources and materials.
- Faculty Management: IUMS would enable the management of faculty-related information, including faculty profiles, teaching assignments, research activities, and evaluation of faculty performance. It could also facilitate communication between faculty members and students.
- Academic Records: The system would maintain comprehensive academic records for each student, including transcripts, degrees awarded, and certificates earned. It would also allow students to access and request official academic documents.
- Financial Management: IUMS would handle financial aspects such as student fees, scholarships, grants, and financial aid. It could provide features for online fee payments, generating financial reports, and tracking budget allocations.
- Library Management: The system would have modules to manage library resources, including cataloging, circulation, and online access to e-books, journals, and databases. It could facilitate book reservations, inter-library loans, and book renewal processes.
- Campus Facilities Management: IUMS could include features to manage campus facilities like classrooms, laboratories, and equipment. It would allow scheduling and booking of facilities, tracking maintenance requests, and monitoring resource utilization.
- Communication and Collaboration: The system would facilitate communication among students, faculty, and staff members. It could include messaging platforms, discussion forums, and notifications for announcements, events, and deadlines.
- Reporting and Analytics: IUMS would provide tools for generating reports and analyzing data related to student performance, enrollment trends, resource utilization, and other relevant metrics. This information can help in decision-making and planning.

While the exact features and functionalities of IUMS may vary depending on the specific needs and requirements of a university, the general aim is to provide a comprehensive and integrated platform to streamline and automate various administrative and academic processes.

Iterative development – Timeboxing

The formulation of time estimates should be done iteratively, with tasks being divided into digestible segments. The project will take 50–60% of the total time for the actual development, 15-20% for documentation, and 15-20% for the foundation and feasibility phases. The task will take between 280 and 320 hours to complete and will take three months. The project is divided into seven segments, each lasting 20 to 40 hours. The "t-shirt" method will be used by me to estimate time boxes. To establish tale points, t-shirt sizes XS, S, M, L, and XL will be used. Greater sizes require more time and effort than smaller sizes.

Time Boxing

Timebox	Start Date	End Date	Duration	Tasks/Deliverables
TB1	**01/01/2023**	**01/16/2023**	16	Feasibility & Foundation
TB2	**01/16/2023**	**01/31/2023**	15	Requirements Specification Document
TB3	**02/01/2023**	**02/19/2023**	19	System Architecture Design, Database Design, User Interface Design
TB4	**02/19/2023**	**03/05/2023**	15	Source Code, Configured Databases, Implemented Interfaces
TB5	**03/05/2023**	**03/25/2023**	20	Test Plans, Test Cases, Test Results, Bug Reports
TB6	**03/25/2023**	**04/19/2023**	26	User Manuals, Technical Documentation, Installation Guides
TB7	**04/19/2023**	**04/27/2023**	8	Deployed IUMS on Production Servers
TB8	**04/27/2023**	**05/15/2023**	18	Training Materials, User Support Channels
TB9	**05/15/2023**	**05/30/2023**	15	Migrated Data, Data Migration Reports
TB10	**06/01/2023**	**06/25/2023**	25	Maintenance Reports, Upgraded System Versions
TB11	**01/01/2023**	**06/25/2023**	180	Project Documentation

Table 2: Time Box Estimation

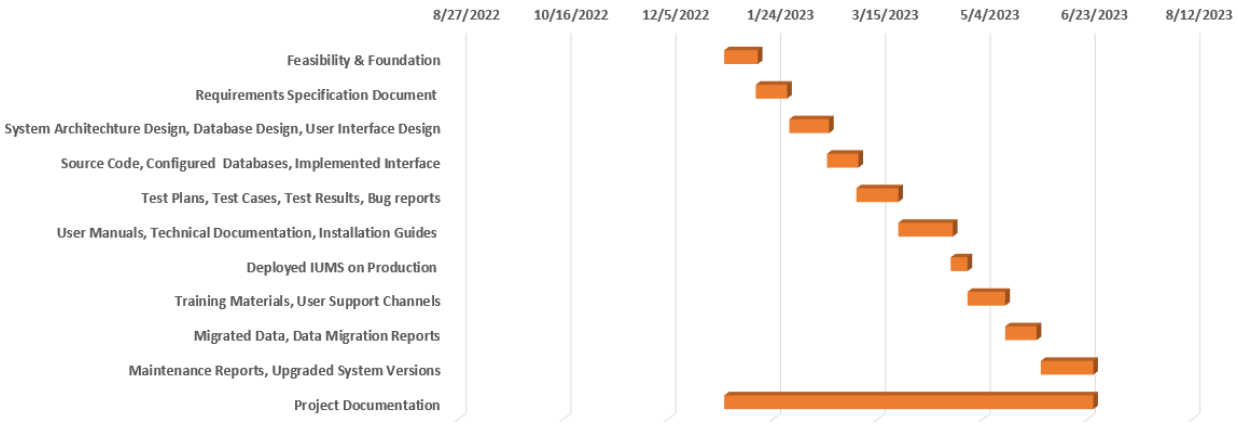


Figure 1: Gantt chart of Proposed Time Box Estimation

2.2 Background of the project

The background of the Integrated University Management System (IUMS) project involves the need for a comprehensive and centralized system to streamline and enhance the management of various administrative and academic processes in a university. Here is a brief overview of the background:

- Existing Challenges: Prior to the IUMS project, the university faced challenges in managing and coordinating various operations manually or through disparate systems. This resulted in inefficiencies, data redundancy, lack of real-time information, and difficulties in decision-making. The university recognized the need for a unified solution to address these challenges.

- Scope and Objectives: The IUMS project aimed to develop an integrated system that would cover essential areas such as student information management, course management, faculty management, financial management, library management, and more. The system's main objectives were to improve operational efficiency, enhance data accuracy and accessibility, enable effective communication, and provide better services to students, faculty, and administrators.

- Requirements Gathering: The IUMS project team conducted thorough requirements gathering activities to understand the specific needs and pain points of the university. This involved interviewing stakeholders, conducting surveys, analyzing existing systems and processes, and considering industry best practices.

- System Design and Development: Based on the gathered requirements, the project team proceeded with system design and development activities. This included defining the system architecture, database design, user interface design, and integration with existing systems or third-party solutions.

- Testing and Deployment: To verify the IUMS's dependability and quality, rigorous testing procedures were used. Testing of several kinds, including user acceptance testing, system testing, integration testing, and unit testing, was done. After testing was finished, the system was implemented gradually to ensure smooth transitions and little interference with regular university activities.

- Training and Adoption: To ensure successful adoption of the IUMS, comprehensive training programs were conducted for faculty, staff, and students. Training materials and user documentation were provided to facilitate system understanding and usage.

- Ongoing Maintenance and Support: Following the successful deployment of the IUMS, an ongoing maintenance and support mechanism was established. This included addressing bug reports, providing system upgrades, and catering to evolving user requirements.

Overall, the IUMS project aimed to revolutionize the university's management processes by implementing a unified and integrated system. By addressing the existing challenges and providing efficient tools and functionalities, the IUMS aimed to enhance the overall academic experience and administrative operations of the university.

2.3 Problem Areas

In Bangladesh, the higher education sector faces several challenges that an Integrated University Management System (IUMS) could potentially address. Here are some common problem areas in university management that could be targeted by an IUMS:

- Manual and Paper-based Processes: Many universities in Bangladesh still rely heavily on manual and paper-based processes for tasks such as student admissions, course registration, fee management, and examination management. This leads to inefficiencies, delays, and increased administrative burden.
- Data Redundancy and Inaccuracy: The lack of a centralized system often results in data redundancy and inaccuracy. Different departments maintain separate databases, leading to inconsistencies and difficulties in data retrieval and analysis.
- Limited Access to Information: Students, faculty, and administrators often face challenges in accessing up-to-date and relevant information. Key information, such as course offerings, schedules, academic resources, and student records, may not be readily accessible.
- Inefficient Communication and Collaboration: Communication and collaboration among stakeholders, such as faculty members, students, and administrators, may be fragmented and inefficient. Traditional modes of communication, like physical notices and manual forms, hinder effective collaboration and timely information sharing.
- Inadequate Academic and Administrative Support: Students and faculty may face difficulties in accessing academic resources, library services, and administrative support. Manual processes for issuing transcripts, certificates, and other academic documents may lead to delays and inconvenience.

Implementing an IUMS can address these problem areas by streamlining and automating processes, centralizing data, improving access to information, facilitating collaboration, and enabling effective monitoring and evaluation. By providing a unified platform for managing academic and administrative

activities, an IUMS can enhance operational efficiency, data accuracy, and overall student and faculty experience in the higher education sector of Bangladesh.

2.4 Possible Solutions

Here are some possible solutions to address the problem areas in university management in Bangladesh through an Integrated University Management System (IUMS):

Manual and Paper-based Processes:

Solution: Implement an IUMS that automates administrative processes such as student admissions, course registration, fee management, and examination management. Online application forms, automated workflows, and electronic document management can streamline these processes, reducing manual work and improving efficiency.

Data Redundancy and Inaccuracy:

Solution: Develop a centralized database within the IUMS that integrates data from different departments and ensures data integrity. Implement data validation checks, standardized data entry formats, and robust data synchronization mechanisms to eliminate redundancy and maintain accurate and up-to-date information.

Limited Access to Information:

Solution: Create a user-friendly web portal or mobile application as part of the IUMS. This platform should provide a centralized hub where students, faculty, and administrators can access relevant information, such as course schedules, academic resources, student records, and university-wide announcements. Personalized dashboards and notifications can keep users informed and facilitate easy information retrieval.

Inefficient Communication and Collaboration:

Solution: Integrate communication and collaboration tools within the IUMS, such as internal messaging systems, discussion forums, and announcement boards. These features enable seamless communication among students, faculty, and administrators, fostering collaboration, information sharing, and timely updates.

Inadequate Academic and Administrative Support:

Solution: Enhance academic and administrative support by providing online access to academic resources, library services, and self-service portals. Students should be able to request academic documents, track their progress, and receive timely support through the IUMS. Additionally, implement helpdesk or ticketing systems to address support queries efficiently.

Implementing these solutions through an IUMS can revolutionize university management in Bangladesh. It streamlines processes, improves access to information, enhances communication and collaboration, provides comprehensive support, enables effective monitoring and evaluation, and facilitates integration with external systems, resulting in a more efficient and effective higher education ecosystem.

Chapter 3 – Literature Review

An integral part of project documentation, a literature review entails a thorough analysis of prior published studies and academic works pertinent to the topic or subject matter of the project. It serves as a basis for the project and exemplifies the relevance and contribution of the work to earlier research. It is a critical study and synthesis of the state of the art in the field. A project literature review assists in identifying issues and potential fixes. This element paves the way for a fruitful project by enabling comparisons within the same field of study as well as prospective research conversations.

3.1 Discussion on the Problem Domain IUMS

Integrated University Management System (IUMS) is a web-based and mobile app that automates the management processes of educational institutions or universities. Course management, financial management, faculty management, library management and resource management can be done through this application. With this application, all the universities of Bangladesh will be able to manage their institutions with their own automation system. Also, since it is an online based system there are some security concerns. Sometimes there are network problems or micro services service delays. Authentication issues in this case are the most common on my system. Management Information System (MIS) is the core component of My System Integrated University Management System (IUMS). Bangladesh University Granting Commission (UGC) has been trying to manage the automation system of all the universities of our country from the beginning. In view of this most of the universities are trying to make all their systems online based or automation. Many universities are successful in implementing this automation IUMS and many universities are yet to succeed. If we say Southeast University (SEU), Northern University (Northern University) and Pundra University of Science and Technology (PUST). Each organization listed above designs their systems in light of their business activities. However, the fundamental problem is that due to the lack of analysis and in-depth testing, there is no correlation between these groups and them.

The main problems associated with university management systems in managed countries are-.

- Limited system capabilities: University management systems may lack certain features or functionalities required to effectively manage all aspects of a university's operations. This can lead to inefficiencies and the need for manual workarounds.
- Lack of customization: Some university management systems may not provide enough flexibility for customization based on the specific needs of a particular institution. This can result in challenges when trying to align the system with existing workflows and processes.
- Integration challenges: Integrating a university management system with other existing systems and databases can be complex. Incompatibility issues may arise, requiring additional resources and efforts to ensure smooth data flow and communication between systems.
- User adoption and training: Implementing a new university management system often requires users, such as administrators, faculty, and staff, to adapt to new interfaces and processes. Insufficient training and support can hinder user adoption and result in resistance to the system.
- Data security and privacy: The personal and academic data of students, professors, and staff is stored and handled through university management systems. To safeguard against unwanted access or data breaches, it is essential to implement strong data security measures, such as encryption and access controls.

It's important to note that these problems are not specific to Bangladesh but are common challenges faced by university management systems globally. To obtain more accurate and up-to-date information on the specific issues associated with university management systems in Bangladesh, it would be advisable to refer to research papers, official reports, or consult professionals with expertise in the field.

3.2 Discussion on the Problem Solution:

However, since this is an online-based problem, it is important to have access to the Internet. The IUMS project focuses on problem solving methods of university management work and all problem-solving methods under the university sector. In this age of technology and globalization, automation is ubiquitous, from the bedside to the office. Soon all universities of Bangladesh will be included in this automation. Every day, technology is changing our industry. Everything is data-driven and computer-based. The solution should consider the following points:

- Comprehensive system evaluation: Conduct a thorough evaluation of the existing university management system to identify its strengths, weaknesses, and areas for improvement. This evaluation should involve feedback from users, administrators, and technical experts.
- Customization and system enhancement: Work closely with the system provider or develop an in-house team to customize the university management system according to the specific needs of the institution. This customization should aim to streamline workflows, automate processes, and enhance user experience.
- Robust integration and data management: Ensure seamless integration of the university management system with other existing systems and databases, such as finance, HR, and student information systems. This integration should facilitate efficient data exchange and eliminate redundant data entry.
- User training and support: Provide comprehensive training programs and ongoing support to users, including administrators, faculty, and staff. This training should cover system functionalities, best practices, and troubleshooting techniques. User feedback should be actively sought and addressed.
- Data security and privacy measures: Implement strong data security measures, including encryption, access controls, and regular data backups. Conduct periodic audits to identify vulnerabilities and address them promptly. Ensure compliance with relevant data protection regulations.

These solutions should be tailored to the specific context and requirements of universities in Bangladesh. It is advisable to involve stakeholders, including university administrators, IT professionals, and end-users, in the decision-making and implementation processes to ensure successful adoption and long-term sustainability of the university management system.

Safety and security:

The methods and policies implemented to safeguard the system, its data, and the users from potential threats, illegal access, and data breaches are referred to as the safety and security of an Integrated University Management System (IUMS). An integrated university management system's safety and security considerations are described as follows:

- Data protection: To ensure the confidentiality, integrity, and accessibility of sensitive information, the integrated university administration system should use strong data protection mechanisms.

- User authentication and authorization: To confirm the identity of people logging into the system, the system should include robust user authentication techniques. Multi-factor authentication techniques, such passwords, biometrics, or smart cards, may be used in this situation.
- Secure access and network controls: Access to the integrated university management system should be restricted to authorized personnel and protected through secure network configurations.
- Regular security updates and patches: It is essential to keep the integrated university management system up to date with the latest security patches and software updates.
- Incident response and monitoring: An effective incident response plan should be in place to handle security incidents promptly and effectively.

Trust:

Trust in an Integrated University Management System (IUMS) refers to the confidence and reliance that users, such as students, faculty, staff, and administrators, have in the system's ability to fulfill its intended functions accurately, securely, and reliably. Here is a description of trust considerations in an integrated university management system:

- Data accuracy and reliability: Trust in the IUMS relies on the accuracy and reliability of the data it manages. The system should ensure that data entered, processed, and retrieved is accurate, up-to-date, and reflects the real-time information it represents
- System availability and performance: Users need to trust that the IUMS will be available and accessible when they need it. The system should have robust infrastructure, scalability, and redundancy measures to ensure high availability and smooth performance, even during peak usage periods.
- User experience and usability: Trust is enhanced when the IUMS provides a user-friendly interface and intuitive workflows. The system should be designed with user-centered principles, making it easy for users to navigate, input data, retrieve information, and perform necessary tasks.
- Security and privacy: Trust is closely tied to the security and privacy measures implemented in the IUMS. The system should have robust security controls, encryption mechanisms, access controls, and user authentication mechanisms to protect sensitive data from unauthorized access, breaches, or misuse.
- Transparent data handling: The IUMS should be transparent in how it handles user data. Users should have clear visibility and control over their personal information, including the ability to review, update, and delete their data as needed.

Privacy:

Privacy in an Integrated University Management System (IUMS) refers to the protection of personal information and the rights of individuals regarding the collection, use, storage, and disclosure of their data within the system.

Server Down or 24/7 Availability in IUMS:

Server downtime and 24/7 availability in an Integrated University Management System (IUMS) refer to the system's ability to remain operational and accessible to users at all times, without significant disruptions or outages.

Backup:

Backup in an Integrated University Management System (IUMS) refers to the process of creating copies of data and storing them in a separate location to ensure its availability and recoverability in the event of data loss, system failures, or disasters.

3.3 Comparison among the leading solutions

New technologies are developed to keep up with the times and make living simpler as the environment changes. Thanks to the internet, the world has shrunk to the size of a small town. Numerous approaches can be used to solve the same problem. IUMS is a unique solution, but there are many more programs in the same area. The best qualities, strengths, and shortcomings of a few popular and well-known contemporary websites will be compared and evaluated in this essay. Here are some of them:

- https://admission.seu.edu.bd/ (Bangladesh)
- http://182.160.97.196:8088/admission/ (Bangladesh)
- https://pundrauniversity.edu.bd/admissions/student/student_online_admission (Bangladesh)

Southeast University:

With well-designed dashboards packed with stunning and in-depth modules, UMS aids all Southeast University members in completing a variety of academic and co-curricular responsibilities. (University, 2023)

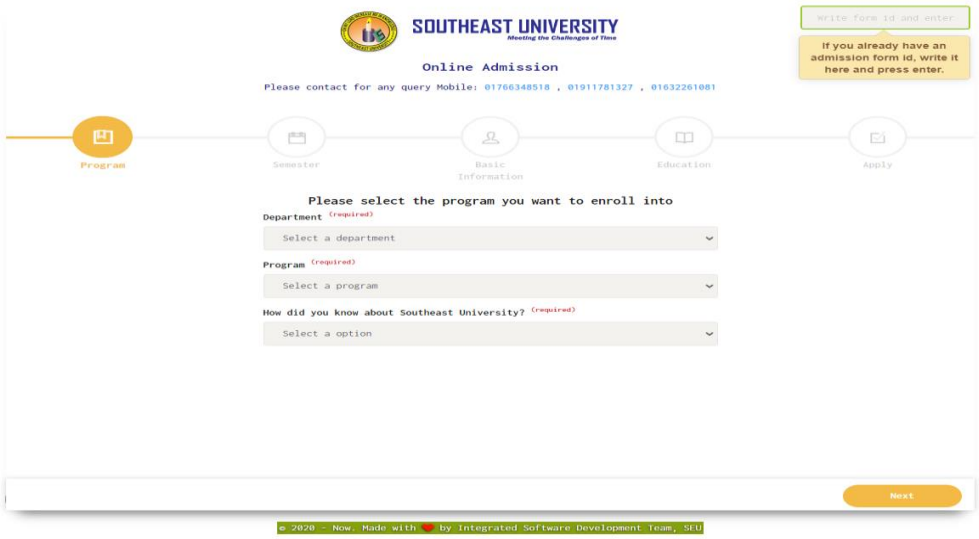


Figure 2: Overview of Southeast University IUMS

Best Features:

- ➢ Southeast UMS allows for efficient management of student records, including personal details, academic performance, attendance, and discipline records.
- ➢ Southeast UMS helps in managing courses, class schedules, and curriculum information. It enables faculty members to update course materials, syllabi, and grading criteria.
- ➢ Southeast UMS simplifies the registration and enrollment process for students, allowing them to select courses, view availability, and sign up for classes online. It may also handle payment processing and fee management.

- ➢ Southeast UMS often includes communication tools such as internal messaging systems, discussion forums, and announcement features to facilitate communication between students, faculty, and administration.

Limitations:

- ➢ A student can apply for admission to any department without university requirements.
- ➢ Implementing a UMS requires significant initial setup, including data migration, customization, and integration with existing systems.
- ➢ UMS heavily relies on internet connectivity. In case of network or internet outages, access to critical information and system functionality may be disrupted, affecting the institution's operations.
- ➢ There is a lack of data security and privacy.
- ➢ The identity of users are not verified.
- ➢ There is no system for processing or notifying requests in real time.

Northern University:

A group of academics now owns and runs Northern University Bangladesh under the illustrious and reputable NUB Trust. They want to develop people with a particular skill set and a sense of social responsibility in addition to offering higher education to deserving students with limited financial resources. In order to give Bangladesh's people a better future, Northern University Bangladesh is dedicated to advancing the nation's socioeconomic empowerment. (history, 2023)

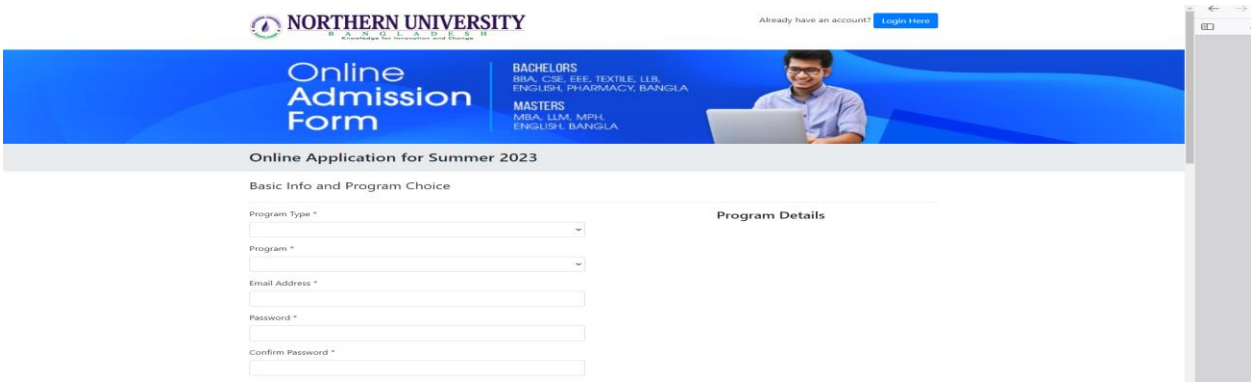


Figure 3: Overview of Northern University UMS

Best Features:

- ➢ The admission form of Northern University is well organized.
- ➢ Northern University carefully compares the board results and checks the eligibility as per the requirements of the university in the admission form.
- ➢ They have included all their technical topics under the ums dashboard.

Limitations:

- ➢ They have to perform system updates, bug fixes and resolve technical issues within the UMS system.

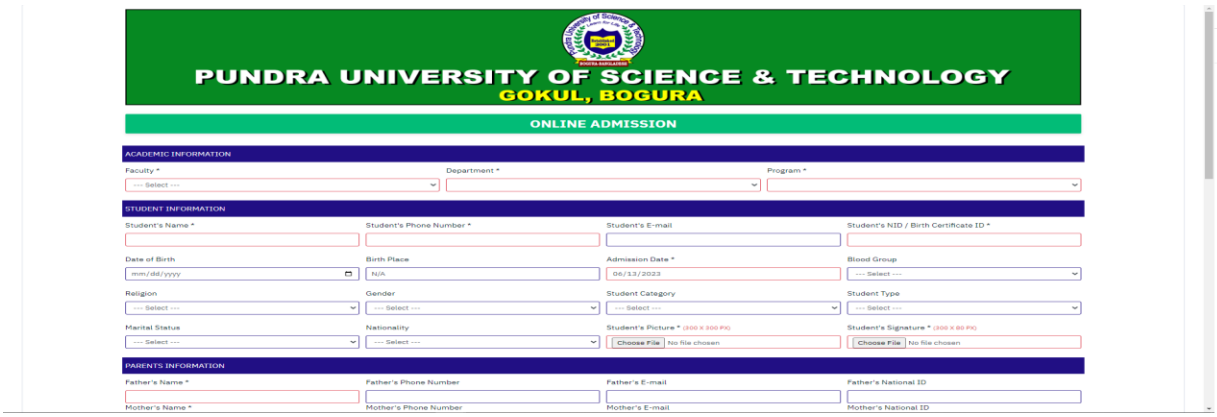


Figure 4: Overview Pundra University of Science and Technology (PUB)

Best Features:

- ➢ Admission form is well organized
- ➢ A very simple admission form can give student data.
- ➢ The whole admission process is very smooth

Limitations:

- ➢ There is no system other than the admission process in their UMS.
- ➢ Also there is a possibility of data theft in the admission process.
- ➢ Major Issue in privacy (University & Student)

3.4 Recommended Approach

The preceding discussion shows that the existing system lacks the required qualities. It also defines the characteristics and problems that the new system must address. The following features should be included in the new system:

- The system's design is appealing and user-friendly.
- IUMS assists in the management of courses, class schedules, and curriculum material. Faculty members can use it to edit course materials, syllabi, and grading standards.
- IUMS enables the efficient management of student records, such as personal information, academic performance, attendance, and disciplinary records.
- IUMS simplifies the registration and enrollment process for students by allowing them to choose courses, view availability, and sign up for classes online. It may also be in charge of payment processing and fee collection.
- To facilitate communication among students, professors, and administration, UMS frequently incorporates communication tools such as internal messaging systems, discussion forums, and announcement features.
- IUMS normally includes a grade book function that allows professors to record and calculate grades. It may also support other forms of evaluation such as quizzes, assignments, and exams.

Chapter 4 – Methodology

Early in the 1960s, as many commercial organizations started looking for efficient ways to streamline the realization of business benefits and organize work into an organized and distinct entity (later referred to as a "project"), the term "project management methodology" was first developed. The most crucial factors for creating effective working relationships between teams and departments within the same organization were communication and collaboration. New meanings, components, and functions have been added since then, and the term has undergone extensive updating and amendment. A set of fundamental principles and rules for managing a particular project with a clear beginning and end constitute a project management methodology today.(Project Management Methodology: Definition, 2023) A project management methodology is a carefully crafted collection of logically connected practices, procedures, and processes that establishes the best way to design, develop, oversee, and deliver a project from the beginning of its continuous implementation phase through to its successful conclusion and termination. It is a tried-and-true method that emphasizes organization, discipline, and systematic planning. (Project Management Methodology: Definition, Project Management Methodology: Definition, types, examples, 2023) This chapter will thoroughly examine the selected approach as well as the justification for implementing the strategy with sufficient study and information.

4.1 What to Use

Software methodology is a strategy for system design, organization, and development in software development. Methodologies for developing software are essential to the process. The waterfall model, prototype model, agile software development, quick application development, dynamic systems development model, spiral model, and collaborative application development are only a few examples of software development methodologies. Both the Dynamic System Development Method (DSDM) and Agile Methodologies are crucial to my project. Below, I'll discuss the benefits and drawbacks of three approaches:

Dynamic Systems Development method (DSDM)

A foundation for creating and maintaining systems is offered by the agile code development approach known as the Dynamic Systems Development Method (DSDM). The DSDM philosophy is based on a modified form of the sociologist's principle, according to which applications are frequently generated in 80% of the time necessary to deliver a 100% program, as opposed to 20% of the time. The 80% rule, which stipulates that only enough effort is needed for each increment to allow proceed to the next increment, is followed by DSDM iterative code method iterations. When numerous business requests have been identified or revisions have been sought and made, the final detail is typically completed later. ((DSDM), 18)

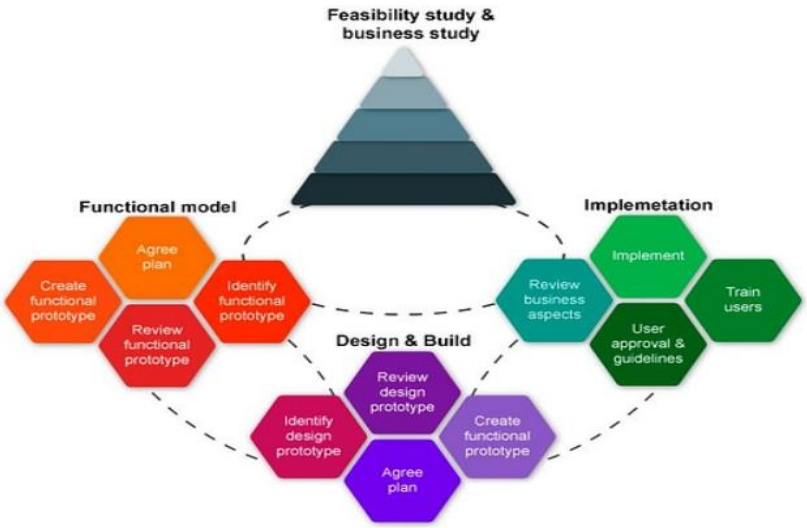


Figure 5: DSDM Methodology

Benefits of DSDM:

• It provides a strategy that is not reliant on a particular technique; and • It is flexible in terms of demand growth.

• Budgets and deadlines must be meticulously followed.

• The development process involves stakeholders.

• Each project team is required to have at least one tester because testing is so crucial.

• Has a unique way of figuring out whether each necessary iteration is relevant.

• Be flexible and quick to react to changes.

• Customers and developers can simply communicate with one another.

Drawbacks of DSDM:

- Focusing on RAD can reduce code robustness, it necessitates complete adherence to the DSDM process, it necessitates significant user involvement, it necessitates a skilled development team in both business and technical areas, and it necessitates a skilled development team in both business and technical areas.
- If the customer is uncertain about the project's end result, the project will be destroyed.

Agile Methodology

Agile is a method for developing software that anticipates the need for flexibility and uses pragmatism to deliver the finished product. Agile software development demands a cultural shift in many businesses because it places an emphasis on the clean delivery of discrete bits or parts of software rather than the whole application. (methodologies)?, 2022) Agile project management and software development practices are

iterative and adaptable. It places a focus on collaborative working, flexible planning, ongoing improvement, and quick delivery of functional software.

Figure 6: Agile Methodology

Benefits of Agile Methodology

- Agile allows for flexibility in project execution, allowing teams to adjust to shifting needs, priorities, and market circumstances. It makes it simpler to respond to new information, consumer input, and changing company demands since it accepts change as a normal part of the development process.
- Customer satisfaction: Agile strongly emphasizes customer participation and collaboration. Consistent feedback loops and the constant supply of functional software ensure that client expectations are satisfied. Agile improves client happiness and provides solutions that more effectively meet their demands by delivering value gradually.
- Quicker Time-to-Market: The iterative nature of Agile's methodology enables the early and frequent delivery of functional software increments. This makes it possible for enterprises to provide valuable improvements and capabilities more quickly, giving them a competitive advantage. Organizations are better able to react swiftly to changing customer preferences and market demands when they can offer increments in shorter cycles.Increased Transparency and Visibility: Agile promotes transparency and visibility across the project team and stakeholders. Practices such as daily stand-up meetings, regular demonstrations, and information radiators (e.g., task boards) provide clear insights into project progress, bottlenecks, and challenges. This transparency fosters effective communication, shared understanding, and collaboration among team members.
- Constant Improvement: Agile values a culture of constant development. Teams can reflect on their workflows, pinpoint areas for improvement, and make adjustments to improve productivity, quality, and efficiency with the help of regular retrospectives and feedback loops. This emphasis on continuing learning and adaptation promotes process optimization and improved team performance.

Drawbacks of Agile Methodology

- Lack of Predictability: Agile projects often prioritize adaptability over predictability. The iterative nature of Agile can make it challenging to accurately estimate project timelines and costs,

especially in the early stages when requirements may be evolving. This can be problematic for organizations that require strict adherence to schedules or fixed budgets.

- Scope Creep: Agile encourages flexibility and embraces changing requirements. However, if not managed effectively, this can lead to scope creep, where the project's scope gradually expands beyond the initial plan. Without proper controls, it can be challenging to maintain a balance between accommodating changes and maintaining project focus and delivery timelines.
- Dependency on Customer Availability: Agile places a strong emphasis on customer collaboration and feedback. This requires active participation and timely decision-making from customers or end-users throughout the project. If customers are not readily available or lack sufficient time to engage with the Agile team, it can hinder progress and impact project outcomes.

Rapid Application Development:

Rapid Application Development (RAD) is a flexible approach to software development that focuses more on early feedback and prototypes than on intricate design. Generally speaking, the RAD approach favors prototyping and development above planning. Without needing to start from zero, rapid application development enables developers to iterate and modify software frequently. This makes it more likely that the finished product will be high-quality and satisfy customers.(Rapid application development (RAD): Definition, 2023)

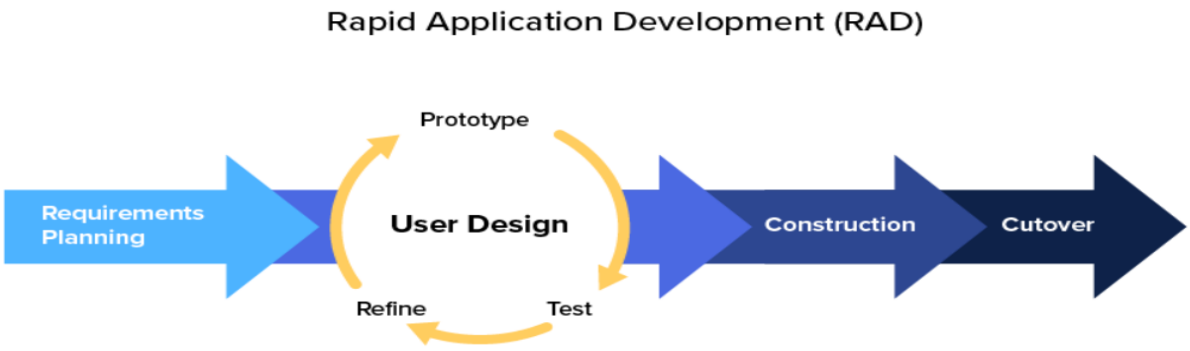


Figure 7: Rapid Application Development Methodology

Benefits of Rapid Application Development

- Faster Time-to-Market: RAD focuses on accelerating the development process by emphasizing iterative and parallel development. This allows for quicker delivery of functional software, reducing the time required to bring products or solutions to market. Faster time-to-market provides a competitive advantage and enables organizations to seize business opportunities promptly.
- Increased Stakeholder Satisfaction: RAD places a strong emphasis on involving stakeholders throughout the development process. By engaging stakeholders early and regularly, RAD ensures that their requirements and feedback are incorporated into the solution. This iterative feedback loop increases stakeholder satisfaction and helps deliver a solution that meets their expectations.
- Improved Collaboration and Communication: RAD encourages close collaboration and active participation from all project stakeholders, including business users, developers, and designers.

Regular communication and information sharing among team members foster a collaborative environment and ensure a shared understanding of project goals and requirements. This leads to better teamwork, reduced miscommunication, and improved project outcomes.

- Flexibility and Adaptability: RAD embraces changing requirements and promotes flexibility in development. It allows for quick adjustments and refinements based on stakeholder feedback and evolving business needs. This adaptability ensures that the developed solution remains aligned with the dynamic nature of the business environment, enabling organizations to respond quickly to changes and market demands.
- Reduced Development Risks: RAD mitigates risks by employing prototyping and iterative development. Prototypes allow stakeholders to visualize and validate the solution early in the process, reducing the chances of misunderstandings and incorrect interpretations of requirements. Early detection of issues and continuous testing during iterations help identify and resolve potential problems, leading to higher quality software and reduced project risks.

Drawbacks of RAD

- Limited Suitability: RAD may not be suitable for all types of projects. It is most effective for projects with well-defined and clear requirements. Projects that involve high complexity, extensive integrations, or critical security requirements may not be well-suited for RAD.
- Dependency on User Availability: RAD heavily relies on active and continuous user involvement. This can pose challenges if users are not readily available or lack the necessary time and resources to participate in the development process. Limited user availability can hinder the feedback and validation loops, impacting the effectiveness of RAD.
- Potential for Scope Creep: Due to the flexible and iterative nature of RAD, there is a risk of scope creep if not managed effectively. Without proper controls and change management processes, there is a possibility of the project expanding beyond the original scope, impacting timelines, budgets, and overall project success.

Choosing Methodology

The agile model has, however, occasionally been applied to university management systems. The DSDM technique will be the greatest choice and the most suitable solution for the university management system project because it consistently produces high-quality products and projects on schedule and is fairly adaptable for use in any firm or organization. During the development phase, clients or clients may modify their requirements if they feel it is important for their project.

4.2 Why to Use Methodology

The system must follow a methodology in order to meet the criteria within the time and financial restrictions. A methodology helps a project achieve its main goals by following its specified procedures. When deciding which method to use for an academic project.

While this may be difficult, it will significantly aid in the project's completion and delivery of outcomes.

4.3 (IUMS) Sections of Methodology

Some established features of the DSDM approach must be followed by the development team. These are as follows:

Phase one of the project

This section develops the initial project concept, as well as the time frame, budget, and basic criteria.

Phase of Feasibility Study

The business case, business strategy, technology solutions, and project budget are all influenced by this phase. The business case, business strategy, technology solutions, and project budget are all influenced by this phase.

Phase of requirement gathering

In this domain, many methodologies are utilized to determine both functional and non-functional project demands.

Phase of Requirement Analysis and Prioritization

In this step, the obtained demands are assessed and prioritized using a prioritizing approach such as MoSCoW.

Prioritization with MoSCow:

MoSCoW has designed a prioritization tool to aid in the development of the system. There are several components, which are listed below:

Must have:

In this section, it is critical to identify the system's core functions; otherwise, the system will be useless to the users.

Should have:

This section assists in the identification of crucial criteria and the development of a key that benefits everyone.

Could Have:

This section aids in identifying a few unimportant needs that have no impact on the system.

Won't Have:

This system aspect aids in identifying wants that aren't met by this system.

Engineering & Exploration Phase:

The engineering process includes iterative testing, system increments, system design, and ensuring that user requirements are met in compliance with system criteria. The MoSCoW prioritization technique used in the Exploration phase helps to define the system's functional and non-functional requirements. The technique's main iterative component was used to iteratively evaluate the requirements and design the solution.

Post-project:

This phase assists in determining the predicted benefit of the project and designing the system's final planned solution.

Phase of Review:

In this step, the deliverables are tested with users, and if changes are needed, they are returned to the previous phase.

4.4 Implementation Plans

The project is almost complete, and sophisticated apps are already usable. Make the new system operational if there are any issues with this identification and resolution. This section establishes the planning, configuration, and release criteria. The new system gets introduced if everything is in order.

Chapter 5 – Planning

5.1 Project Plan

The procedure for project completion planning is covered in this section. In essence, the project is broken down into a number of discrete segments, and if everything goes according to plan, all of the work will be finished within a certain amount of time.

5.1.1 Work Breakdown Structure

In order to accomplish the project within the allotted time frame while also being more effective and efficient, this step entails breaking it down into smaller pieces. This structure provides us with a job and time estimate. If this structure isn't in place, finishing the project can become more challenging. As a result, the suggested system has been split up into groups and subcategories in the WBC, as shown in the chart below:

No.	Task Name	Start	Finish	Duration
1.	Introduction	01/01/2023	01/15/2023	15
2.	Initial Study	01/15/2023	01/28/2023	13
3.	Literature Review	01/28/2023	02/05/2023	8
4.	Methodology	02/05/2023	02/15/2023	10
5.	Planning	02/15/2023	02/27/2023	12
6.	Feasibility Study	02/27/2023	03/15/2023	16

7.	Foundation	03/15/2023	03/22/2023	7
8.	Exploration & Engineering	03/22/2023	04/20/2023	28
9.	Deployment	04/20/2023	05/10/2023	20
10.	Testing	05/10/2023	05/20/2023	10
11.	Implementation	05/20/2023	05/25/2023	5
12.	Critical Appraisal & Evaluation	05/25/2023	06/05/2023	10
13.	Lessons Learned	06/05/2023	06/10/2023	5
14.	Conclusion	06/10/2023	06/18/2023	8
15.	Total	01/01/2023	06/18/2023	167

Figure 8: IUMS Work Breakdown Structure of the

5.1.2 IUMS Resource Allocation

To ensure that the planned project is finished on schedule and effectively, all resources and assets are assigned and managed. The distribution of resources is a crucial aspect of project planning. There is no team in this intellectual endeavor, so I shall take on different tasks at different times. The following resource allocation is being used by the IUMS project in order to reach the pre-scheduled work delivery deadline.

Task	Duration	Resource
Introduction	15	Analyst, User
Initial Study	13	Analyst
Literature Review	8	Analyst
Methodology	10	Analyst, User
Planning	12	Analyst, Team Leader

Feasibility Study	16	Analyst, User
Foundation	7	Analyst, Designer, Developer
Exploration & Engineering	28	Analyst, Designer, Developer
Deployment	20	Designer, Developer
Testing	10	Developer, Tester, User
Implementation	5	Analyst, Developer, User
Critical Appraisal & Evaluation	10	Analyst
Lessons Learned	5	Analyst, Designer, Developer, Tester
Conclusion	8	Analyst

Table 3: IUMS Resource allocation list

5.1.3 Time Boxing

Another important part of DSDM project planning is the segmentation of work into time boxes in order to meet goals ahead of schedule. In this area, all jobs are arranged into time boxes with a predetermined duration. These tasks must be done within the time constraints imposed by the iterative approach.

Time Box	Task Name	Duration	Resource
TB1	Introduction	15	Analyst, User
TB1	Initial Study	13	Analyst
TB1	Literature Review	8	Analyst
TB2	Methodology	10	Analyst,User

TB2	Planning	12	Analyst, Team Leader
TB2	Feasibility Study	16	Analyst, User
TB3	Foundation	7	Analyst, Designer, Developer
TB4	Exploration & Engineering	28	Analyst, Designer, Developer
TB5	Deployment	20	Designer, Developer
TB5	Testing	10	Developer, Tester, User
TB6	Implementation	5	Analyst, Developer, User
TB7	Critical Appraisal & Evaluation	10	Analyst
TB7	Lessons Learned	5	Analyst, Designer, Developer, Tester
TB8	Conclusion	8	Analyst

Table 4: IUMS List of the Time Boxes

5.1.4 Gantt Chart of IUMS Working

An illustration of the project's activity schedule is a Gantt chart. Instead of in days, it displays the length as a progress bar from the start date to the finish date. A Gantt chart for the Integrated University Management System looks like this:

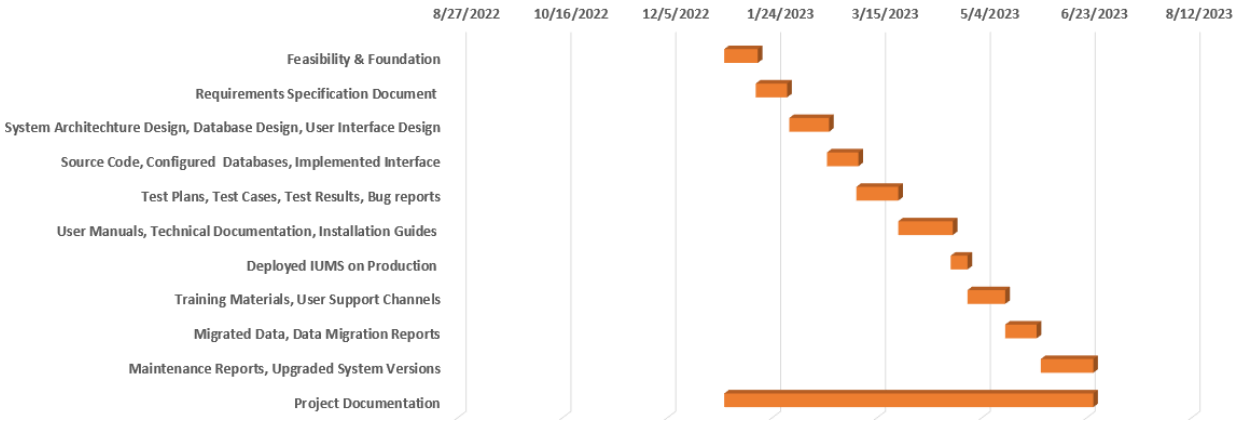


Figure 9: IUMS Project File cycle Gantt chart

5.2 Test Plan

A test plan is a comprehensive document that specifies the testing methodology, goals, schedule, expectations, deadlines, and resources required to execute a project. Consider it a plan with test managers in charge for doing the tests necessary to ensure that the software is working. The test strategy was developed in response to discrepancies between the supplied input and the system's anticipated output. Testing, which was done during the software development process, was done along with verification and validation.

5.2.1 IUMS Testing Against the Time Boxes

The time box approach was used to estimate a fixed and maximum unit of time for a particular segment. Time limits are being put to the test.

Username for testing purpose	Example for testing	Role for testing	Example for testing
ID of the Time Box			
Content of the Time Box			

Type of Test	Steps for testing	Expected outcome	Actual outcomes	Comment
Unit testing				
Integration testing				
System testing				
Acceptance testing				
Security testing				

Usability testing				
Reliability testing				

Figure 10: IUMS Sample of Testing Against the Timebox

5.2.2 Required Test

Although there are many additional types of testing modules available, functional and non-functional testing are the two most common kinds of system testing:

Functional Testing:

Unit Testing: In order to guarantee that individual units or components of a project perform properly, unit testing is a fundamental practice in software development. These units can be methods, functions, classes, or modules, depending on the programming language and project structure.

Unit testing's goal is to validate each unit's behavior and confirm that it responds to inputs in the desired manner. Unit testing enables targeted testing and early error or defect discovery by separating the units from the rest of the system. It helps ensure that each unit works as intended and that any changes made in the future do not introduce regressions or unexpected behavior.

When performing unit testing, the developer typically writes test cases for each unit, covering different scenarios and edge cases. These test cases consist of inputs and expected outputs or assertions. The developer executes the tests using a testing framework or tool, which automatically runs the tests and checks if the actual output matches the expected output. If a test case fails, it indicates a problem with the unit being tested, and the developer can investigate and fix the issue.

Integration Testing:

The goal of the software testing technique known as integration testing is to confirm how various system components or modules behave when they are linked together. Integrity testing is used to find flaws, mistakes, or inconsistencies that may occur as a result of how these components interact.

Unit testing, in which each separate unit is tested separately, is usually followed by integration testing. Integration testing is performed to make sure the units interact as intended once they have been tested and found to be correct. It assists in validating the functionality, data flow, and communication across numerous elements to make sure the system functions as a coherent whole.

System Testing:

System testing is a thorough testing strategy that concentrates on assessing the entire integrated software system to ensure that it complies with predetermined standards. In order to verify the system's functionality, performance, dependability, security, and other quality qualities, the entire system must be tested. One of the last phases of the software testing life cycle is system testing, which is carried out after integration testing.

Acceptance Testing:

User acceptance testing (UAT), often known as acceptance testing, is a testing process used to ascertain whether a software system satisfies end users' and stakeholders' expectations. Before the system is deployed or made available for usage in production, it must pass one last round of testing. The goal of acceptance testing is to make sure the system is prepared for usage and acceptance by the intended users.

Non-Functional Testing:

Security Testing:

A software system's security controls are evaluated during security testing in order to find any gaps or flaws that could be exploited by bad actors. Assuring that the system has strong security measures in place to guard against unauthorized access, data breaches, information leaks, and other security concerns is the main goal of security testing. It entails evaluating different facets of the system's security, including its network, applications, and data.

Usability Testing:

Usability testing is a technique for assessing a software system's, application's, or website's effectiveness and user-friendliness from the viewpoint of its target users. It focuses on evaluating how simple it is for users to understand, use, and interact with the system to achieve their objectives. Usability testing's main goal is to find usability problems, collect user feedback, and make the required adjustments to improve the user experience as a whole.

Reliability Testing:

Testing that focuses on assessing the dependability, consistency, and stability of a software system or product is known as reliability testing. Reliability testing measures a system's capacity to consistently carry out its intended functions over a predetermined time period and under anticipated operational circumstances. It entails putting the system through a variety of tests and scenarios to find any potential flaws, mistakes, or performance problems that could affect how reliable it is.

5.2.3 Test Case of IUMS

The test case comprised the test case number, test type, test description, and test procedures.

No. of Test Case
Type of Test
Description of the test

Steps of testing	Expected outcome	Actual outcome	Comment

Table 5: Sample Test case

5.2.4 User Acceptance Test Plan

No. of Test Case
Type of Test
Description of the test
Preparation for testing
Name of the User
Assume the role of

Steps of testing	Expected outcome	Actual outcome	Comment

Table 6: IUMS User Acceptance Test Plan

5.3 Risk Management of IUMS

Risk management is the process of identifying, evaluating, and addressing any risks that might arise throughout the project development life cycle in order to keep track of the project and achieve its goals and objectives. Project risks include things like budget, schedule, and performance. There are several ways to handle risks.

- ➢ Identifying risks
- ➢ Risk evaluation
- ➢ Risk mitigation / action plan
- ➢ Preparation for potential dangers

5.3.1 Risk Identification

Risk identification is one of the most crucial steps and jobs in the planning process for risk management.

Making ensuring that any potential hazards are identified, documented, and that any associated qualities are identified, documented, and associated with their characteristics is the first stage in risk identification. This study will justify a number of risk identification criteria, including the following:

- ➢ The identification and documentation of potential dangers.
- ➢ Actual causes were taken into account while identifying risks.
- ➢ The risk's influence on future repercussions and effects.

Elaboration of prospective risk identification and project risk monitoring, as indicated below:

Type of Dangers	Causes	Consequences & Impact
Lack Of Training	Delivery hazards can occur due to a number of factors, including: ➢ Not being able to provide accurate data due to lack of training. ➢ Getting unexpected results from the system if correct data is not given in the system. ➢ All the work of management, students and admin are disturbed to do wrong things.	➢ Late result published and submission of a project
Database Error and Destroy	➢ This effort involved a variety of users, including: Admin, Students, Teachers, Exam Controllers, and Department Heads, among others. ➢ If a student was unable to find a solution to a problem due to a lack of system materials failure implemented at the time of requirement. ➢ Due to a database error, the	➢ Management, Teachers, and Students were unable to identify the anticipated solution.

	teacher was unable to access the system.	
Unauthorized Entry	➢ If someone attempts to get unauthorized access to this system, this may result in a problem as well as a risk. ➢ This unauthorized user has access to the data of the Teacher, Students, and Management.	➢ Confidential data can be hacked and altered.
System Technical Problem ➢Hardware	➢ Inadequate speed ➢ As a result of a lack of critical performance ➢ Lack of required configuration.	➢ It is conceivable that the merchandise will not be delivered. ➢
Project Security Backup	➢ There is no backup of the project on GitHub or anywhere else equivalent.	➢ The financial ramifications ➢ The entire project has been ruined.
Failure of the Network	➢ Internet connectivity is necessary for certain integrated API and email management functions.	➢ Emails to the Student are strictly prohibited.

Table 7: Risk Identification of IUMS

5.3.2 Risk Assessment of IUMS

Risk assessment is concerned with the quantitative or qualitative value of a project's risks and dangers.

Type of risks	Likeliness	Impact	Restore time
Lack Of Training	5	6	8
Database Error and	5	5	5

Destroy			
Unauthorized Entry	6	6	6
System Technical Problem ➢Hardware	4	5	5
Project Security Backup	5	6	4
Failure of the Network	4	4	8

Table 8: Risk Assessment of IUMS

5.3.3 Risk Precaution / Action Plan

After risk identification and risk assessment, a risk action plan was created. There are many techniques that can be used to get ready for danger.

• To stay away from potential risks as well as those that have already occurred;

• To address issues that have emerged as well as possible threats;

• To reduce risks that have already occurred as well as those that may do so in the future.

Type of risks	Action	Taking Action	Action Required
Lack Of Training	Create accurate time estimates using a Gantt chart and other tools. Utilization of technology: Microsoft Excel and Asana	Actively participating developer	Make sure we have everything we need before we start working on a project.
Database Error and Destroy	➢ To carry out the required normalization	Actively participating developer	When it comes to data addition, updating, and

	➢ To build the precise required relationship on the table,		deletion,
Unauthorized Entry	➢ To write code while emitting a lovely odor. ➢ Create a multi-authentication system for this project.	Actively participating developer	When the authentication components are configured
System Technical Problem ➢Hardware			
Project Security Backup	Backing up our/my project can be done using any cloud service, such as GitHub.	Actively participating developer	Frequently updated
Failure of the Network	To effectively communicate with network service specialists.	Actively participating developer	On time for project completion.

Table 9: Integrated University Management (IUMS) Risk Precaution

5.3.4 Steps Taken for Possible Risks of IUMS

After discovering and analyzing potential threats, the responsible individual implements the necessary procedures. These are as follows:

Type of risks	Description	Likeliness	Impact	Taking Step

Lack Of Training	One of the most important risks is missing the Result Delivery and all Work deadlines.	Rare	High	The Gantt chart and task breakdown structure, as well as the project time boxing, were created.
Database Error and Destroy	There is a chance of a duplicate entry.	Likely	High	The entity relationship diagram and third normal form normalization were used to generate the data dictionary.
Unauthorized Entry	There is no access allowed to the system.	Unlikely	High	A suitable framework for access authorization and authentication has been put in place.
System Technical Problem ➢Hardware	It's conceivable for our or my hardware to malfunction or stop working.	Frequent	Mediu m	Backup is offered, along with continuing maintenance and supervision.
Project Security Backup	All project files should be backed up (using a service such as Github).	Rare	High	keeping an eye on the project file securely
Failure of the Network	There is a problem with the connection or the internet has stopped working.	Frequent	Low	High-bandwidth access has been given to the proxy server and it has been set up.

Table 10: Risk dealing steps for IUMS

5.4 Change Management

5.4.1 Factors that Might Cause Change

This system requires multiple types of adjustments, which are as follows:

• System training elements have been modified.

• Changes to the functional requirements

• Modifications to several user interface modules.

• Modifications to the Update Student Information and teacher's update result and student review segments in the teacher's portal.

• Security handling elements have been modified.

5.4.2 DSDM Welcome Change of IUMS

The project development life cycle should be adjusted as necessary based on the needs of the project. Due to these factors and the needs of the system, we should choose a strategy that can address a wide range of problems. Consequently, "DSDM" is one of the methods that is contemporary, dependable, and easy to use. This technique is used in this system to maintain and implement development time changes.

It works in several ways:

- It provides a system to keep track of each system module that has to be updated and asks several students for input on when to make segment adjustments.
- Only a few examples include changes to automation in student admission, result publication, applications, and management universities.
- It ensures security is maintained, it enables scalability and dependability, and it will operate after all the modifications have been made.

5.4.3 Considering Business Priority

An Integrated University Management System (IUMS) is a sophisticated software solution that prioritizes the efficient management of administrative processes within a university or higher education institution. It serves as a centralized platform, integrating various modules and functionalities to streamline operations and support the institution's business priorities

Priorities were as follows:

Topical Area	Priority Level
Training Segments	6
Functional requirements	8
Non-Functional requirements	6
Segments for ratings and reviews	5
Segments for security handling	8

5.4.4 Change Workshop

Communication will be improved by the user being aware of the team members' profiles. Developers that are interested in the project can organize a workshop and create a questionnaire to determine what changes and additions should be made to the system. Normal dialogues and implementations based on user feedback are made possible by understanding the profiles of different user categories, such as Admin, Teachers, and Students, which is what this change workshop will monitor.

• A change workshop is required to specify the change and include the management.

• Create a complete profile of each Student, teacher and admin including user profiles.

5.4.5 Changes that are allowed

Changes need to be approved in accordance with their priority. Accepting modifications was necessary due to a variety of circumstances, including the availability of resources, the updating of information, the release of results, the authorization section, as well as privilege and risk. We must allow for priority-based adjustments as we create a specific area of this system.

5.4.6 Key Decision Taker of Change

A variety of specialists in respective fields will advise change-makers. This project included separate decision makers for modifications.

- Analyst
- System Prerequisites System
- Tester System Developer

The developer can change the pertinent code, and the analyst can change the functional requirement as needed.

5.5 Quality Management

Quality management is concerned with upholding the standard that the user agreed to accept at the start of the project. A variety of factors influence quality management, including

- Standards and Best Practices: Adhering to recognized standards and best practices is fundamental to ensuring quality in an IUMS. The system should align with industry standards such as ISO 9001 (Quality Management System) and follow established guidelines specific to higher education institutions. This includes compliance with data protection regulations, accessibility standards, and other relevant requirements.
- User Requirements and Feedback: Quality management in an IUMS involves actively engaging with users, including students, faculty, and administrative staff. Gathering user requirements through surveys, feedback mechanisms, and usability testing helps ensure that the system meets their needs and expectations. Regular feedback loops enable continuous improvement and refinement of the IUMS.
- Testing and Validation: Rigorous testing and validation processes are essential to ensure the quality and reliability of an IUMS. This includes conducting functional testing, performance testing,

security testing, and user acceptance testing. By thoroughly testing the system before implementation, potential issues and bugs can be identified and resolved, ensuring a robust and reliable system.

- Data Accuracy and Integrity: Quality management in an IUMS involves maintaining the accuracy and integrity of data. The system should have built-in data validation mechanisms to prevent data entry errors and enforce data consistency. Regular data audits and verification processes can help identify and rectify any inconsistencies or discrepancies, ensuring the reliability of the system's data.
- Training and Support: Providing comprehensive training and ongoing support to system users is crucial for quality management. Users should receive proper training on system functionalities, workflows, and best practices. Adequate support channels, such as help desks, FAQs, and user guides, should be available to address user queries and concerns promptly.

5.5.1 Rules Applied to Maintain Quality

To maintain quality in an Integrated University Management System (IUMS) project, several rules and practices can be applied. Here are some key rules that can help ensure quality throughout the project lifecycle:

- Clearly Defined Requirements: Start by establishing clear and comprehensive requirements for the IUMS project. This involves engaging stakeholders, including university administrators, faculty, students, and IT staff, to gather their needs and expectations. Document these requirements in a detailed and unambiguous manner to serve as a foundation for the project.
- Robust Project Management: Implement a strong project management framework to ensure effective planning, coordination, and control of the IUMS project. This includes defining project objectives, establishing a realistic project schedule, allocating resources appropriately, and setting up a communication plan. Adhering to project management best practices ensures that the project stays on track and quality goals are met.
- Thorough Testing and Quality Assurance: Establish a comprehensive testing and quality assurance process to identify and rectify any defects or issues in the IUMS. This includes conducting functional testing, performance testing, security testing, usability testing, and user acceptance testing. By thoroughly testing the system at each stage, potential problems can be identified early, reducing the risk of issues in the production environment.
- Change Management: Implement a change management process to effectively handle any modifications or updates to the IUMS throughout the project. This involves documenting and evaluating change requests, assessing their impact on the system, and following a structured change control process. Adhering to change management practices ensures that changes are properly assessed, approved, and implemented while maintaining system quality.
- Risk Management: Implement a risk management framework to identify, assess, and mitigate potential risks that may impact the quality of the IUMS project. This involves conducting risk assessments, creating risk mitigation plans, and monitoring risks throughout the project lifecycle. By proactively addressing risks, the project team can minimize their impact and maintain project quality.

5.5.2 DSDM Standard Quality Measures

Solution quality: Based on anticipated business needs and user expectations, solution quality has been ensured by monitoring user needs.

To carry out this plan, two different priority techniques were used.

- Time Boxing
- MoScoW

The deadline has passed after completion of these two procedures. The process quality has been guaranteed based on the predicted organizational structure. To make sure that these procedures are of high quality, two approaches have been created and used.

- CMMI
- DSDM

5.5.3 Quality Plan and Measuring Meter

At this time, several types of quality plans have been kept in place, including

- The delivery and distribution of resources were done well.
- Manage every piece of data related to the university in a risk-free way.
- Testing was conducted in each section after each module was finished.
- To move forward with the execution process after all changes have been made, it is critical to track and document all changes.

Chapter 6 – Feasibility

6.1 All Possible Types of Feasibility Integrated University Management System (IUMS)

The proposed IUMS has proper admin input validation and verification, a very easy and smooth navigation system, easy to identify all functions, and up-to-date information, all of which make the system much easier and effortless to use. Usability, ease-of-use, and required functionalities are critical to operational feasibility. Administrators, teachers, students, managers, and parents can quickly visit their site and complete their tasks with the proper credentials.

Technical Feasibility:

Some technical feasibility considerations include:

This type of feasibility assesses the organization's technical capabilities to implement an IUMS. It evaluates factors such as the existing IT infrastructure, hardware and software requirements, compatibility with existing systems, availability of technical expertise, and the feasibility of integrating the IUMS with other systems. According to a technical feasibility test, IUMS users can easily manage their activities when compared to the previous manual technique. Considerations for technical feasibility include determining the organization's technical capacity to execute an IUMS. It assesses elements including the current IT infrastructure, the necessary hardware and software, compatibility with current systems, the accessibility of technical skills, and the viability of integrating the IUMS with other systems. A technical feasibility test revealed that IUMS users can easily manage their activities in comparison to the earlier manual method. The system was created utilizing cutting-edge web and mobile technologies, making it compatible with any web browser and low-bandwidth Internet connection. To ensure secure access to system data, the system contains an access control hierarchy. The web application's development environment and resources are independent of platforms.

So, plainly, the technological parts of this project are as follows:

Hardware:

• Dell (PC)

• Wi-fi Router Config (Tp-link)

Software:

• Microsoft office (MS-2013)

• Microsoft Excel (MS-2013)

• Edge browser

• Windows 11(operating system)

•VS Code

Database:

• MySQL

Technology:

Front End or Design Slide:

• Html 5

• CSS

• Java Script

• Bootstrap

• Angular

Server Side:

• Java

• Spring boot

Financial Feasibility: Financial feasibility examines the financial resources required for implementing and maintaining an IUMS. It involves assessing the cost of system acquisition, development or customization, hardware and software costs, ongoing maintenance and support expenses, and potential return on investment (ROI). A cost-benefit analysis and budget considerations are important aspects of financial feasibility.

The suggested system can be developed in a variety of ways, including web-based and desktop apps.

➢ Web-Based Application Cost:

The entire application and data are saved on a remote server, which can be accessed from anywhere at any time via the Internet.

Equipment	Cost per unit	Cost
Extranet network with VPN access	2000 per/m	2,000
Dell PC (core i5, 2.50-271 GHz processor, 16 GB DDR4 RAM, 1 TB HDD)	85,000	85,000
File and Email and cloud servers	17,000 per/m	17,000 per/m
Total		1,04,000

➢ Desktop Application cost:

Equipment	Cost per unit	Cost
Dell PC (core i5, 2.50-271 GHz processor, 16 GB DDR4 RAM, 1 TB HDD)	85,000	85,000
File and Email and cloud servers	17,000 per/m	17,000 per/m
Total		1,02,000

We'll need a domain name and a hosting provider to get the system up and running. Because instructor and student materials must be updated, the system must constantly address problems and monitor all information. The remainder of the system will be built as a web app.

Integrated University management system Market Research Analysis Based on the Feasibility Factors:

Market research analysis based on feasibility factors can provide valuable insights into the potential success and adoption of an Integrated University Management System (IUMS). Here's an overview of how each feasibility factor can be considered in the market research analysis

Technical Feasibility: Assess the technological landscape of the target market, including the availability and compatibility of existing IT infrastructure and systems. Evaluate the market's readiness for adopting new technologies and identify any technical challenges that may arise during implementation. Determine the market's ability to support the technical requirements of an IUMS.

Financial Feasibility: Analyze the market's financial capabilities and willingness to invest in an IUMS. Evaluate the affordability of the system for different types of educational institutions, considering factors such as size, funding sources, and budget constraints. Conduct a cost-benefit analysis to demonstrate the potential financial advantages of implementing an IUMS in the market.

Operational Feasibility: Understand the market's operational processes and workflows to assess the potential impact of an IUMS. Identify any specific operational challenges or requirements that need to be addressed. Evaluate the market's readiness to embrace new processes and technologies, and assess the level of support and training needed for successful implementation.

Legal and Compliance Feasibility: Research the regulatory landscape of the target market, including data protection laws, privacy regulations, and educational compliance requirements. Identify any legal barriers or constraints that may impact the implementation of an IUMS. Ensure that the proposed system meets all relevant legal and compliance standards in the market

Time Feasibility: Evaluate the market's urgency and timeline for implementing an IUMS. Consider factors such as academic calendars, enrollment cycles, and any specific time constraints. Assess the market's readiness to commit resources and dedicate the necessary time to implement and adopt an IUMS within the desired timeframe.

By conducting market research analysis based on these feasibility factors, you can gain valuable insights into the market's readiness, challenges, and opportunities for implementing an IUMS. This analysis will help you understand the market dynamics, target the right audience, and tailor the system to meet the specific needs and requirements of the market, ultimately increasing the chances of success.

6.2 Cost Benefit Analysis for Integrated University Management System:

The main goal of a cost-benefit analysis is to determine the earnings and spending estimates. Comparing the total cost and earnings yields the benefit. The following is the cost-benefit analysis for this project:

Total cost:

Serial No.	Equipment	First Year	Second Year	Third Year	Fourth Year	Five Year	Total

1	Web based application cost	□ 1,04,000	-	-	-	-	□ 1,04,000
2	Desktop Application cost	□ 1,02,000	-	-	-	-	□ 1,02,000
3	The cost of a domain plus hosting	□25,000	□25,000	□25,000	□25,000	□25,000	□ 125,000
4	Expenses for Employees	□40,000	□40,000	□40,000	□40,000	□40,000	□2,00000
5	Other Cost	□20,000	□20,000	□20,000	□20,000	□20,000	□1,00000
6	Total Cost	□ 291,000	□ 85,000	□ 85,000	□ 85,000	□ 85,000	□ 631,000

Table 11: Total Cost Estimation for the project IUMS

Earning Sector:

Serial No.	Earn Sector	First Year	Second Year	Third Year	Fourth Year	Five Year	Total
1	Govt. Tax one	□1,50,000	□1,80,000	□2,10,000	□2,40,000	□2,70,000	□1,050,000
2	Govt.	□2,00000	□2,10,000	□2,40,000	□2,60,000	□3,00000	□1,210,00

	Tax two						0
	Total	□3,50,000	□3,90,000	□4,50,000	□5,00000	□5,70,000	□2,260,000

Table 12: Earning estimation for the project IUMS

Total Revenue of project:

Serial No.	Sector	First Year	Second Year	Third Year	Fourth Year	Five Year	Total
1	Total Earning	□3,50,000	□3,90,000	□4,50,000	□5,00000	□5,70,000	□2,260,000
2	Total Equipment Cost	□4,02,000	□80,000	□80,000	□80,000	□80,000	□7,22,000 0
	Total Revenue	□4,02,000	□3,10,000	□3,70,000	□4,20,000	□4,90,000	□1,538,000

Table 13: IUMS Estimated Revenue on a Five-year scale

6.3 Explain DSDM Good or Bad for this Project

This analysis examines the suitability of the Dynamic Systems Development Method (DSDM) for an Integrated University Management System (IUMS) project. DSDM is an agile project management framework that focuses on delivering results quickly while maintaining a strong emphasis on stakeholder collaboration and continuous feedback. By evaluating the advantages and potential challenges of adopting DSDM for an IUMS project, we aim to provide insights into its effectiveness and applicability.

Ultimately, the suitability of DSDM for an IUMS project depends on the specific context, organizational culture, and project requirements. Careful consideration of the advantages and challenges discussed can

help project stakeholders make an informed decision about adopting DSDM or selecting an alternative project management approach for their IUMS project.

Chapter 7 – Foundation

7.1 The Problem Area Identification

This problem area identification focuses on identifying key challenges and pain points within higher education institutions that an Integrated University Management System (IUMS) project aims to address. By understanding these problem areas, the IUMS project can prioritize and develop solutions that effectively meet the needs of the institution and its stakeholders.

7.1.1 Interview

The greatest technique to gather information, identify the issue, or ascertain the objective is through an interview. Finding the problem could help you find the right answer. As a result, a number of user-related questions have been developed to be asked during the following interviews for the proposed system:

1. Admin:

• What kind of issue does an administrator face at the moment?

• Is the notice real-time, and how can I get it?

• Is the earlier problem solution noteworthy in history?

• Have you encountered any difficulties in obtaining updated information or in determining eligibility?

2. For the Teacher:

• At the time of delivering the results, what type of problem do you have?

• Is the current distribution technique operating properly, or does it need to be improved to make it more seamless and user-friendly?

• Any assistance from the admin in managing that portion of our system would be greatly welcomed.

3. For Students:

• Have you encountered any difficulties in obtaining entrance exam or academic results in order to check your record?

• Had to deal with a management issue.

7.1.2 Observations

A tool and methodology used in the PMBOK's Collect Requirements method is observation, often called as job shadowing. The process of collecting requirements identifies and records stakeholder requests in relation to the project's goals. This approach involves seeing individuals carry out their duties. (chapter, 2022)

- Observations can also be used to gather user demands and identify the true problem. The majority of observation strategies are employed to go to the workplace and figure out how to accept the users. The following are the most important considerations:
- User-Friendly Interface: The IUMS has a user-friendly interface that is easy to navigate and understand. The menus and options are logically organized, making it convenient for users to access the desired information and functionalities.
- Centralized Data Management: The IUMS effectively centralizes data management, eliminating the need for multiple systems and databases. It allows users to access and update data from various departments and administrative functions, promoting data consistency and integrity.
- Streamlined Processes: The IUMS streamlines various processes within the university, such as student admissions, course registration, and faculty evaluations. By automating these processes, it reduces paperwork, minimizes manual errors, and improves operational efficiency.
- Efficient Communication Channels: The IUMS provides efficient communication channels for students, faculty, and administrative staff. Users can easily communicate, share information, and collaborate through the system, improving coordination and reducing communication gaps.
- Student Support and Services: The IUMS offers comprehensive support and services for students. It provides easy access to academic counseling, library resources, career services, and student activities, enhancing the overall student experience.
- Robust Reporting and Analytics: The IUMS includes robust reporting and analytics capabilities, enabling data-driven decision-making. Users can generate reports, analyze data, and monitor performance indicators, facilitating strategic planning and evaluation.
- Secure Data Management: The IUMS prioritizes data security and privacy. It implements robust security measures, including user authentication and access controls, to protect sensitive information and comply with data protection regulations.
- Mobile Compatibility: The IUMS is compatible with mobile devices, allowing users to access the system on-the-go. The responsive design ensures a seamless user experience across different screen sizes and devices.
- Integration with External Systems: The IUMS seamlessly integrates with external systems, such as financial management or learning management systems, to facilitate data exchange and streamline processes.
- Training and Support: The IUMS provides training and support resources to help users effectively utilize the system. Training materials, user guides, and helpdesk support are available to assist users in navigating the system and resolving any issues.

These observations highlight the positive aspects of the IUMS, showcasing its ability to centralize data, streamline processes, improve communication, and enhance overall efficiency within the university.

7.1.3 Questionnaires

In project management, gathering information is critical for making sound decisions. There are numerous strategies for getting information from diverse types of people. Project managers frequently employ observation, focus group discussions, supported seminars, and benchmarking.

One of the best methods for gathering data and polling consumers and other stakeholders is through questionnaires. A series of inquiries will be made of the user, some of which may be MCQs or brief responses. The questions that are offered below are a result of this.

Identification of Problems Questions

Name		**Age**	
General User	Admin	Problem	

Question-one	How frequently do you use the system and submit all relevant information for resolution?
Answer	
Question-two	How would you like to be notified if a transmitter error occurs again?
Answer	
Question-three	What are your thoughts on resolving your problem?
Answer	
Question-four	Tell us what you think about the Integrated University Management System.
Answer	

Identification of Problems Questions

Name		**Age**	
General User	Teacher	Problem	

Question-one	How do you remember which uploaded mark you wish to update?
Answer	
Question-two	What system should be utilized to process the resources that have been uploaded to the system?
Answer	
Question-three	What strategy do you employ when communicating with the students?
Answer	

Identification of Problems Questions

Name		**Age**	
General User	Student/Applicant	Problem	

Question-one	How do you achieve frequent results?
Answer	
Question-two	What communication strategy do you utilize with the teachers?
Answer	
Question-three	What should I do about admission or tuition fees?

Answer	

7.2 Rich Picture

A rich picture is a top-down or bird's-eye view of the activities of a system's users. It also portrays conflicting issues, as well as communication and business methods. The IUMS is depicted in greater detail below: -

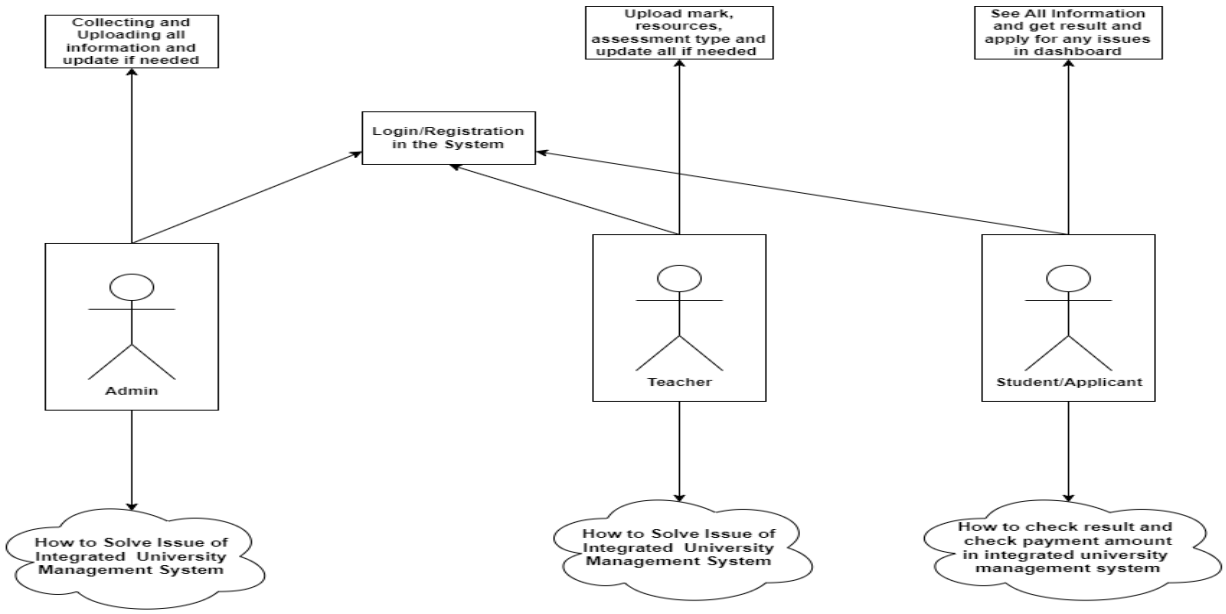


Figure 11: Rich Picture of the IUMS

IUMS The legends of rich picture

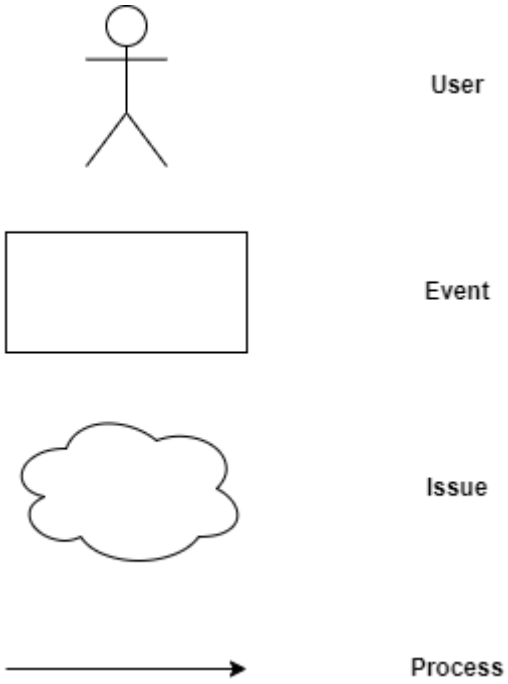


Figure 12: IUMS Legends of the rich picture

Key actors

In the IUMS system, there are three different sorts of actors.

➢Admin

➢ Teacher

➢ Student

The illustration indicates that the administrator may monitor and manage all processes as well as resolve problems.

7.3 Specific Problem Area Identification

Through information gathering procedures, a variety of issues are discovered during the problem area identification phase. A comprehensive graphic depicts all of the IUMS' system processes. By analyzing them, I was able to identify particular issues, such as:

- Inefficient Student Registration Process
- Lack of Integration with External Systems

- Limited Accessibility and User Experience
- Ineffective Communication Channels
- Data Security and Privacy Concerns
- Lack of Robust Reporting and Analytics
- Inadequate Support and Training
- Difficulty in Content Management
- Scalability and Performance Issues
- Compliance with Regulatory Requirements

These problem areas highlight the specific challenges or issues that can be identified within the IUMS project. Addressing these problem areas will contribute to improving the functionality, usability, security, and overall effectiveness of the system.

7.4 Possible Solutions

The following are the suggested solutions to the identified issues:

- Implement a streamlined online registration process with clear instructions and user-friendly forms.
- Integrate automated validation checks to minimize errors and provide instant feedback to students.
- Offer support resources, such as online tutorials or help desk assistance, to guide students through the registration process.
- Develop robust APIs and data integration mechanisms to facilitate seamless communication between IUMS and external systems.
- Collaborate with external system providers to establish standardized data formats and protocols for smooth data exchange.
- Conduct thorough testing and debugging during the integration process to ensure data consistency and accuracy.
- Conduct accessibility audits to identify and address barriers for users with disabilities.
- Implement responsive design principles to optimize the user experience across different devices and screen sizes.
- Incorporate user feedback through usability testing and iterative design improvements.
- Implement a centralized messaging system within IUMS to facilitate efficient and timely communication among users.

7.5 Overall Requirement List

The following is the final list of requirements:

Functional Requirement

Functional requirements for IUMS (Integrated University Management System) typically include the following:

User Authentication and Access Control:

- The system should support user registration and authentication to ensure secure access.
- Different user roles (students, faculty, staff, administrators) should have appropriate access permissions based on their roles and responsibilities.

Student Information Management:

- The system should capture and maintain accurate student information, including personal details, academic records, and enrollment history.
- It should support functionalities such as student admissions, course registration, and student profile updates.

Course Management:

- The system should provide features for managing course offerings, including course scheduling, curriculum management, and prerequisites.
- It should facilitate course enrollment, withdrawal, and waitlist management for students.

Grade Management:

- The system should support recording and management of student grades.
- It should provide functionality for faculty to enter and submit grades, calculate GPA (Grade Point Average), and generate official transcripts.

Faculty Management:

- The system should enable faculty members to manage their profiles, including personal information, qualifications, and teaching assignments.
- It should support functionalities such as course assignment, grade submission, and communication with students.

Administrative Functions:

- The system should provide administrative features for managing various processes, including admissions, financial aid, billing, and student records.
- It should support functionalities like generating reports, tracking student progress, managing academic policies, and facilitating administrative approvals.

Resource Management:

- The system should support the management of university resources, such as library resources, facilities, and equipment.
- It should enable students and faculty to access and reserve resources as needed.

Financial Management:

- The system should facilitate financial processes, including billing, payment management, and financial aid administration.
- It should provide features for generating invoices, tracking payments, and managing student financial records.

These functional requirements serve as a starting point and can be tailored based on the specific needs and requirements of the university. It is important to engage stakeholders and conduct detailed requirements gathering to capture all necessary functionalities for the IUMS.

Non-functional Requirements

Non-functional requirements for IUMS (Integrated University Management System) typically include the following:

Performance:

- The system should be responsive and provide fast response times, even during peak usage periods.
- It should be able to handle a large number of concurrent users without significant degradation in performance.

Reliability:

- The system should be reliable, with minimal downtime and interruptions in service.
- It should have backup and recovery mechanisms to ensure data integrity and system availability in case of failures.

Security:

- The system should have robust security measures to protect sensitive data and prevent unauthorized access.
- It should support secure user authentication, data encryption, and access control mechanisms.
- Compliance with data protection regulations and industry best practices should be ensured.

Scalability:

- The system should be scalable and able to accommodate future growth in terms of user base, data volume, and system load.
- It should have the ability to handle increasing demands without significant performance degradation.

Usability:

- The system should have a user-friendly interface and intuitive navigation.
- It should be easy to learn and use, requiring minimal training for users.
- Accessibility features should be implemented to ensure usability for users with disabilities.

Compatibility:

- The system should be compatible with various platforms, operating systems, and web browsers.
- It should support multiple devices, including desktops, laptops, tablets, and mobile devices.

These non-functional requirements help ensure that the IUMS meets the necessary standards for performance, reliability, security, usability, and compatibility. They provide guidelines for designing and developing a robust and efficient system that meets the needs of the university and its stakeholders.

7.6 Technology to be implemented

The proposed system can be implemented using a range of technological strategies. It is crucial to pick the appropriate technology if you want to succeed. The choices include:

Client-Server Application Technology

In the client-server computing paradigm, clients and servers operate independently but are connected by a computer network. Any instance of the client can ping a running server and wait for a response. (2, n.d.)

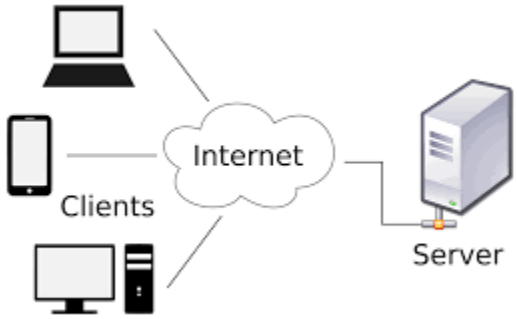


Figure 13: Client Server Application Model

The following are the fundamental characteristics of a client-server application:

• Physical installation of the application is required.

• It's important to keep both the server and client programs up to date.

• The pricey application must be purchased by users.

• It is not always available and is not portable.

Web Application

A web application (or web app) is application software that runs on a web server as opposed to computer-based software applications that run locally on the device's operating system (OS). The user needs a web browser and an active network connection in order to use web apps.

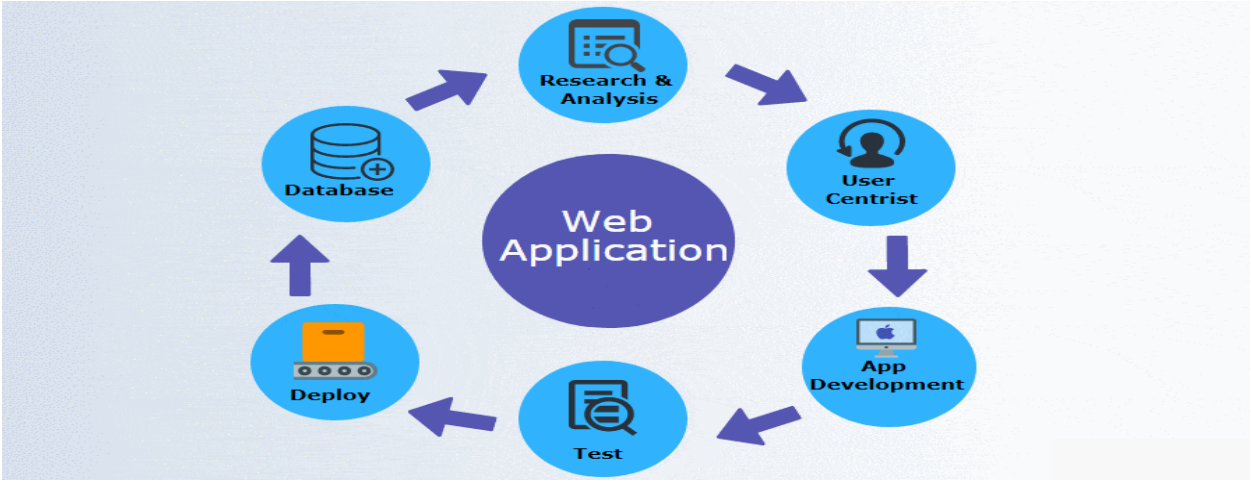


Figure 14: Web Application

The following are the primary characteristics of a web-based application:

• A web browser can be used to access the cloud over the internet; there is no requirement for installation.

• The cost is reasonable.

• Simple to access from anywhere at any time.

7.7 Recommendation and Justification

The planned Integrated University Management System must always be reachable from all points across the nation. Any internet-capable device, such as a tablet, laptop, desktop, or smartphone, should be able to access the system. Additionally, it must to be able to support numerous concurrent users. A web application is the best option for the suggested system because client-server application technology does not support it.

Chapter 8 – Exploration

8.1 University Management Sector Old System Use Case

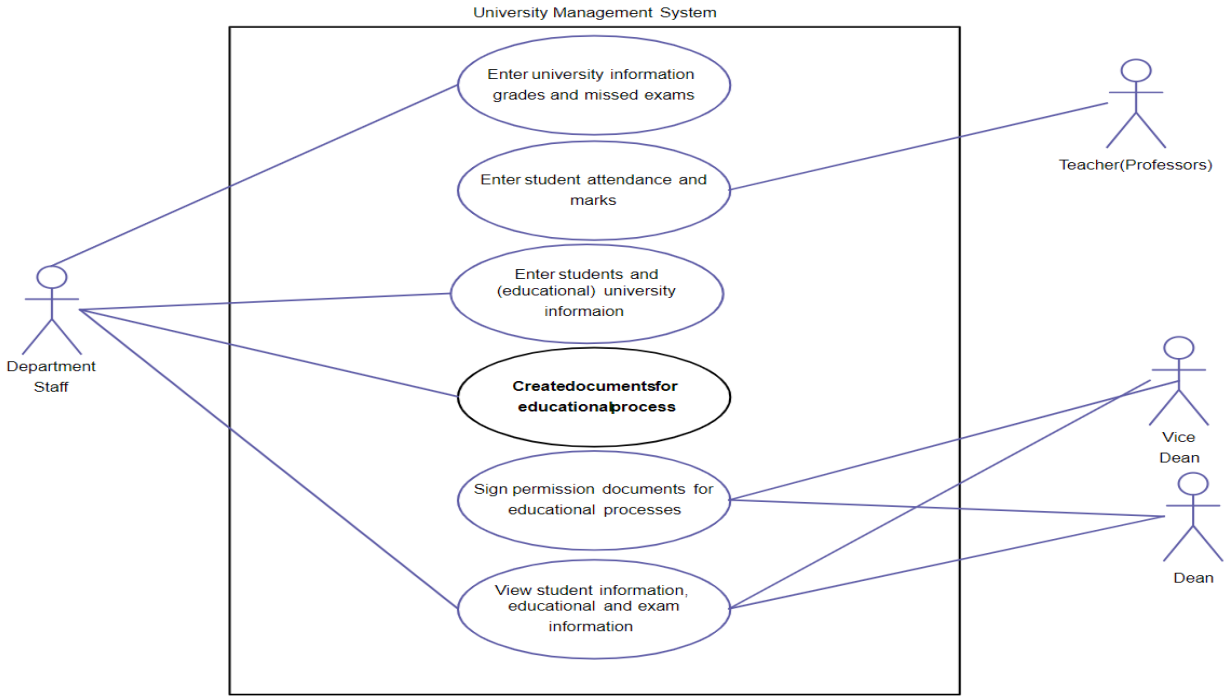


Figure 15: Use Case Diagram of University Management System

8.2 Activity Diagram

IUMS Admin Activity Diagram

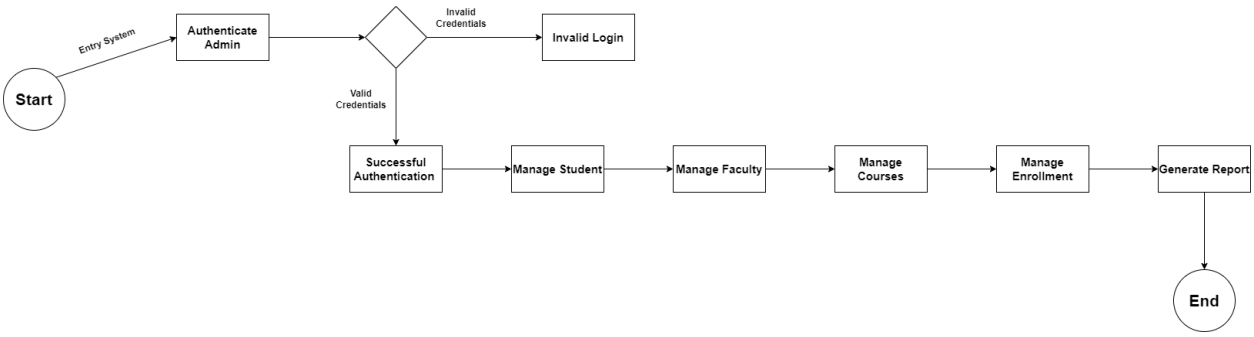


Figure 16: Admin Activity Diagram of IUMS

8.3 Full System Use Case

The following is a use case diagram for the proposed IUMS:

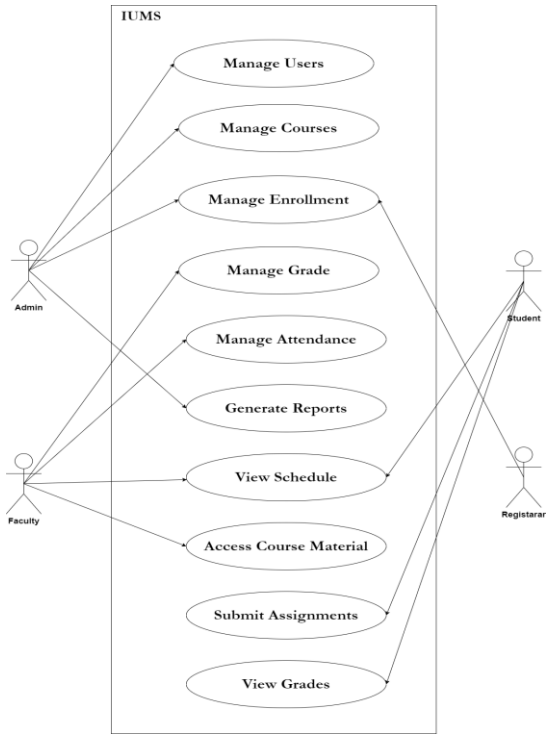


Figure 17: Use Case of the Proposed IUMS System

8.4 Full System Activity Diagram

The projected IUMS system will have a variety of users, each with their own workflow. Below are activity diagrams for various work processes.

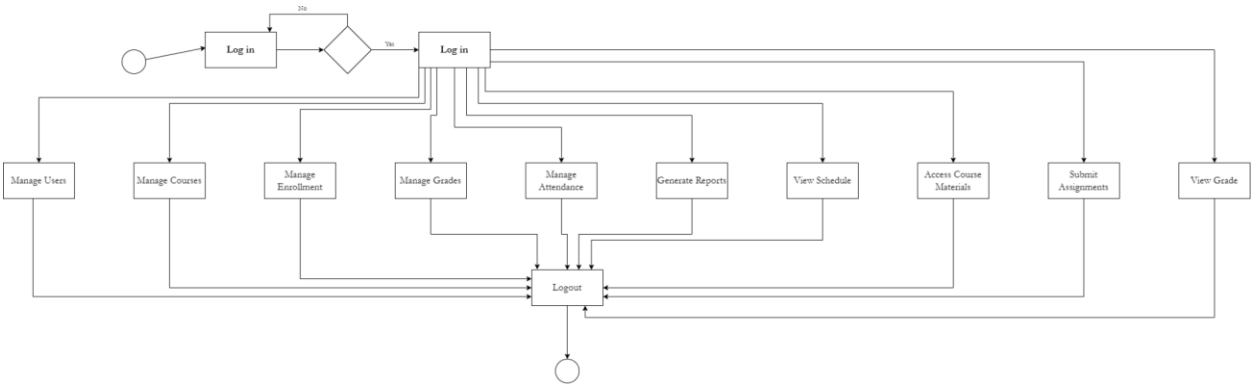


Figure 18: Admin Activity Diagram IUMS System

8.5 Teacher's Activity Diagram

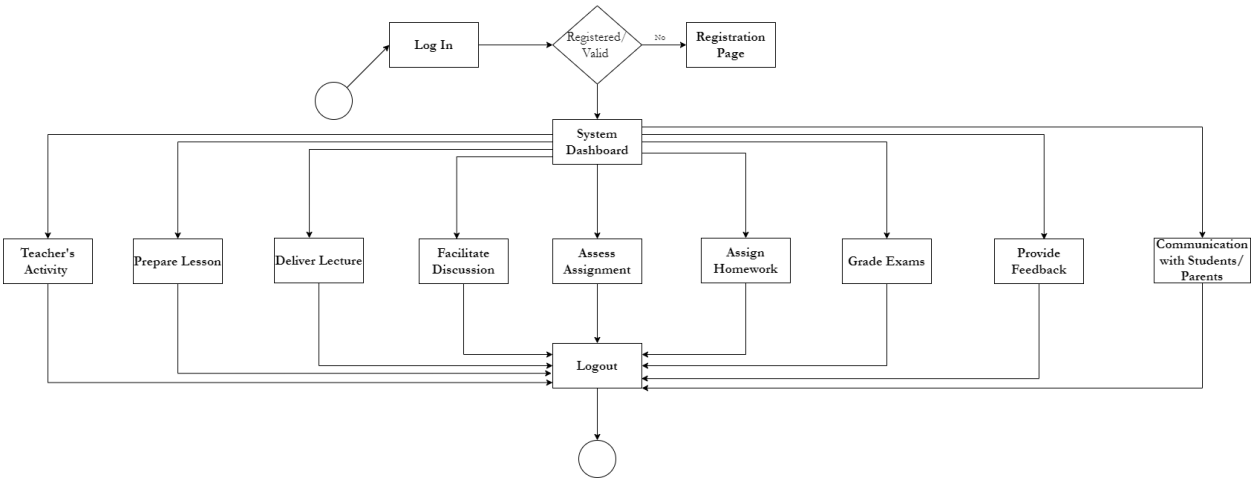


Figure 19: Teacher's Activity Diagram IUMS System

8.5 Catalogue of Requirements

A requirement catalogue is a list of all the identified needs for an agriculture management system project. Now, using the standard approach, I'll enter the identified needs into the requirement catalog.

User Requirements Catalogue

Source	Sing Off	Priority	Requirement
User Authentication and Access Control	Admin/Management	Must Have	M-01

Functional Requirement User Requirement Catalogue: User Registration and Authentication Allow users (administrators, faculty, staff, and students) to register and create accounts.

Non-functional requirement

Description	Target Value	Acceptance Value	Comment
User (per/day)	12000 (per/day.)	9500	

Table 14: Requirement Catalogue for User

Student Requirements Catalogue

Source	Sing Off	Priority	Requirement
Student Management	Admin & Student	Must Have	M-02

Functional Requirement Student Management Requirement Catalogue: Student Information System (SIS) Enable student registration, enrollment, and course management.

Non-functional requirement

Description	Target Value	Acceptance Value	Comment

Student (per/day)	10000 (per/day.)	7500	

Table 15: Requirement Catalogue for Student

Course Requirements Catalouge

Source	Sing Off	Priority	Requirement
Course Management	Admin, Student & Teacher	Must Have	M-03

Functional Requirement Course Management Requirement Catalogue: Learning Management System (LMS) Enable course creation and management, including content organization and delivery

Non-functional requirement

Description	Target Value	Acceptance Value	Comment
Course(per/day)	15(per/day.)	20	

Table 16: Requirement Catalogue for Course

Financial Requirements Catalogue

Source	Sing Off	Priority	Requirement
Financial Management	Admin	Must Have	M-04

Functional Requirement

Financial Management Requirement Catalogue: Enable financial transactions, such as tuition fee payment and invoice generation

Non-functional requirement

Description	Target Value	Acceptance Value	Comment
Financial (per/day)	150000(per/day.)	200000	

Table 17: Requirement Catalogue for Financial

Faculty and Staff Management Requirements Catalouge

Source	Sing Off	Priority	Requirement
Faculty and Staff Management	Admin	Must Have	M-05

Functional Requirement Faculty and Staff Management Requirement Catalogue: Allow faculty and staff to view and update their personal information, including contact details and qualifications.

Non-functional requirement

Description	Target Value	Acceptance Value	Comment
Faculty and Staff	15(per/day.)	20	

(per/day)			

Table 18: Requirement Catalogue for Faculty and Staff

8.6 Prioritized Requirements List (PRL)

I utilized the MoSCoW prioritization approach to construct a priority list of recognized requirements.

The priority requirements of the Agriculture Management System are shown below.

Must Have requirements-

Serial No.	**Requirement for IUMS**	**Priority of IUMS**
01.	User Authentication and Access Control	Must-have
02.	Student Management	Must-have
03.	Course Management	Must-have
04.	Financial Management	Must-have
05.	Faculty and Staff Management	Must-have

Should Have Requirements-

01.	Academic Advising	Should-have
02.	Online Registration and Enrollment	Should-have
03.	Grade Management	Should-have
04.	Reporting and Analytics	Should-have
05.	Communication and Notifications	Should-have

Could Have Requirements-

01.	Learning Management System Integration	Could-have
02.	Alumni Management	Could-have
03.	Research Administration	Could-have
04.	Resource Booking	Could-have

8.7 Admission Process

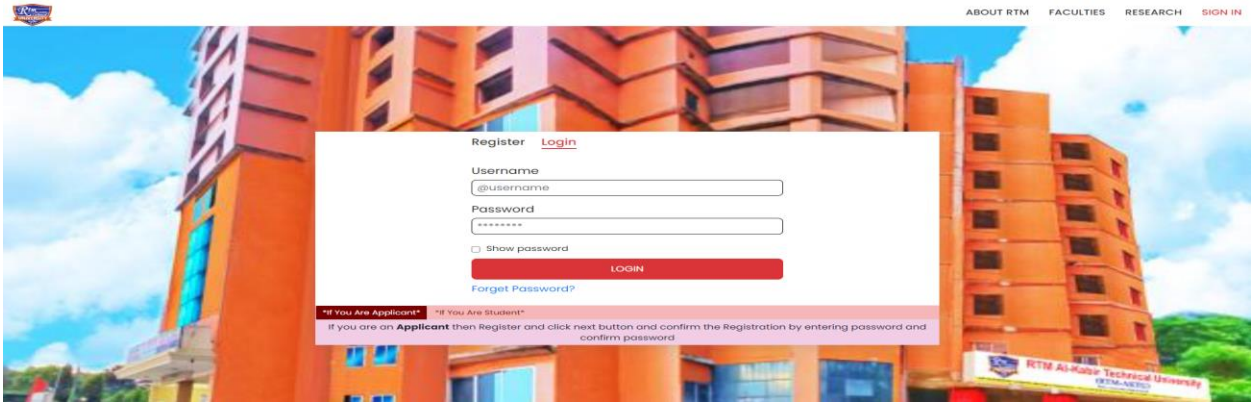


Figure 20: Login Page

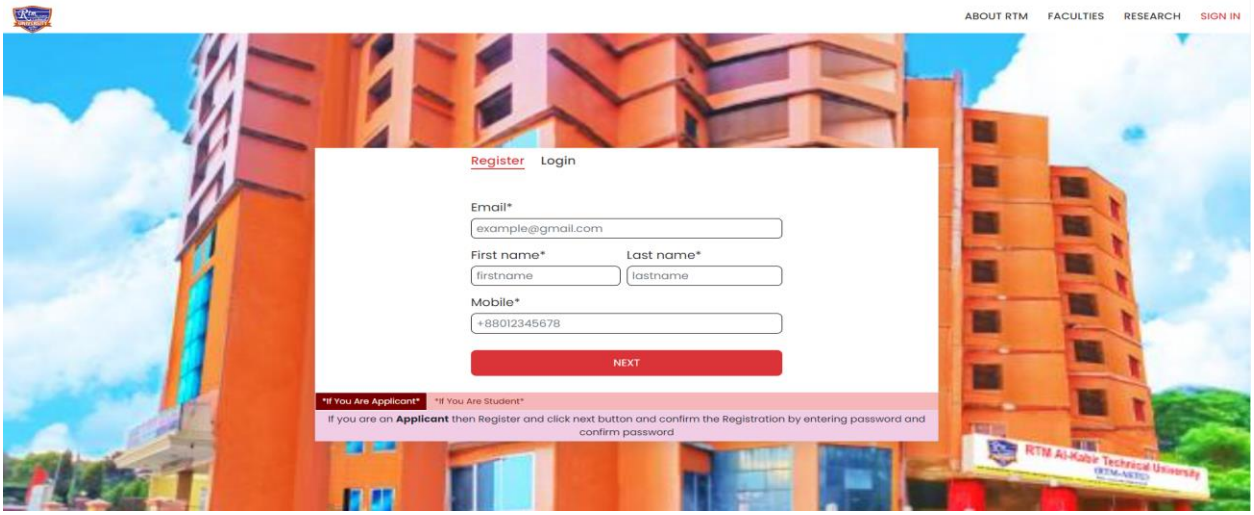


Figure 21: Registration

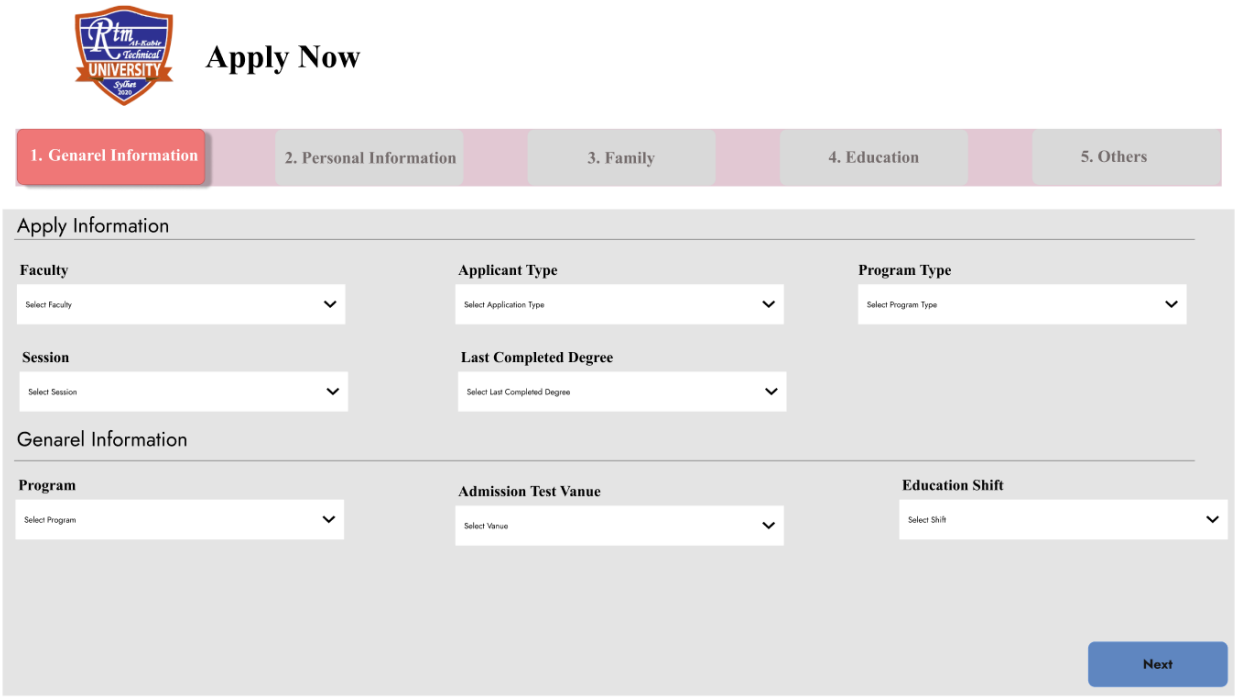


Figure 22: General Information

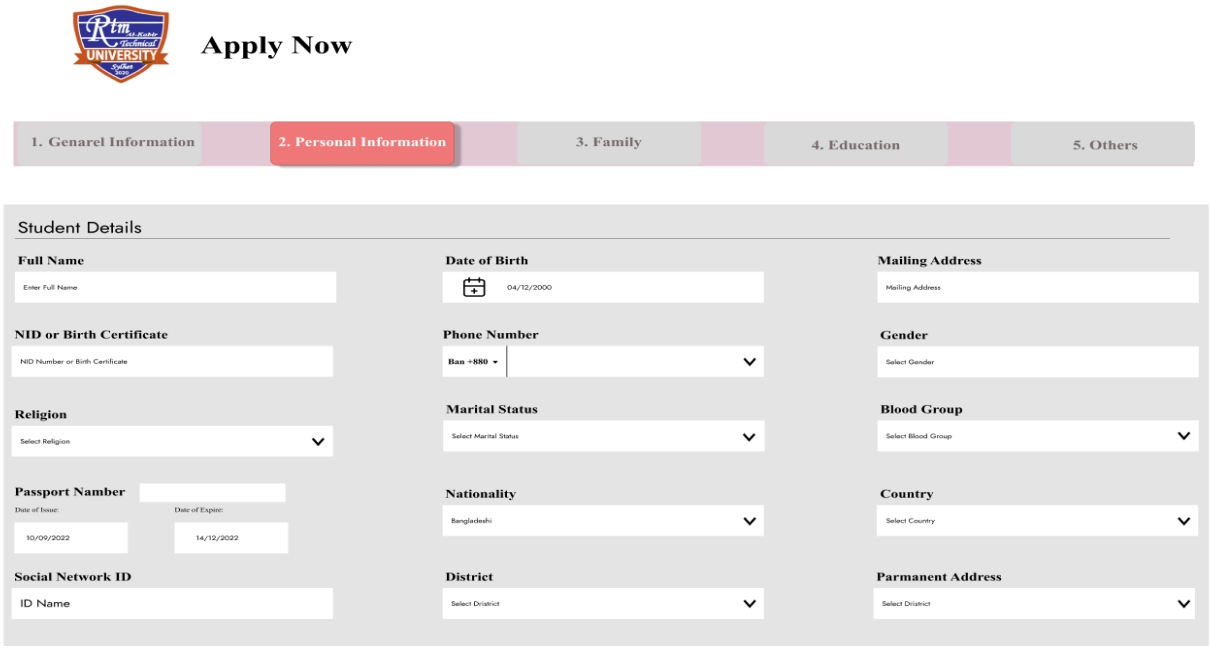


Figure 23: Personal Information

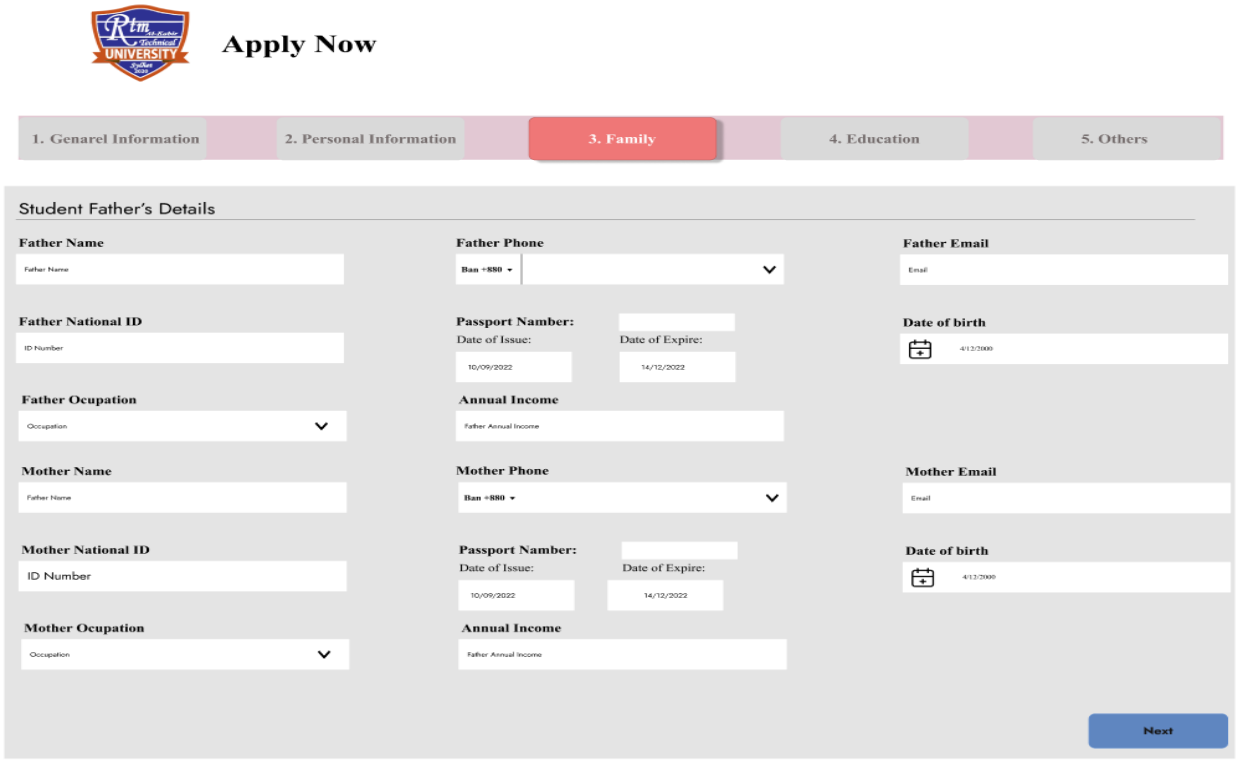


Figure 24: Family Information

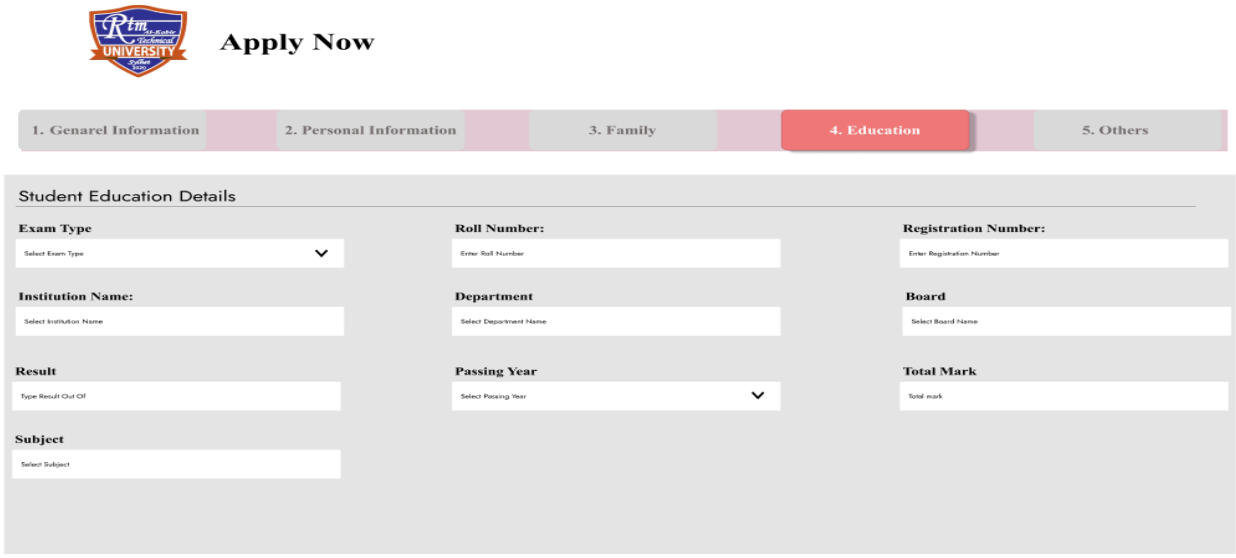


Figure 25: Education Information

Chapter 9 – Engineering

9.1 Approach New System Modules:

Module for logging in

Serial No.	Action of User	Serial No	Action of System
1.	To gain access, the administrator, Teacher, Student will click on the login link.	1.	Before users can use the system, a login form will appear.
2.	To login, the administrator, teacher, and student must input their credentials.	2.	The information entered into the login form is verified/validated by the system.
3.	The login button must be clicked by the administrator, teacher, and student.	3.	The system verifies the administrator's identity. You will be sent to the admin dashboard if you are accepted; otherwise, you will be sent to the login page with an error notice.

Table 19: Module for logging in

Admin Module

Serial No.	Action of User	Serial No	Action of System
1.	The user enters their credentials to authenticate and access the administrative features.	1.	The user can generate various reports related to student enrollment, faculty assignments, course schedules, or any other relevant administrative data.
2.	The user can add, edit, or delete student information, such as personal details, enrollment status, and academic records	2.	The system validates the user inputs for student, faculty, course, and enrollment management to ensure data accuracy and integrity.
3.	The user can add, edit, or delete faculty information, including personal details, qualifications, and teaching	3.	The system updates the database with any changes made by the user, such as adding or modifying student, faculty, course, or

	assignments.		enrollment records.
4.	The user can add, edit, or delete course information, such as course details, prerequisites, and scheduling.	4.	The system validates enrollment requests against prerequisites, availability, and other criteria before approving or rejecting them.

Table 20: Admin University management Module

Teacher's Module

Serial No.	Action of User	Serial No	Action of System
1.	The teacher enters their credentials to authenticate and access the teacher-specific features	1.	The system verifies the teacher's credentials during the login process and grants access to the teacher-specific features.
2.	The teacher can view the courses they are assigned to teach, along with relevant details such as course name, schedule, and enrolled students.	2.	The system retrieves the course assignments for the teacher, providing them with the necessary information about the courses they are assigned to teach.
3.	The teacher can mark attendance for students in their assigned courses and update attendance records as needed.	3.	The system stores the attendance data entered by the teacher, ensuring accurate and up-to-date attendance records for each student.
4.	The teacher can enter and update grades for students in their assigned courses, based on assessments, assignments, and exams.	4.	The system stores the grades entered by the teacher, maintaining accurate and secure grade records for each student in their assigned courses.

Table 21: Teacher's manage Module

Student's Module

Serial No.	Action of User	Serial No	Action of System

1.	The student enters their credentials to authenticate and access the student-specific features.	1.	The system verifies the student's credentials during the login process and grants access to the student-specific features
2.	The student can view their course schedule, including course names, timings, locations, and instructors	2.	The system retrieves and displays the student's course schedule based on their enrolled courses, providing relevant details.
3.	The student can search for available courses, select the desired ones, and enroll in them for the upcoming semester or term.	3.	The system retrieves and displays the student's course schedule based on their enrolled courses, providing relevant details.
4.	The student can access and view their grades for completed courses or ongoing assessments, allowing them to track their academic progress.	4.	The system retrieves and displays the student's course schedule based on their enrolled courses, providing relevant details.

Table 22: Student's manage Module

Admission Module

Serial No.	Action of User	Serial No	Action of System
1.	Prospective students can fill out the admission application form with their personal information, educational background, and other required details.	1.	The system validates the entered data in the application form, ensuring that all necessary information is provided and in the correct format.
2.	Applicants can submit their completed application forms along with any necessary supporting documents.	2.	The system processes and reviews the submitted applications, verifying the eligibility and authenticity of the applicants.
3.	Applicants can make payment for the application fee through the provided payment options.	3.	The system securely processes the application fee payment, confirming the payment status and updating the

			applicant's record.
4.	Applicants can track the status of their admission application, such as whether it is under review, accepted, or rejected.	4.	The system evaluates the applications based on predefined criteria and determines the admission decision (accepted, rejected, or waitlisted).

Table 23: Admission Module

9.2 Use Case Diagram of the IUMS

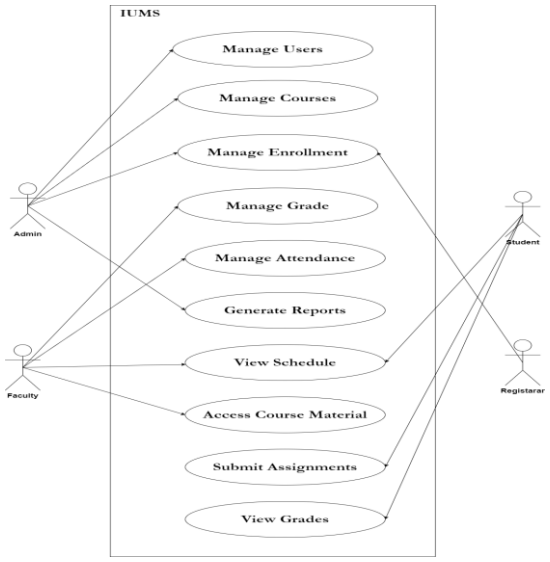


Figure 26: IUMS Use Case

9.3 The IUMS Class Diagram

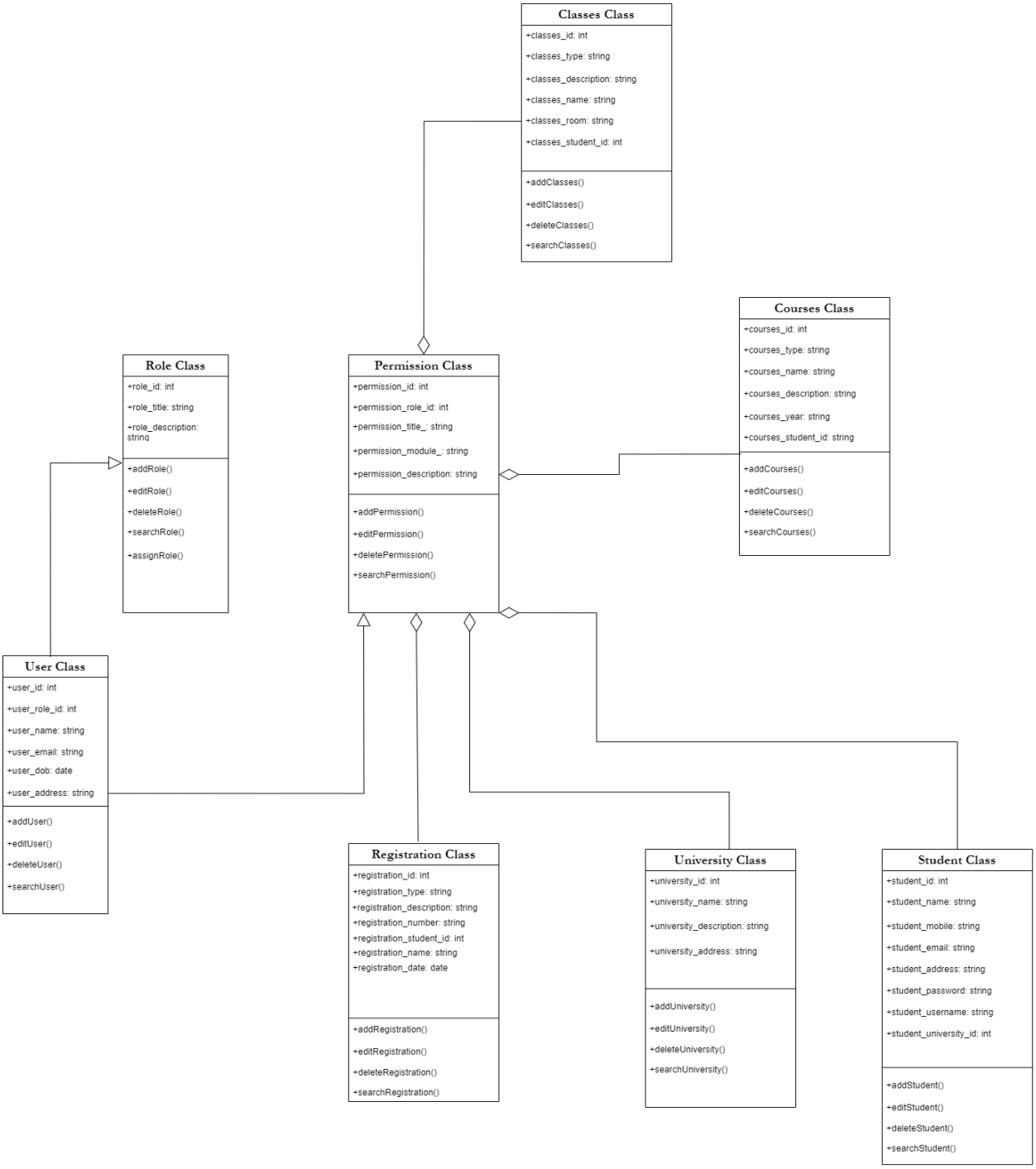


Figure 27: IUMS Class Diagram

9.4 Entity Relationship Diagram

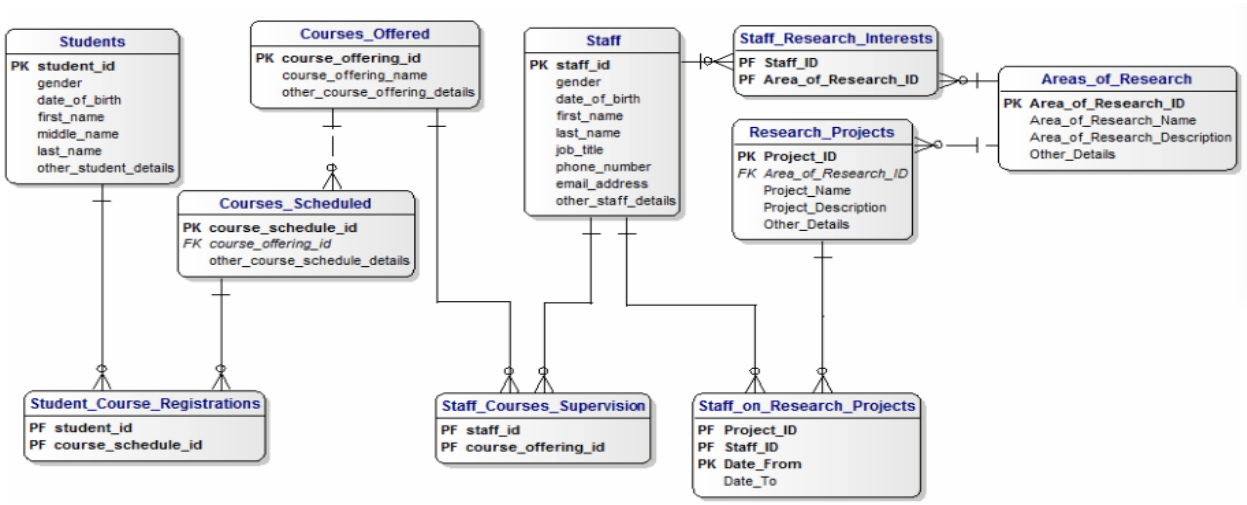


Figure 28: IUMS ERD

9.5 IUMS Sequence Diagram

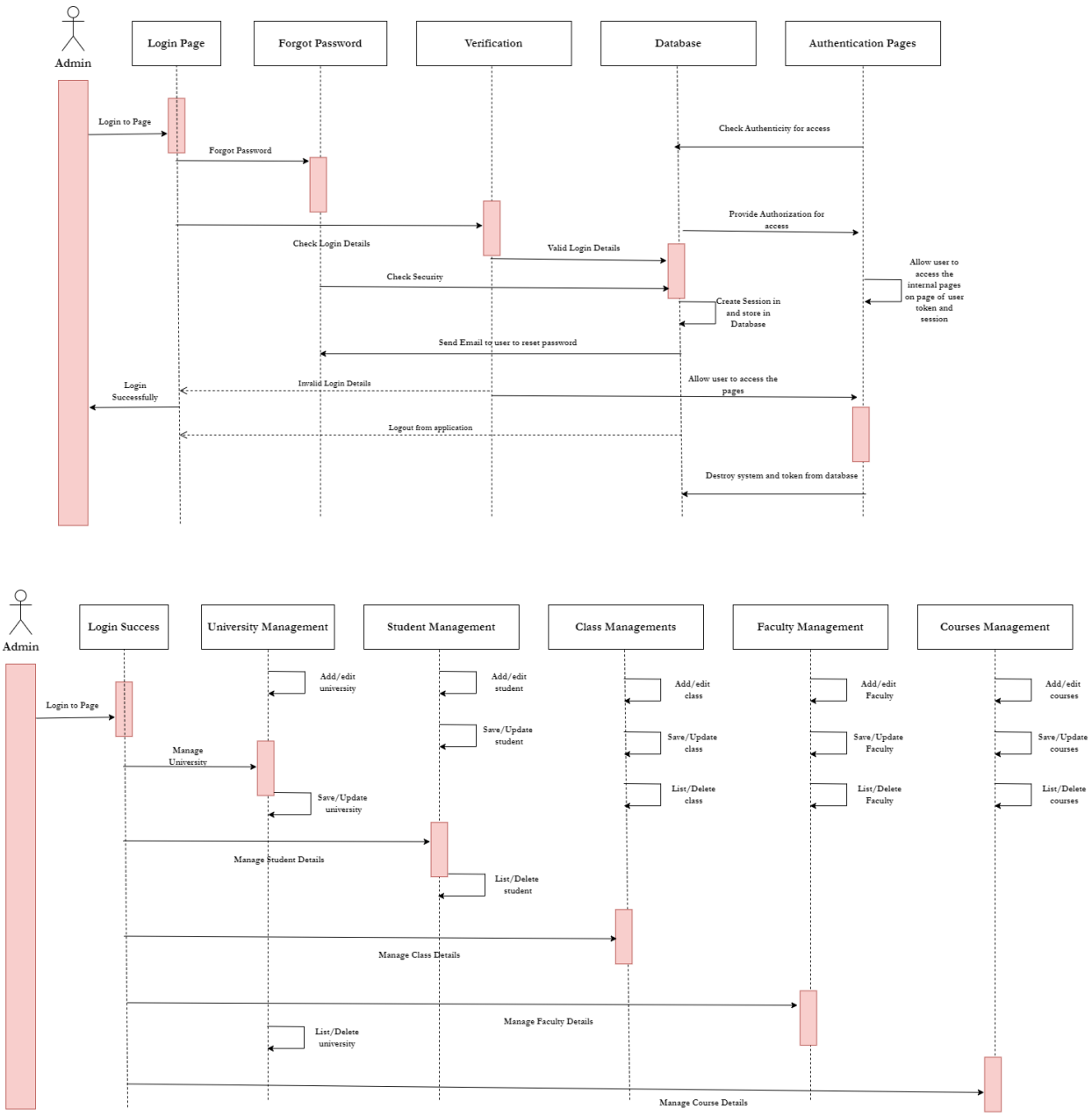


Figure 29: IUMS Sequence Diagram

9.6 The IUMS component diagram

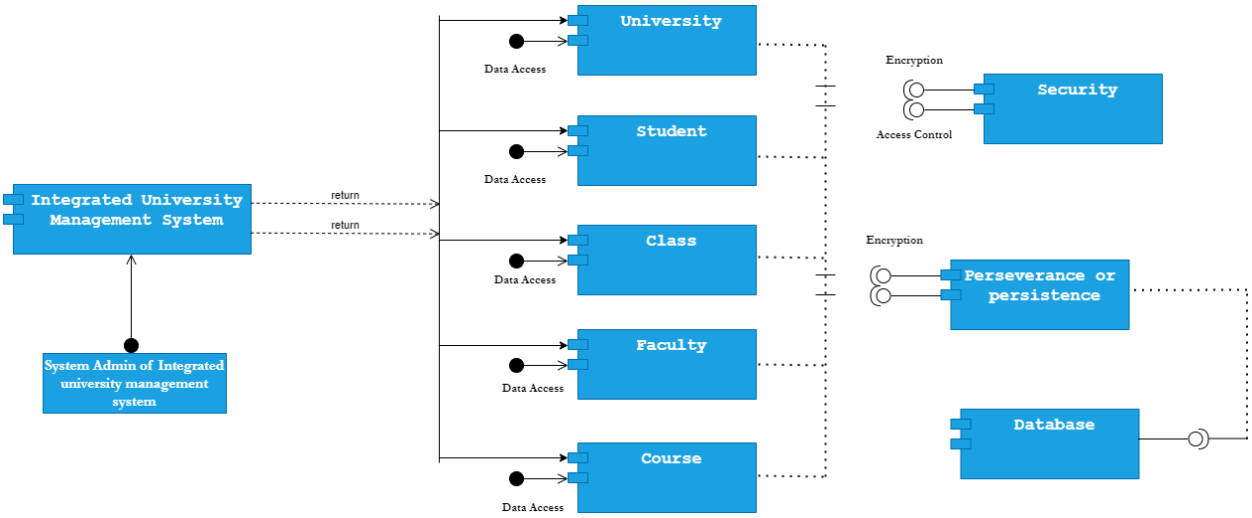


Figure 30: Component Diagram of IUMS

9.7 Deployment Diagram of IUMS

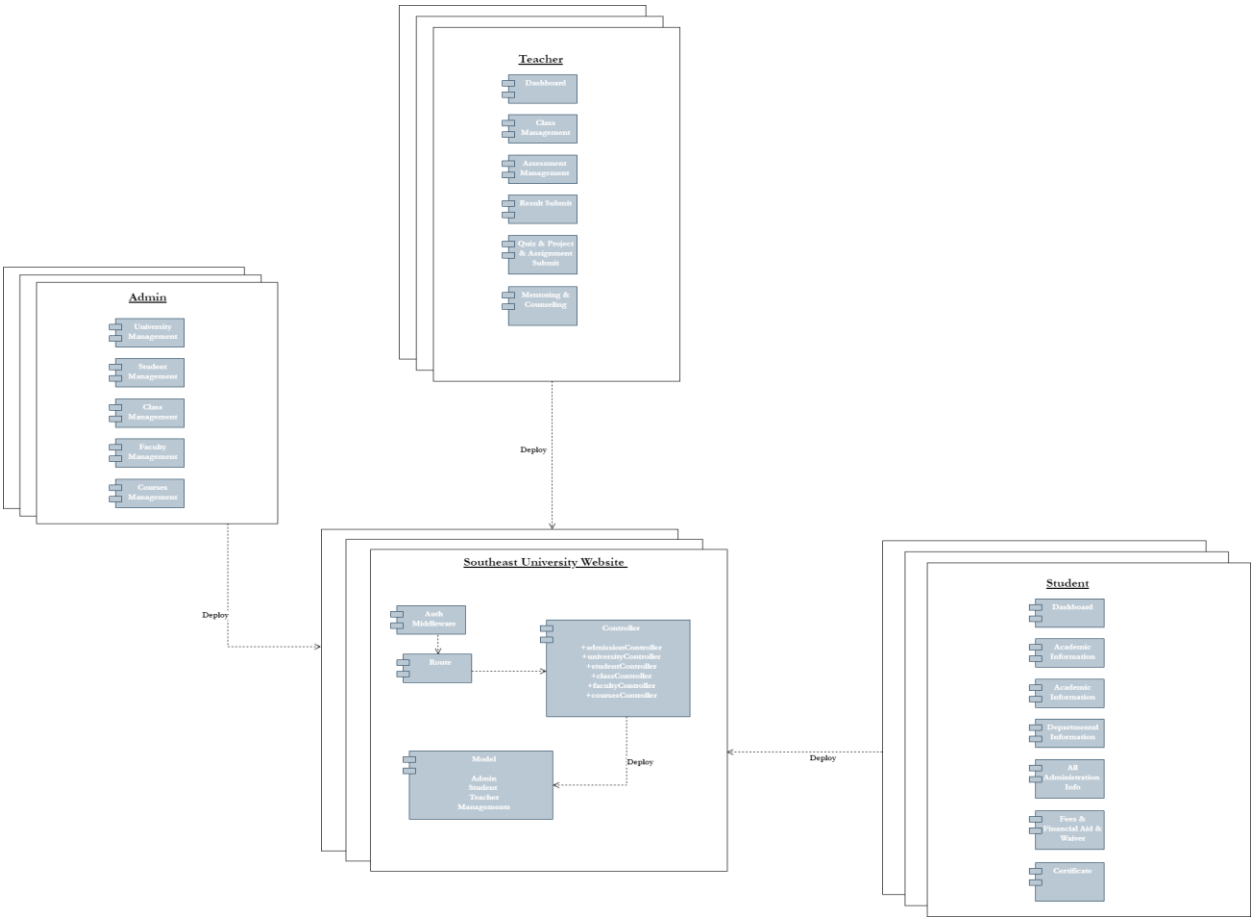


Figure 31: Deployment Diagram of IUMS

9.8 System Interface Design

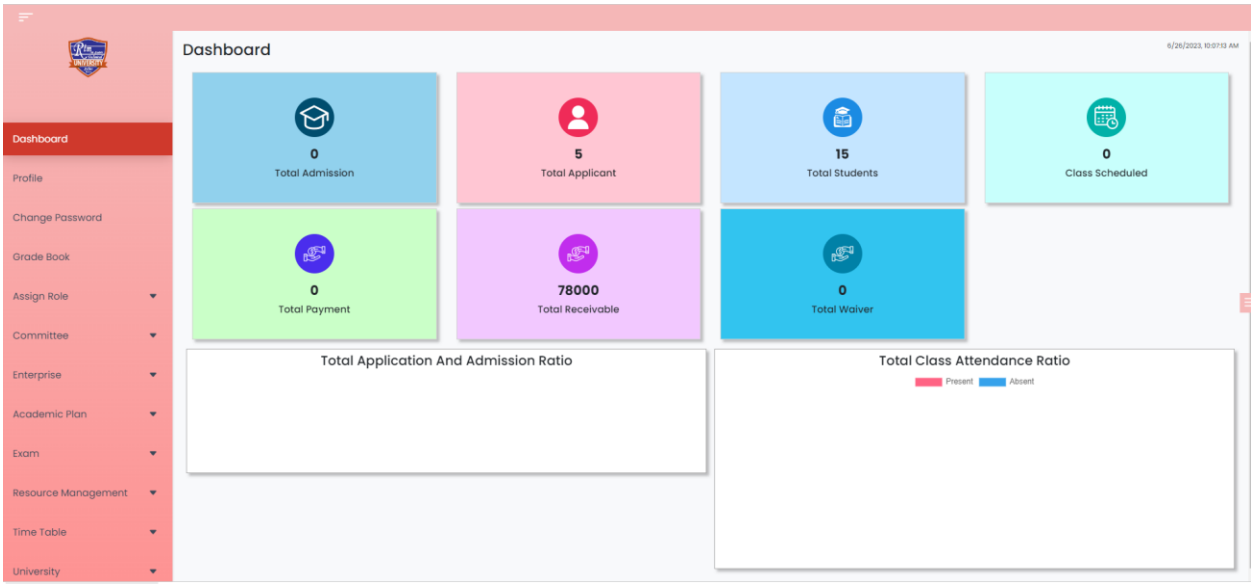


Figure 32: Admin Dashboard

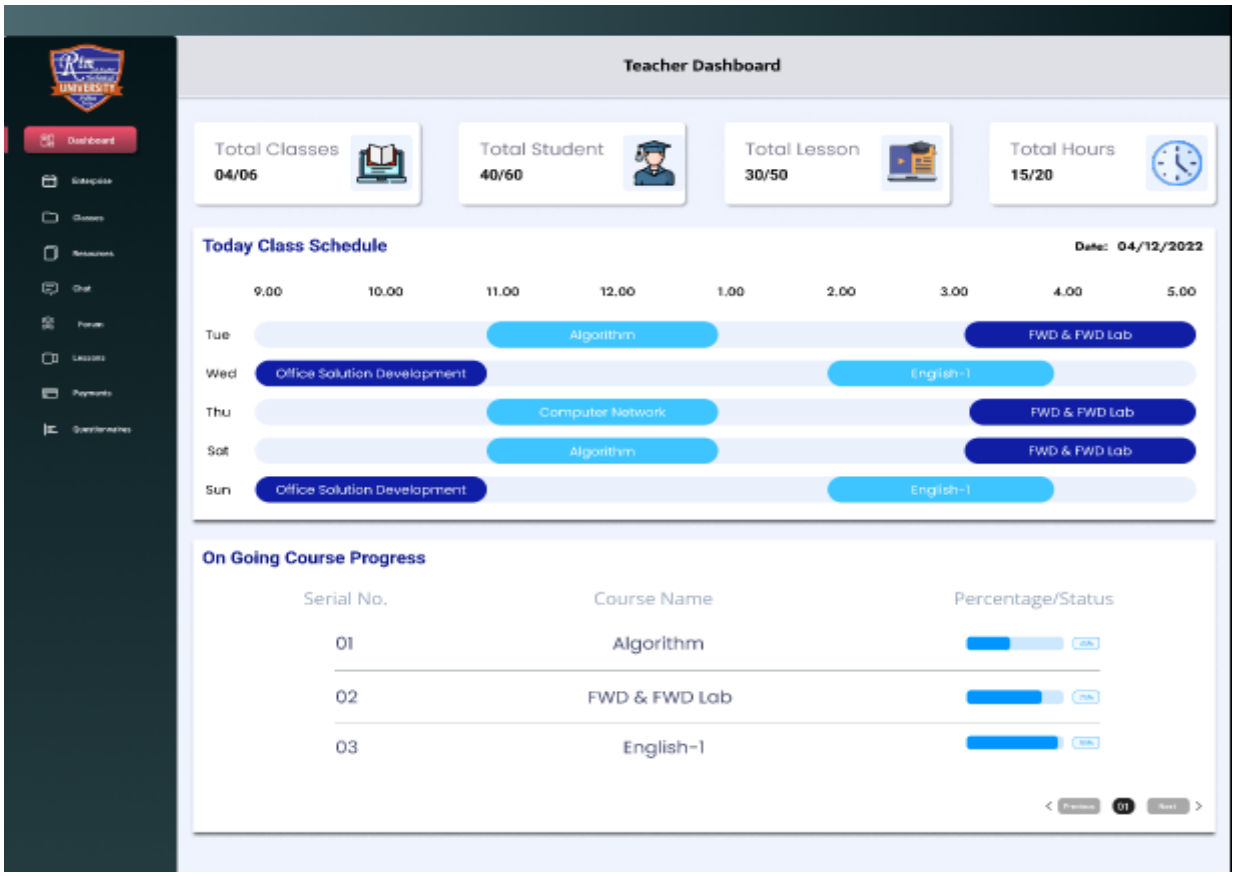


Figure 33: Teacher's Dashboard

Figure 34: Student Dashboard

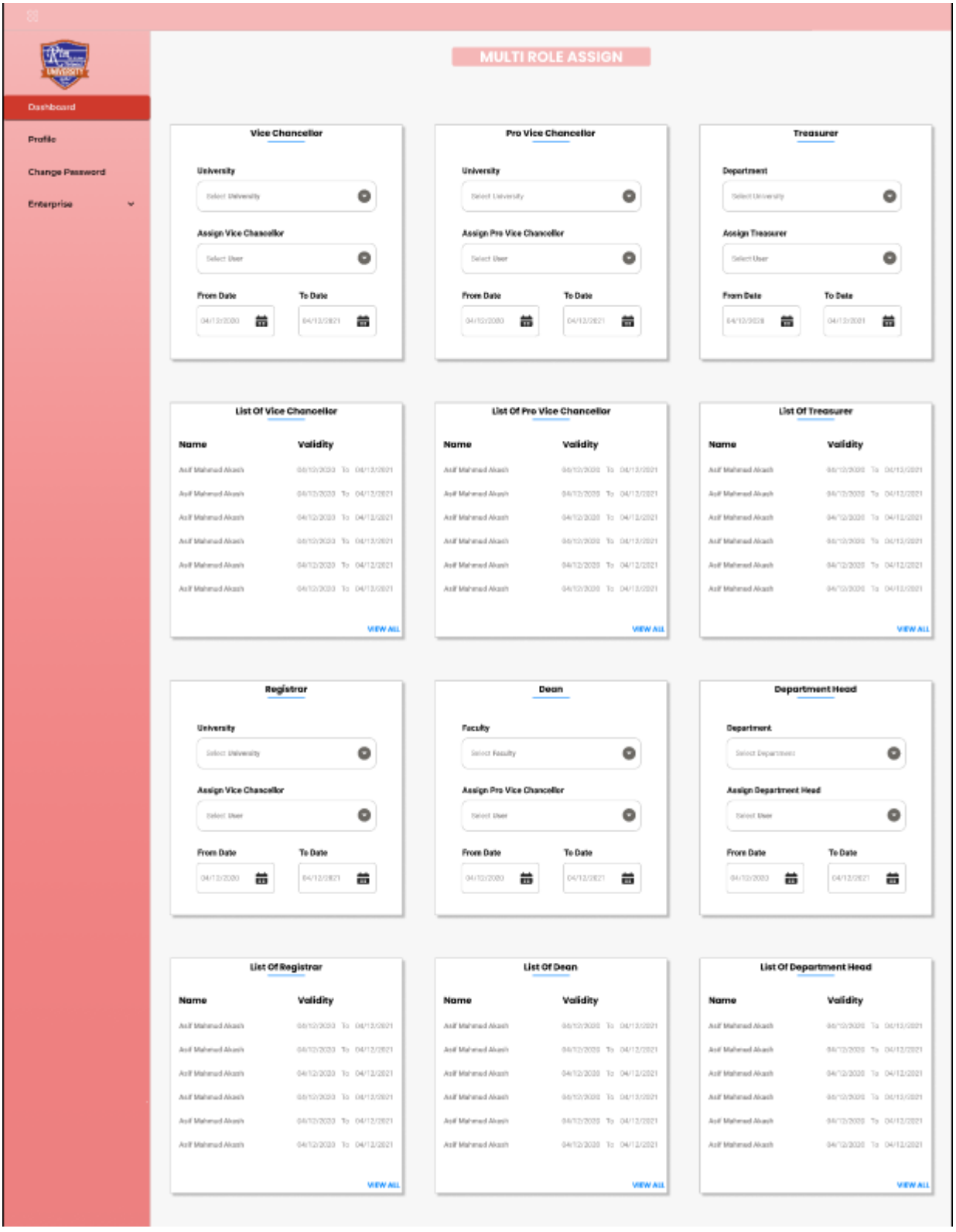


Figure 35: Multi Access Role Assign

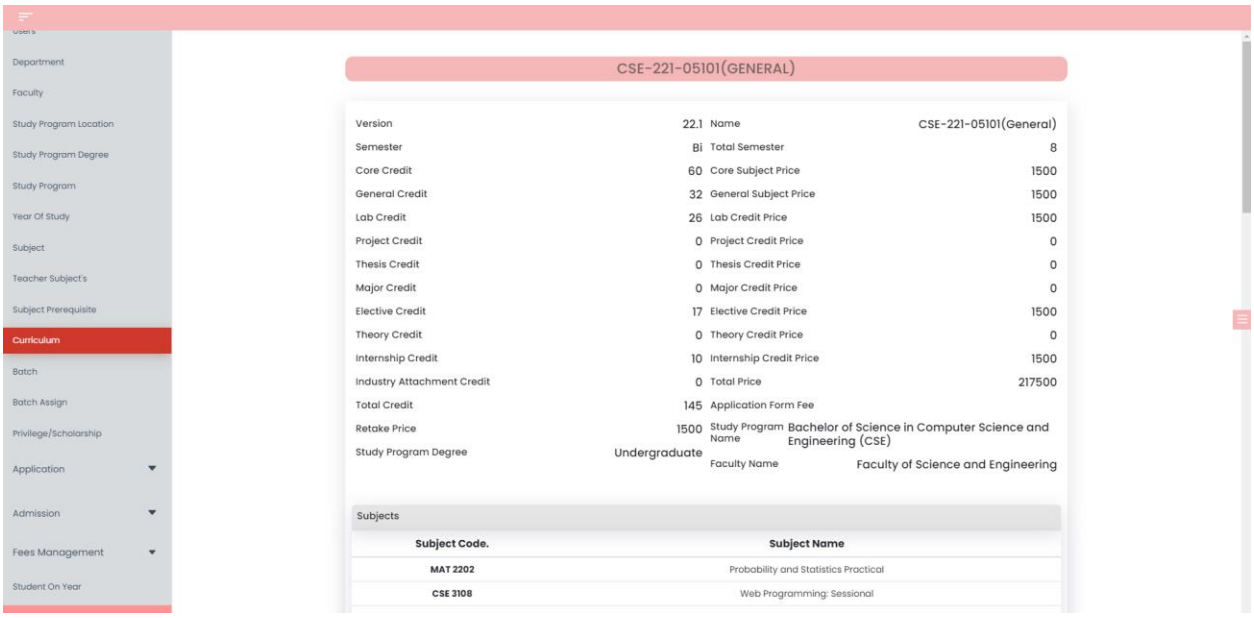


Figure 36: Curriculum

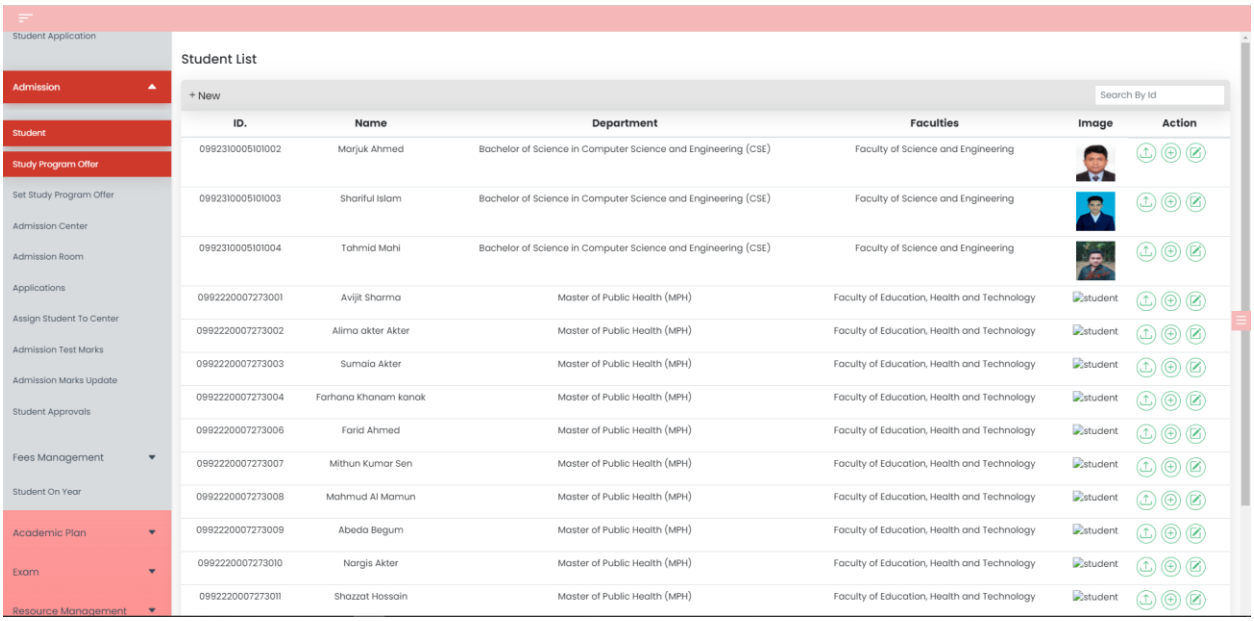


Figure 37: Student List

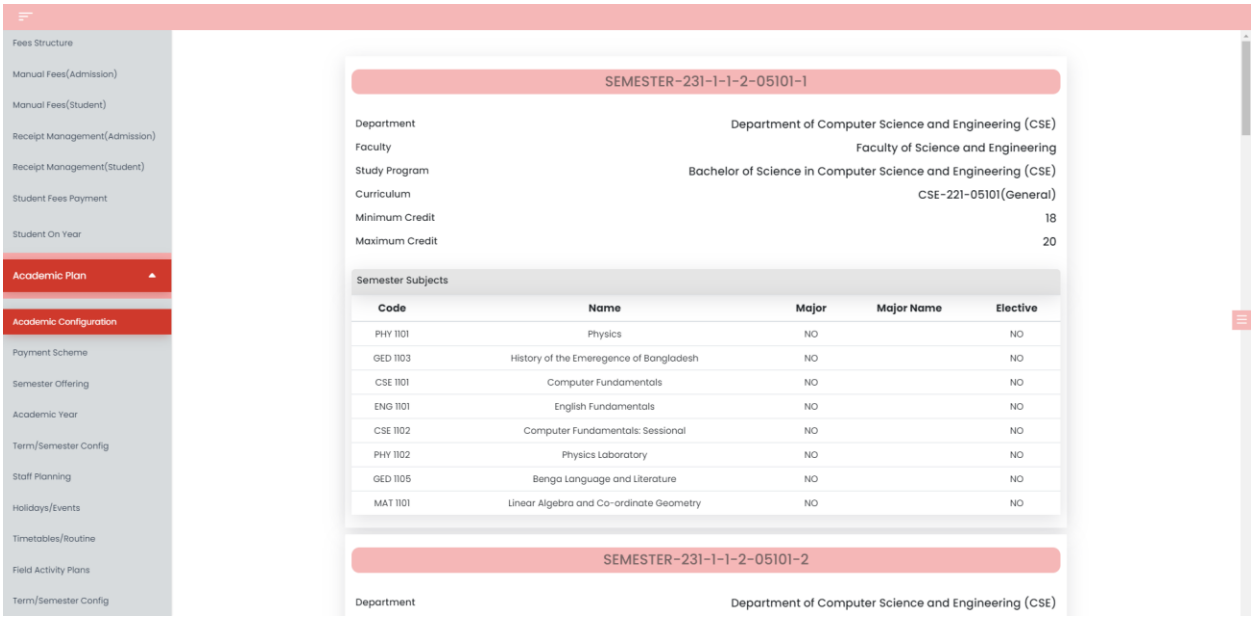


Figure 38: Academic Configuration

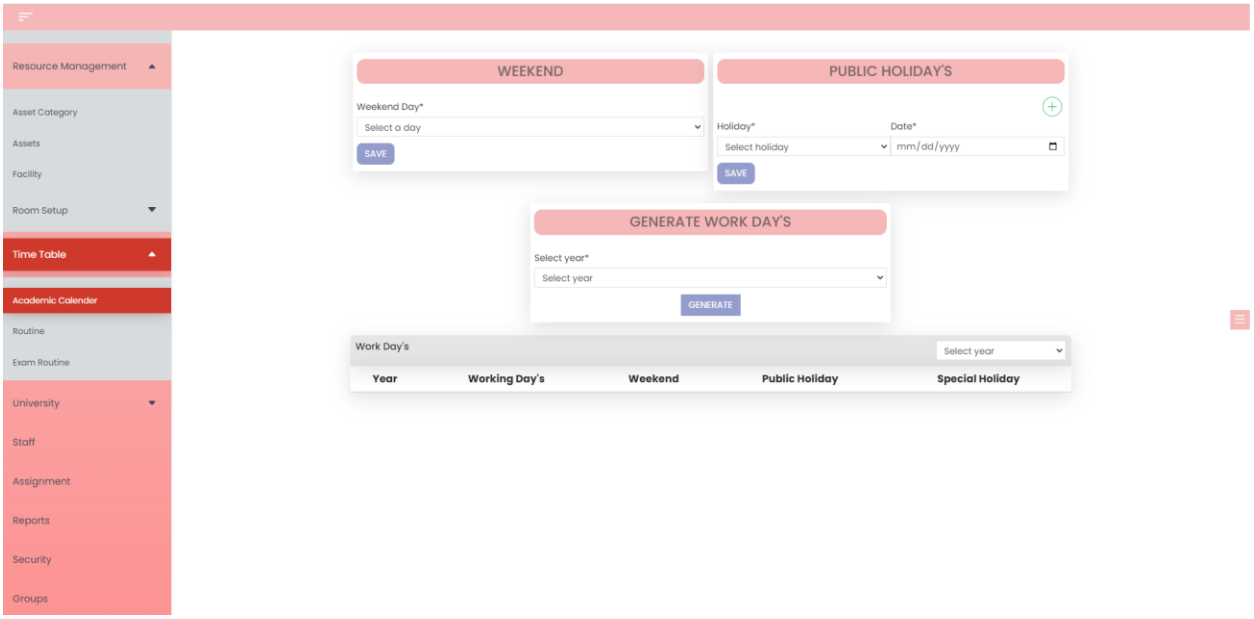


Figure 39: Academic Calendar

Chapter 10 – Deployment

10.1 Core Module Coding Sample:

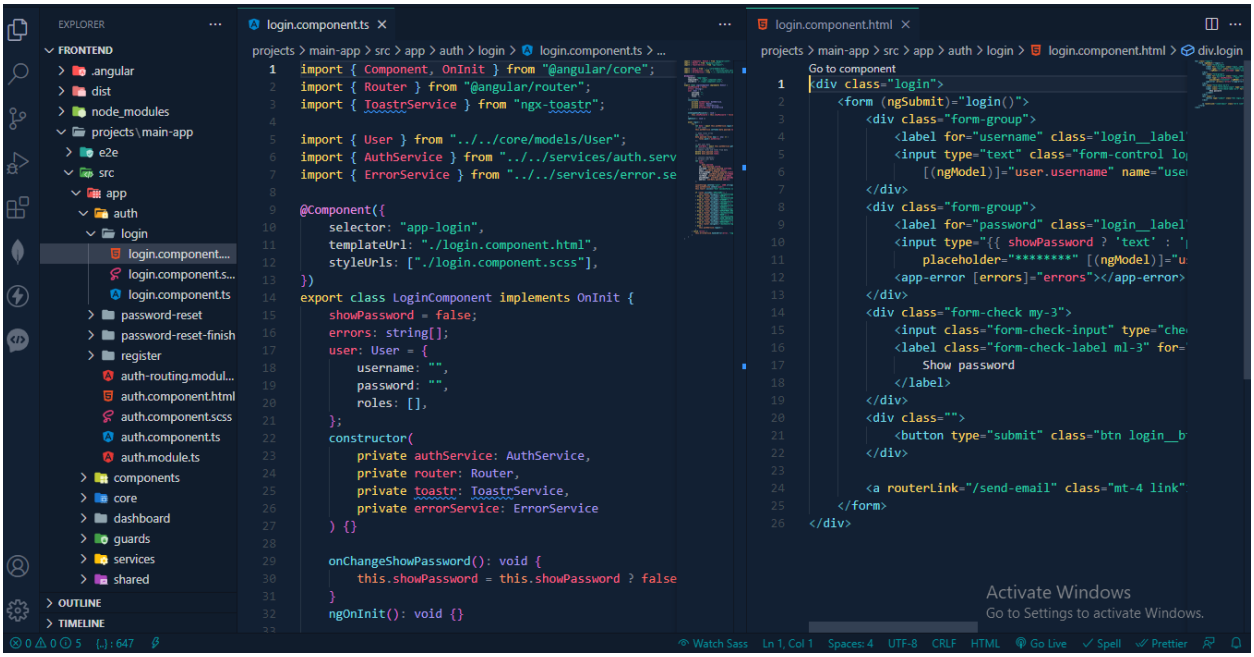


Figure 40: Login Page

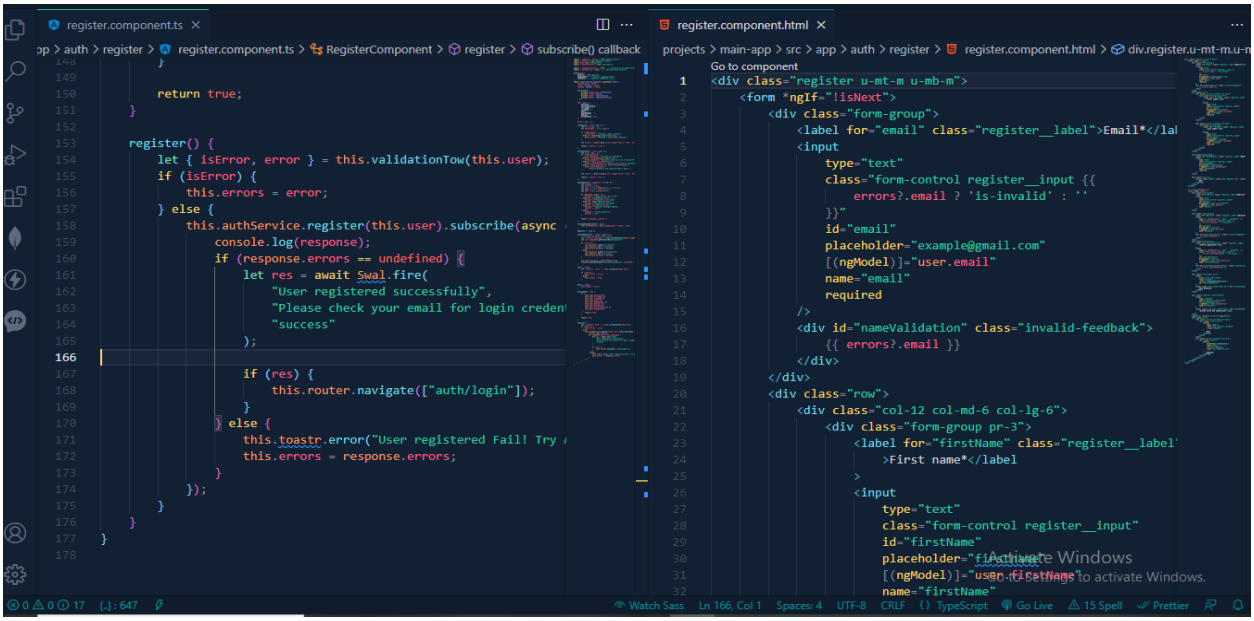


Figure 41: Registration Page

Figure 42: Admin Dashboard

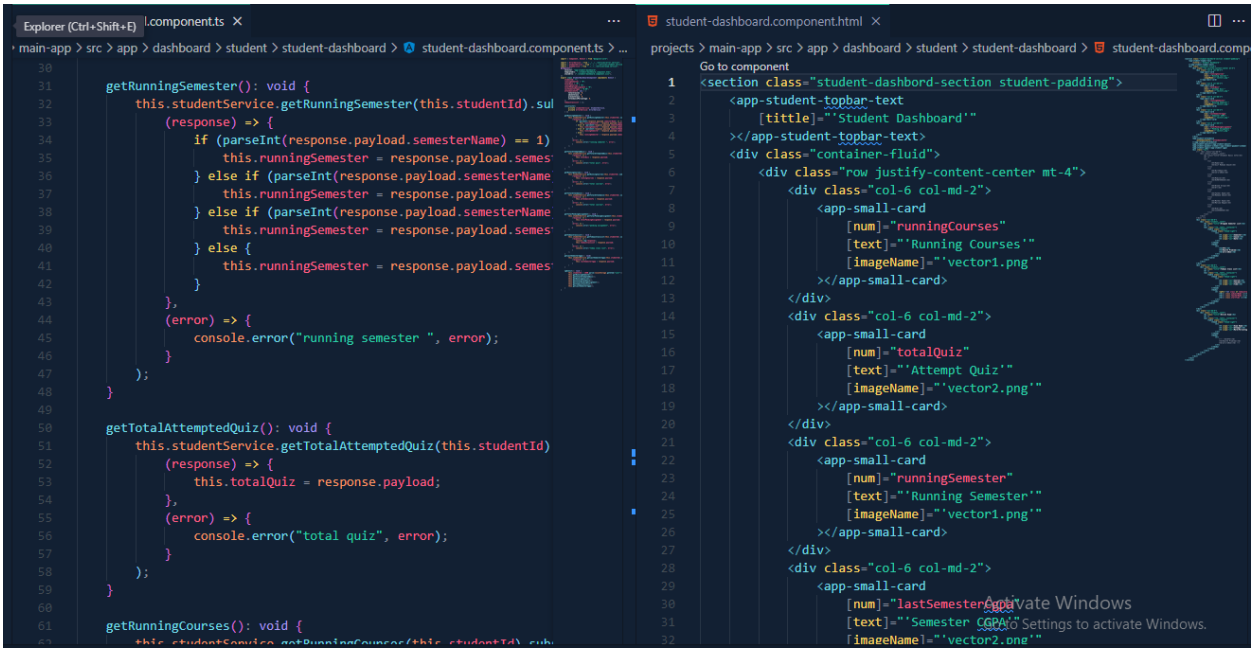


Figure 43: Student Dashboard

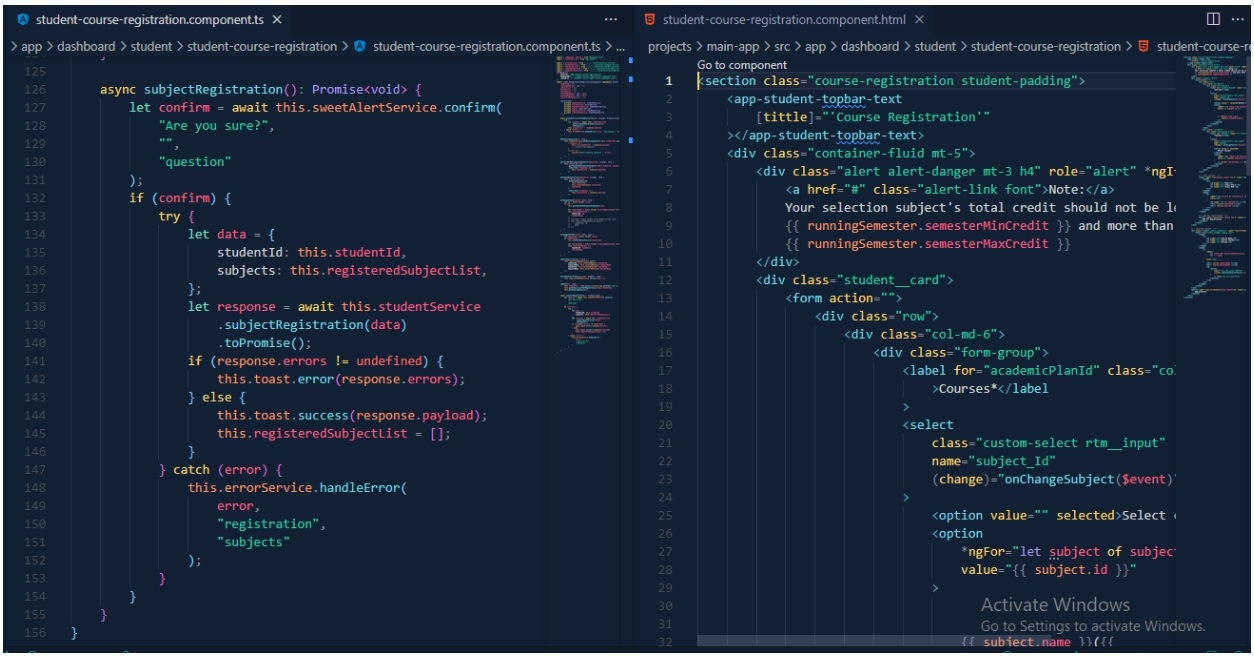


Figure 44: Teacher Course Registration

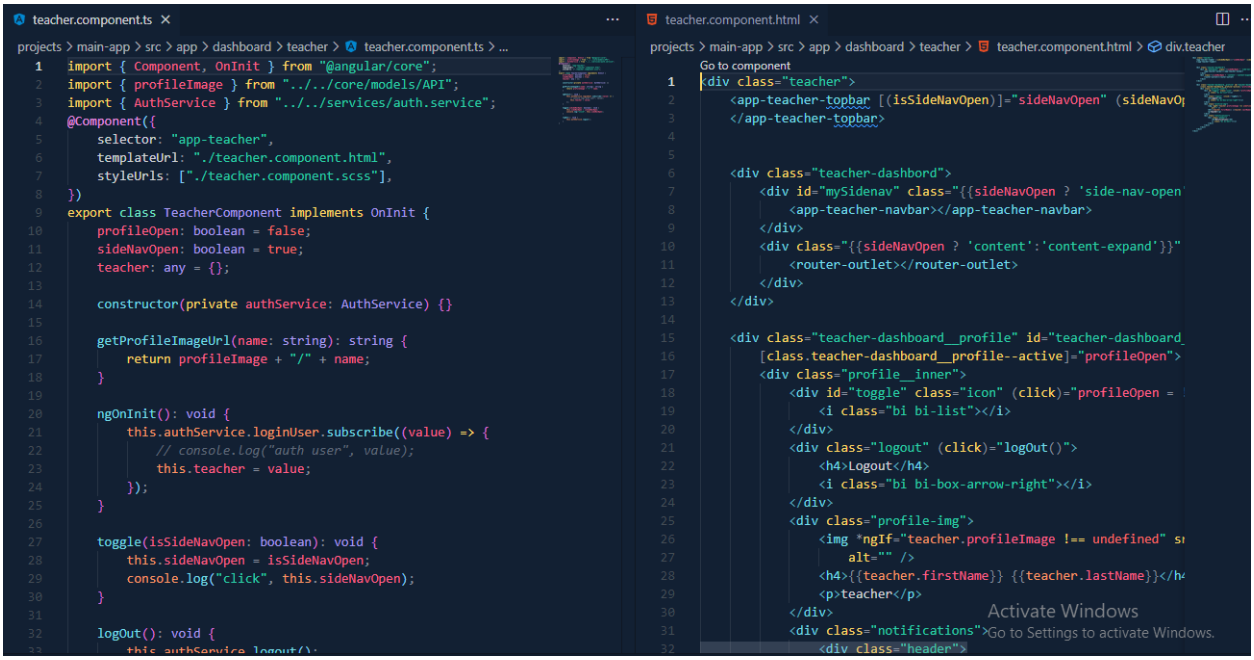


Figure 45: Teacher Module

10.2 Possible Problem Breakdown

When analyzing an Integrated University Management System (IUMS), it's essential to identify potential problem areas or breakdowns that may arise. Here are some possible problem breakdowns that can occur in an IUMS:

System Downtime:

- The IUMS may experience unexpected downtime due to hardware or software failures, network issues, or system upgrades, causing disruption in accessing critical functionalities and data.

Performance Issues:

- The system may face performance bottlenecks during peak usage periods, resulting in slow response times, delays in data retrieval, or system unresponsiveness.

Data Integrity and Security:

- There could be vulnerabilities in data integrity and security, leading to unauthorized access, data breaches, or loss of sensitive information.

Integration Challenges:

- Difficulties may arise when integrating the IUMS with existing university systems or third-party applications, leading to data inconsistencies, interoperability issues, or data synchronization problems.

Usability and User Adoption:

- Users, such as faculty, staff, and students, may encounter challenges in learning and adapting to the new system interface, resulting in low user adoption rates and decreased efficiency.

Data Management and Storage:

- Inadequate data management practices, including inefficient data storage, backup, and archiving strategies, can lead to data loss, corruption, or difficulties in retrieving historical data.

Technical Support and Maintenance:

- Insufficient technical support or delayed system maintenance activities can hinder problem resolution, system updates, bug fixes, and general system stability.

Change Management and Training:

- Inadequate change management strategies and lack of comprehensive training programs for system users can lead to resistance, confusion, and errors during system implementation and upgrades.

Scalability and Flexibility:

- The IUMS may face challenges in scaling up to accommodate growing user and data volumes, or in adapting to changing university requirements and processes.

Identifying these potential problem breakdowns in advance can help stakeholders and project teams develop mitigation strategies, allocate resources appropriately, and proactively address these challenges during the implementation and ongoing management of the IUMS.

10.3 Prioritization while Developing the Solution

When developing a solution for an Integrated University Management System (IUMS), it's crucial to prioritize features and functionalities based on their importance, feasibility, and impact on the university's operations. Here are some factors to consider when prioritizing during the development of an IUMS solution:

Critical System Functionality:

Identify the core functionalities that are essential for the functioning of the university, such as student registration, course management, and financial processes. These should be given the highest priority.

User Needs and Impact:

Understand the needs and requirements of different user groups, including administrators, faculty, staff, and students. Prioritize features that have a significant impact on improving user experience, efficiency, and productivity.

Immediate Pain Points:

Identify and address the immediate pain points or challenges faced by the university. Prioritize features that can alleviate these issues and provide immediate value to the stakeholders.

Regulatory and Compliance Requirements:

Give priority to features and functionalities that ensure compliance with relevant regulations,

Integration and Interoperability:

Prioritize features that facilitate integration with existing systems and applications within the university's IT infrastructure. Consider the potential impact on data consistency, workflows, and user experience across different systems.

Scalability and Future Expansion:

Evaluate the scalability of the solution and prioritize features that can accommodate the university's growth and evolving needs. Consider the long-term vision and roadmap of the IUMS to ensure that the solution can be expanded and enhanced in the future.

Risk Assessment and Mitigation:

Assess potential risks and dependencies associated with different features. Prioritize features that help mitigate critical risks, enhance system stability, and ensure data security.

Incremental Development and Iterative Approach:

Adopt an iterative development approach and prioritize features that can be delivered in incremental releases. This allows for continuous user feedback, validation of assumptions, and course correction during the development process.

It's important to note that prioritization is a collaborative process involving project stakeholders, including university administrators, IT personnel, and end-users. Regular communication, feedback collection, and reassessment of priorities are key to ensuring the solution aligns with the university's goals and effectively meets the needs of its users.

Chapter 11 – Testing

11.1 Test Plan Acceptance

The acceptance description in a test plan outlines the criteria and conditions that must be met for a specific test case or test suite to be considered as "accepted" or successful. It provides a clear definition of what constitutes a passing test and serves as a reference for evaluating the test results.

Functional Testing

The three forms of functional testing are as follows:

Unit Testing:

Unit testing in an Integrated University Management System (IUMS) involves testing individual units or components of the system to ensure they function correctly in isolation. The purpose of unit testing is to verify that each unit of code performs as expected and meets the specified requirements. Here's an overview of unit testing in an IUMS:

Identify Units: Break down the IUMS into smaller units or modules, such as user management, student registration, course management, etc. Each unit should have a well-defined scope and set of responsibilities.

Define Test Cases: Create test cases for each unit, specifying the inputs, expected outputs, and any preconditions or dependencies required. Test cases should cover a range of scenarios and edge cases to ensure thorough coverage.

Set up Test Environment: Set up a test environment that mimics the production environment but allows for isolated testing. This may involve creating mock objects or using stubs or fakes to simulate dependencies.

Write Unit Tests: Develop unit tests using a testing framework (e.g., JUnit for Java) to execute the test cases. Each unit test should focus on testing a single unit of code and its specific functionality.

Test Input/Output: Provide inputs to the unit under test and verify that the outputs match the expected results. This may involve checking return values, state changes, or interactions with other components.

Module Testing:

In the context of IUMS (Integrated University Management System), functional module testing refers to the process of testing individual modules or components within the system to ensure that they are working as intended and meeting the functional requirements.

Here are the general steps involved in functional module testing within an IUMS:

Requirements Analysis: Understand the functional requirements of the module being tested. This includes studying the module's specifications, design documents, and any relevant user stories or use cases.

Test Planning: Create a test plan that outlines the objectives, scope, and test scenarios for the module. Identify the test cases that need to be executed to validate the module's functionality.

Test Case Design: Design test cases that cover different aspects of the module's functionality. Test cases should include inputs, expected outputs, and any specific conditions or prerequisites.

Test Environment Setup: Set up the necessary test environment, including the required hardware, software, and test data. This may involve configuring the IUMS instance with the module under test and any associated dependencies.

Test Execution: Execute the test cases designed in the previous step. Input the specified inputs and compare the actual outputs against the expected outputs. Record any discrepancies or defects encountered during the testing process.

Integration Testing:

Integration testing in IUMS (Integrated University Management System) focuses on testing the interaction between different modules or components within the system. The goal is to ensure that the integrated system functions correctly and meets the requirements specified for its various interconnected parts. Here are the key steps involved in integration testing within an IUMS:

Identify Integration Points: Determine the specific interfaces or points of interaction between the different modules or components in the IUMS. This could include APIs, data exchange mechanisms, message queues, database connections, or any other means of communication.

Integration Test Planning: Develop a test plan that outlines the objectives, scope, and test scenarios for integration testing. Identify the integration points to be tested and define the sequence of testing.

Integration Test Execution: Execute the integration test cases designed to validate the interactions between the modules. Test the flow of data, messages, and operations across the integrated system. Verify that the modules are properly communicating, exchanging data correctly, and adhering to the specified interfaces.

Non-functional Testing

In this project, I want to apply four types of non-functional testing.

Acceptance Testing:

Non-functional acceptance testing in IUMS (Integrated University Management System) focuses on evaluating the system's compliance with non-functional requirements. Non-functional requirements typically define the system's characteristics, such as performance, usability, security, scalability, reliability, and maintainability. Conducting non-functional acceptance testing helps ensure that the IUMS meets these requirements and performs optimally. Here are some key aspects to consider when conducting non-functional acceptance testing in IUMS:

Performance Testing: Evaluate the system's performance by testing its response time, throughput, scalability, and resource usage under expected or maximum load conditions. Measure the system's ability to handle concurrent users, process large volumes of data, and respond within acceptable time frames.

Usability Testing: Assess the system's usability and user-friendliness. Evaluate how intuitive and efficient the user interface is, and whether it meets the needs of different user roles within the university. Test navigation, input validation, error handling, and overall user satisfaction.

Security Testing: Verify that the IUMS maintains a high level of security. Test authentication and authorization mechanisms, data encryption, access controls, and vulnerability management. Identify and address potential security vulnerabilities, ensuring that sensitive data is protected and user accounts are secure.

Reliability and Availability Testing: Test the system's reliability and availability by assessing its ability to handle failures and recover gracefully. Verify backup and restoration procedures, fault tolerance, and system availability during planned and unplanned downtimes. Evaluate how well the system recovers from errors, crashes, or interruptions.

Scalability and Capacity Testing: Determine the system's scalability by testing its ability to handle increasing workloads and growing user bases. Assess the performance of the system as the number of users or data volume increases. Validate that the IUMS can scale up or scale out to accommodate future growth without significant degradation in performance.

Security Testing:

Security testing for an Integrated University Management System (IUMS) is crucial to ensure the protection of sensitive data, maintain user privacy, and prevent unauthorized access or malicious activities. Here are some key aspects to consider when conducting security testing for an IUMS:

Authentication Testing: Verify the effectiveness of the authentication mechanisms in the IUMS. Test different authentication methods such as username/password, multi-factor authentication, and integration with identity providers. Evaluate the strength of password policies, session management, and account lockout mechanisms.

Authorization Testing: Test the system's authorization controls to ensure that only authorized users can access specific functionalities or data. Verify that access controls are enforced correctly based on user roles, permissions, and user groups. Test scenarios where users try to access restricted areas or perform unauthorized actions.

Data Confidentiality and Encryption: Assess the protection of sensitive data within the IUMS. Test if data is encrypted during transmission over networks using secure protocols such as HTTPS. Evaluate how the system handles encryption of stored data, including user credentials, personal information, and financial records.

Input Validation and Security: Test the IUMS for vulnerabilities related to input validation. Check for potential security risks such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Verify that the system properly validates and sanitizes user inputs to prevent these types of attacks.

Session Management: Evaluate the security of session management within the IUMS. Test session timeouts, session fixation vulnerabilities, and session hijacking. Verify that session tokens are securely generated, transmitted, and invalidated when the user logs out or after a period of inactivity

Accessibility Testing:

Accessibility testing is a crucial aspect of software testing, ensuring that the Integrated University Management System (IUMS) is usable and accessible to individuals with disabilities. It focuses on evaluating the system's compliance with accessibility standards and guidelines, enabling equal access and usability for all users. Here are some key considerations for conducting accessibility testing in an IUMS:

Color Contrast and Visual Design: Evaluate the color contrast ratios within the IUMS to ensure readability for individuals with visual impairments. Test for appropriate color combinations, including text color against background color, and ensure that information is not conveyed solely through color. Evaluate the visual design for clarity and legibility.

Text Resizing and Zooming: Verify that users can resize text or zoom in without causing content to overlap or become inaccessible. Test the scalability of text and interface elements to ensure that they remain usable at different zoom levels or font size adjustments.

Usability Testing:

Usability testing is a critical part of evaluating the user experience and overall usability of an Integrated University Management System (IUMS). It focuses on gathering feedback from users to identify any usability issues, improve the system's design, and ensure that it meets the needs of its intended users. Here are the key steps involved in usability testing for an IUMS:

Define Test Objectives: Clearly outline the objectives and goals of the usability testing. Determine the specific aspects of the IUMS that need to be evaluated, such as navigation, task completion, information architecture, or overall user satisfaction.

Identify User Profiles: Define the target user profiles for the IUMS. Identify representative users, including students, faculty members, administrators, and other relevant stakeholders. Consider factors such as their technical expertise, familiarity with similar systems, and specific needs or preferences.

Create Test Scenarios: Develop realistic and representative test scenarios that reflect common tasks and activities performed within the IUMS. These scenarios should cover a range of functionalities and

workflows. Ensure that the scenarios are clear, concise, and aligned with the objectives of the usability testing.

Recruit Test Participants: Recruit participants who match the identified user profiles. Aim for a diverse group of participants to gather varied perspectives. The number of participants will depend on the scope of the usability testing, but typically, a minimum of 5-8 participants is recommended for identifying major usability issues.

Conduct Usability Sessions: Conduct individual usability sessions with the participants. Provide clear instructions for each scenario and encourage participants to think aloud, expressing their thoughts, feedback, and difficulties encountered during the testing process. Observe their interactions with the IUMS and take note of their feedback.

11.2 Test Case

Test cases must be prepared when the test acceptance strategy has been completed. The IUMS system's test cases are listed below.

Unit test – test case:

Name of the test case	**Unit testing**
Test Class	
Description of the test	

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**

Module Test – test case:

Name of the test case	**Module Test**

Test Class	
Description of the test	

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**

Integration Testing – test case:

Name of the test case	**Integration Test**
Test Class	
Description of the test	

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**

11.3 Unit Testing

Test Case

Name of the test case	**Unit Test**
Test Class	Create a test class that corresponds to the class or method being tested in the IUMS.
Description of the test	The test class will contain test methods that exercise the functionality and behavior of the specific unit under test.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**
Determine the source of data for unit testing. This may involve creating test data directly in the test class or using mock objects or test fixtures to simulate the required data for the unit being tested.	Write test methods that execute specific scenarios and exercise the desired functionality of the unit being tested. Each test method should follow a standardized structure, including the following steps: a. Setup: Prepare the necessary preconditions and test data required for the test case. This may involve initializing objects, setting up dependencies, and configuring the test environment. b. Execution: Invoke	Define the expected outcome for each test case. This includes determining the correct behavior, output, or state that the unit under test should exhibit based on the specific scenario being tested. The expected outcome should be documented and known before running the test.	After executing each test case, compare the actual outcome with the expected outcome. If the actual outcome matches the expected outcome, the test case is considered successful. Otherwise, if the actual outcome differs from the expected outcome, it indicates a potential issue or bug in the unit being tested.

	the specific method or functionality being tested within the IUMS. Pass the test data and any required parameters.		

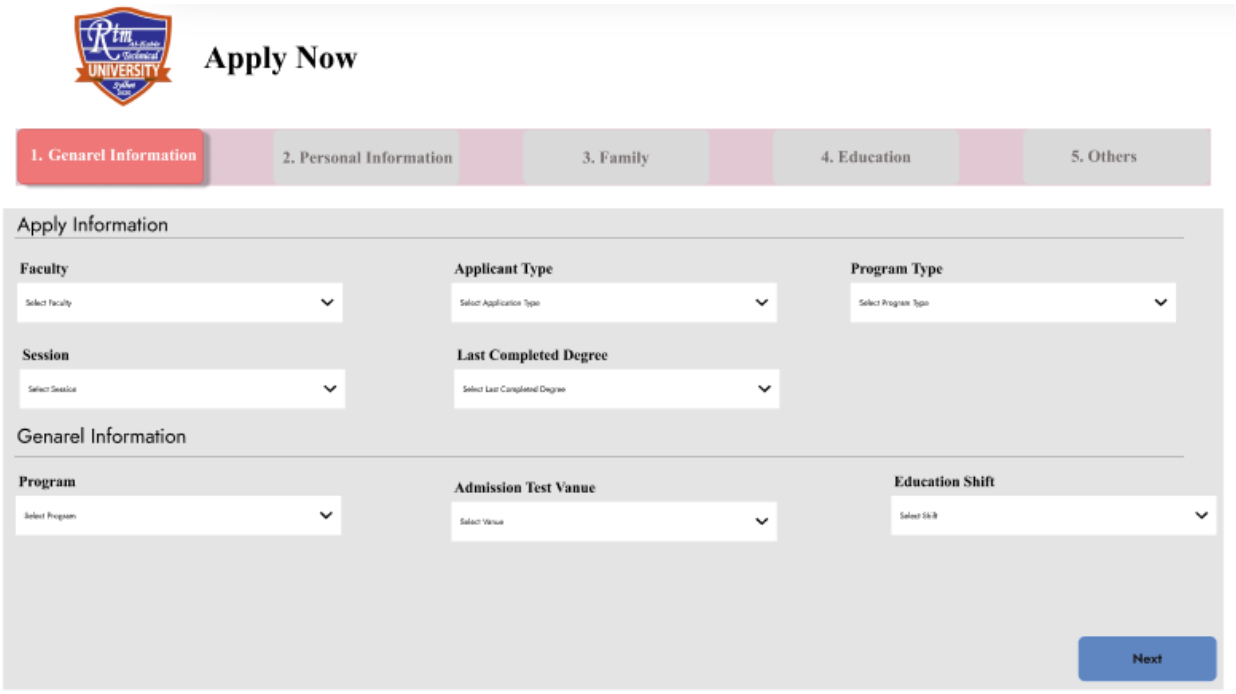


Figure 46: unit test one test case

Name of the test case	**Unit Test**
Test Class	Create a test class that corresponds to the class or method being tested in the IUMS.
Description of the test	The test class will contain test methods that exercise the functionality and behavior of the specific unit under test.

Source of Data	**Steps in the Testing**	**Expected Outcome**	**Actual Outcome**

	Process		
Determine the source of data for unit testing. This may involve creating test data directly in the test class or using mock objects or test fixtures to simulate the required data for the unit being tested.	Write test methods that execute specific scenarios and exercise the desired functionality of the unit being tested. Each test method should follow a standardized structure, including the following steps: a. Setup: Prepare the necessary preconditions and test data required for the test case. This may involve initializing objects, setting up dependencies, and configuring the test environment. b. Execution: Invoke the specific method or functionality being tested within the IUMS. Pass the test data and any required parameters.	Define the expected outcome for each test case. This includes determining the correct behavior, output, or state that the unit under test should exhibit based on the specific scenario being tested. The expected outcome should be documented and known before running the test.	After executing each test case, compare the actual outcome with the expected outcome. If the actual outcome matches the expected outcome, the test case is considered successful. Otherwise, if the actual outcome differs from the expected outcome, it indicates a potential issue or bug in the unit being tested.

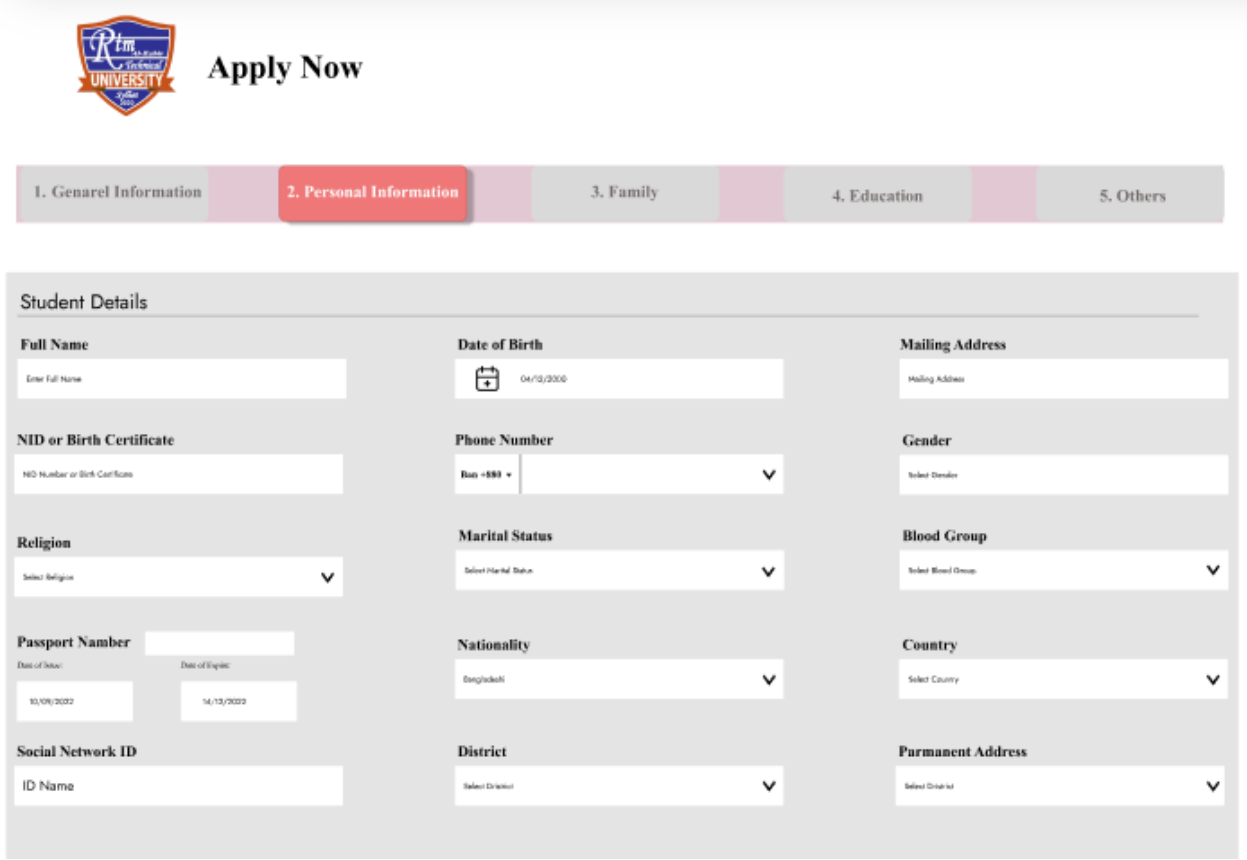


Figure 47: Unit test two result

Unit Test three

Test Cas

Name of the test case	**Unit Test**
Test Class	CourseServiceTest
Description of the test	This test case aims to verify the functionality of the CourseService class when enrolling a student in a course.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**

For this example, we will use mock objects to simulate the necessary data for the CourseService class.	Setup: Create a test instance of the CourseService class and any required dependencies or mock objects. Execution: Invoke the specific method being tested, such as enrollStudentInCourse() , within the CourseService class. Pass the necessary parameters and test data. Assertion: Compare the actual outcome with the expected outcome using assertions.	The expected outcome of the testEnrollStudentInCourse() method is that the student is successfully enrolled in the course. The enrollmentSuccess flag is set to true, and the mock database's enrollStudentInCourse() method is called. Additionally, the student is added to the list of enrolled students for the course.	During the test, the enrollStudentInCourse() method is executed, and the enrollmentSuccess flag is set to true. The mock database's enrollStudentInCourse() method is also called, indicating a successful enrollment. Finally, the student is added to the list of enrolled students for the course.

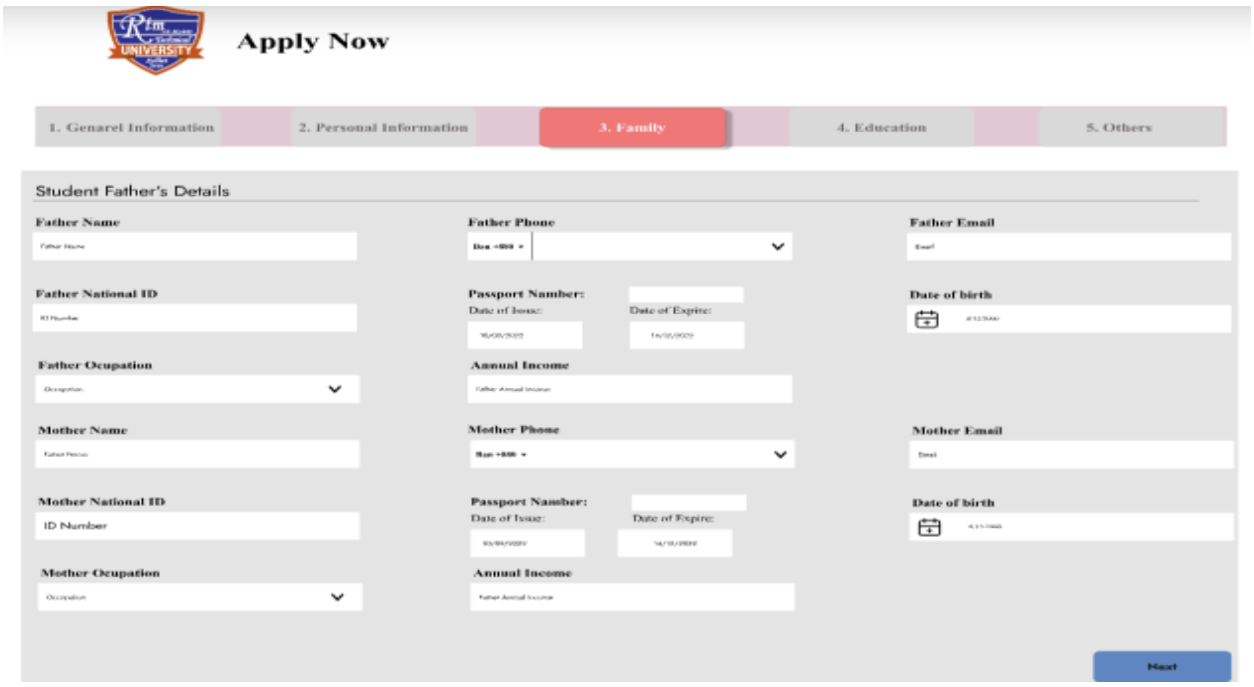


Figure 48: Unit Test three

Unit Test four

Name of the test case	**Unit Test**
Test Class	EnrollmentServiceTest
Description of the test	This test case aims to verify the functionality of the EnrollmentService class when withdrawing a student from a course.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**
For this example, we will use mock objects to simulate the necessary data for the CourseService class.	Setup: Create a test instance of the EnrollmentService class and any required dependencies or mock objects. Execution: Invoke the specific method being tested, such as withdrawStudentFromCourse(), within the EnrollmentService class. Pass the necessary parameters and test data.	The expected outcome of the testWithdrawStudentFromCourse() method is that the student is successfully withdrawn from the course. The withdrawalSuccess flag is set to true, and the mock database's withdrawStudentFromCourse() method is called. Additionally, the student is removed from the list of enrolled students for the course.	During the test, the withdrawStudentFromCourse() method is executed, and the withdrawalSuccess flag is set to true. The mock database's withdrawStudentFromCourse() method is also called, indicating a successful withdrawal. Finally, the student is removed from the list of enrolled students for the course.

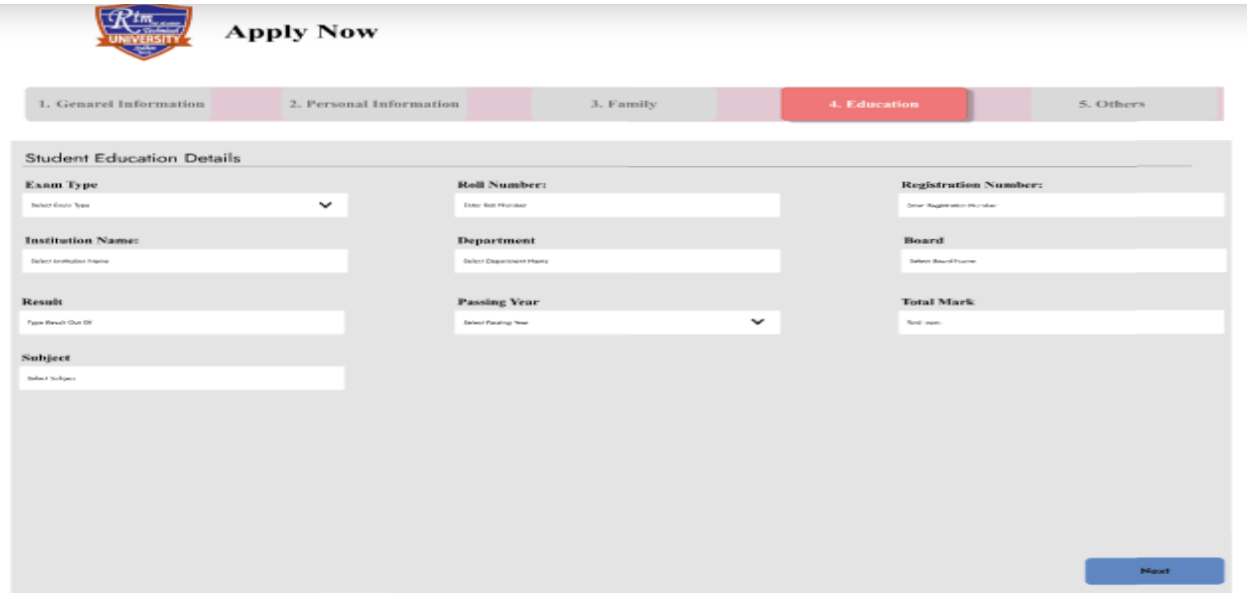


Figure 49: unit test four test case

11.4 Module Testing

Module Test one

Test Case

Name of the test case	**Module Testing**
Test Class	EnrollmentModuleTest
Description of the test	This test case aims to verify the functionality of the enrollment module in the IUMS, specifically the process of enrolling a student in a course.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**
For this example, we will use mock objects to simulate the necessary data for the enrollment	Setup: Create a test instance of the enrollment module and any required	The expected outcome of the testEnrollStudentInCourse() method is that the	During the test, the enrollStudentInCourse() method of the enrollment module is

module.	dependencies or mock objects. Execution: Invoke the specific method being tested, such as enrollStudentInCourse(), within the enrollment module. Pass the necessary parameters and test data. Assertion: Compare the actual outcome with the expected outcome using assertions.	student is successfully enrolled in the course. The enrollmentSuccess flag is set to true, and the mock course service's enrollStudentInCourse() method is called. Additionally, the mock student service's updateEnrollmentStatus() method is called to update the enrollment status of the student.	executed, and the enrollmentSuccess flag is set to true. The mock course service's enrollStudentInCourse() method is also called, indicating a successful enrollment. The mock student service's updateEnrollmentStatus() method is called to update the enrollment status of the student.

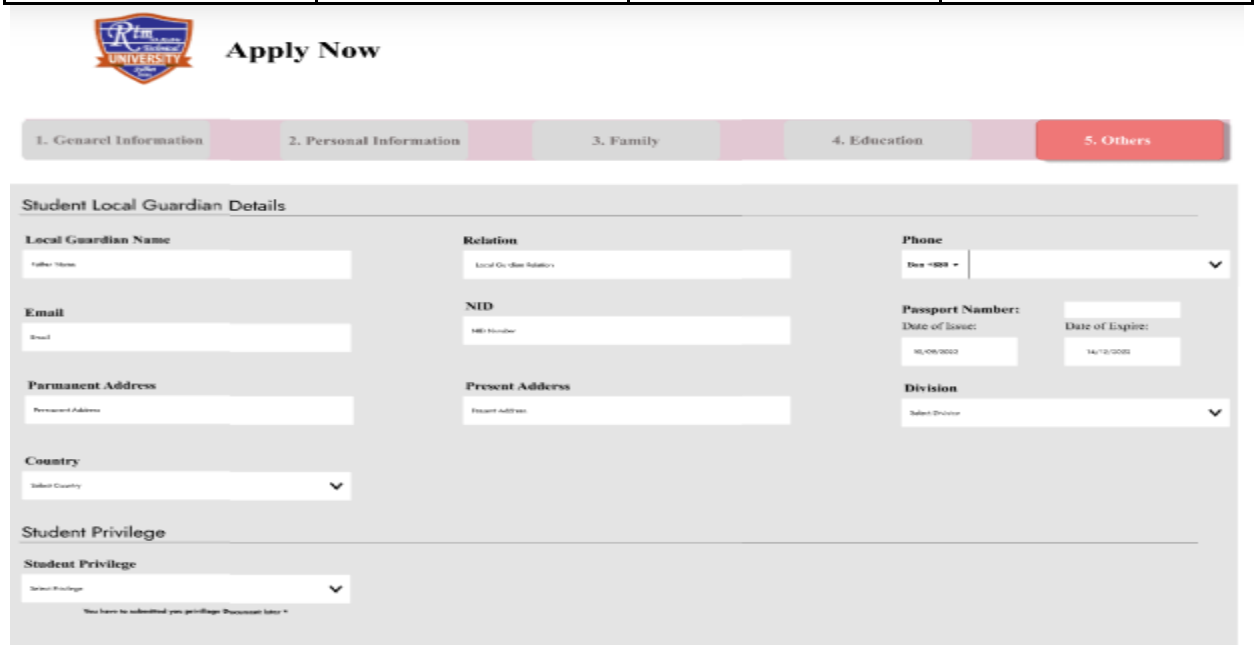


Figure 50: Module Test One

Module Test two

Test Case

Name of the test case	**Module Testing**
Test Class	CourseRegistrationModuleTest

Description of the test	This test case aims to verify the functionality of the course registration module in the IUMS, specifically the process of registering a student for a course.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**
For this example, we will use mock objects to simulate the necessary data for the course registration module.	Setup: Create a test instance of the course registration module and any required dependencies or mock objects. Execution: Invoke the specific method being tested, such as registerCourseForStudent(), within the course registration module. Pass the necessary parameters and test data. Assertion: Compare the actual outcome with the expected outcome using assertions.	The expected outcome of the testRegisterCourseForStudent() method is that the student is successfully registered for the course. The registrationSuccess flag is set to true, and the mock course service's addStudentToCourse() method is called to add the student to the course. Additionally, the mock student service's addCourseToStudent() method is called to update the student's course list.	During the test, the registerCourseForStudent() method of the course registration module is executed, and the registrationSuccess flag is set to true. The mock course service's addStudentToCourse() method is also called, indicating a successful registration. The mock student service's addCourseToStudent() method is called to update the student's course list.

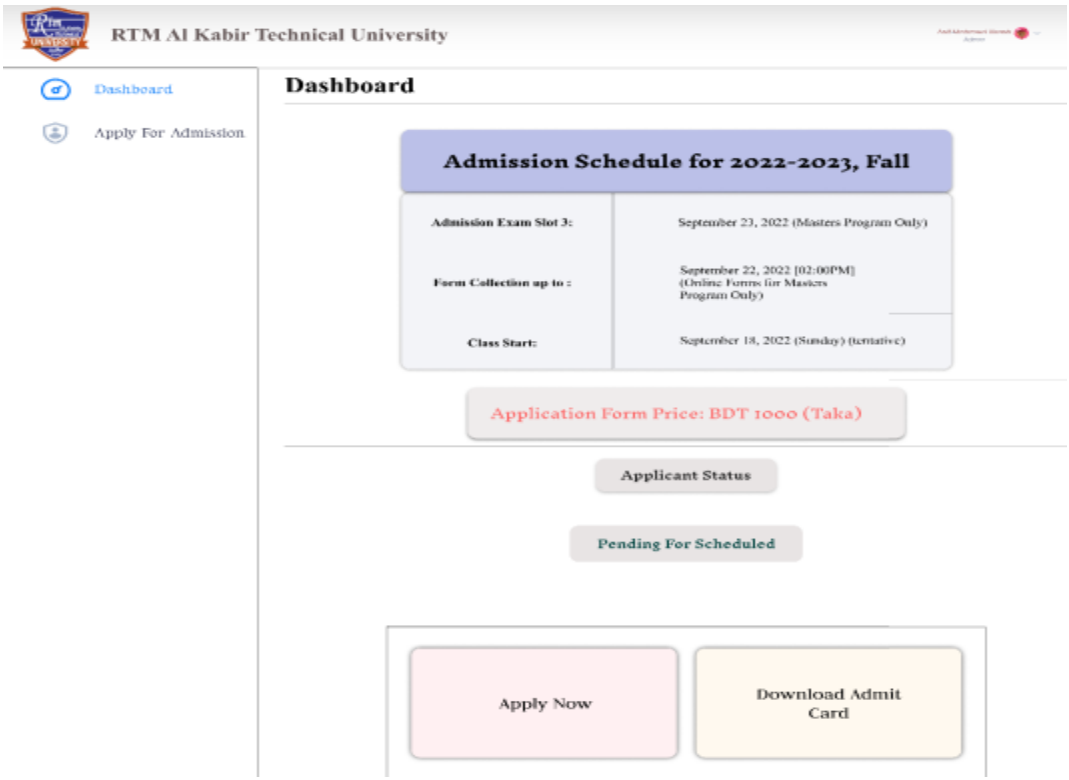


Figure 51: Module Test two

11.5 Integration Testing

Integration Test one

Test Case

Name of the test case	**Integration Testing**
Test Class	StudentCourseIntegrationTest
Description of the test	This test case aims to verify the integration between the StudentService and CourseService classes in the IUMS, specifically the process of enrolling a student in a course.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**

For this example, we will use mock objects to simulate the necessary data for the integration test.	Setup: Create a test instance of the StudentService and CourseService classes along with any required dependencies or mock objects. Execution: Invoke the specific method being tested, such as enrollStudentInCourse() , which involves the interaction between the StudentService and CourseService classes. Pass the necessary parameters and test data. Assertion: Compare the actual outcome with the expected outcome using assertions.	The expected outcome of the testEnrollStudentInCourseIntegration() method is that the student is successfully enrolled in the course. The enrollmentSuccess flag is set to true, and the mock database's enrollStudentInCourse() method is called. Additionally, the student is added to the list of enrolled students for the course.	During the test, the enrollStudentInCourse() method of the StudentService class is executed, and the enrollmentSuccess flag is set to true. The enrollStudentInCourse() method in the mock database is also called, indicating a successful enrollment. Finally, the student is added to the list of enrolled students for the course.

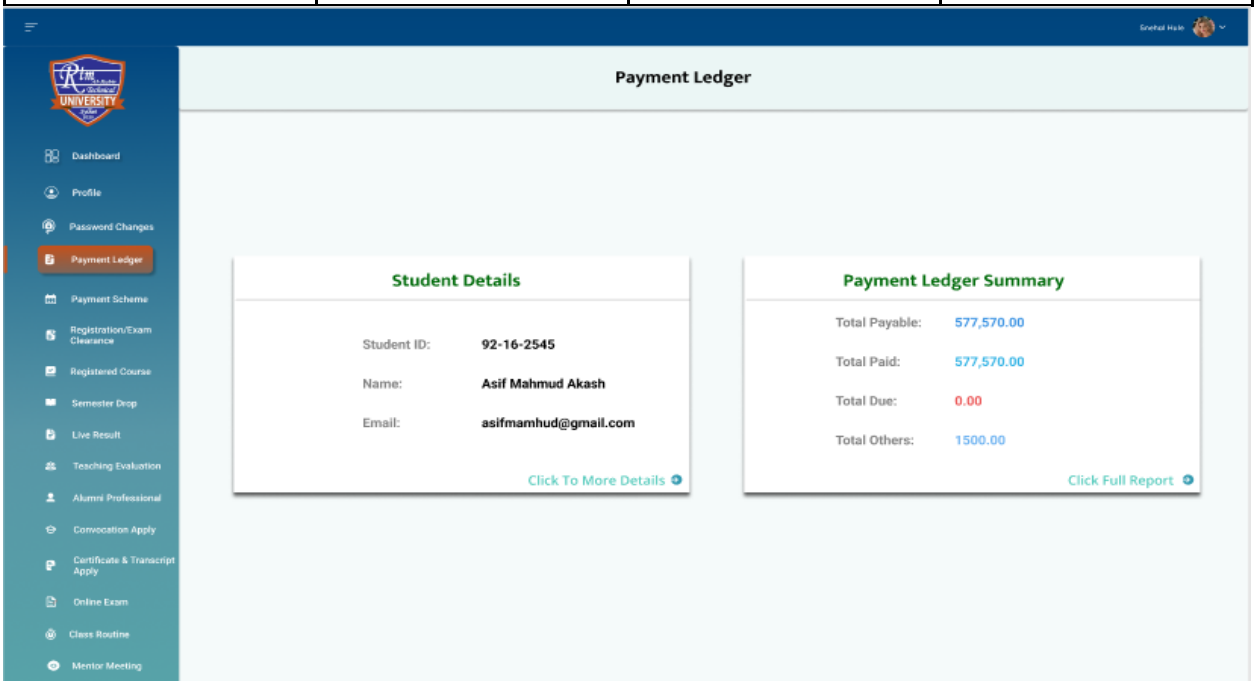


Figure 52: Integration Test one

Integration Test two

Test Case

Name of the test case	**Integration Testing**
Test Class	CourseEnrollmentIntegrationTest
Description of the test	This test case aims to verify the integration between the EnrollmentService and CourseService classes in the IUMS, specifically the process of enrolling a student in a course.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**
For this example, we will use mock objects to simulate the necessary data for the integration test.	Setup: Create a test instance of the EnrollmentService and CourseService classes along with any required dependencies or mock objects. Execution: Invoke the specific method being tested, such as enrollStudentInCourse() , which involves the interaction between the EnrollmentService and CourseService classes. Pass the necessary parameters and test data. Assertion: Compare the actual outcome with the	The expected outcome of the testEnrollStudentInCourseIntegration() method is that the student is successfully enrolled in the course. The enrollmentSuccess flag is set to true, and the mock database's enrollStudentInCourse() method is called. Additionally, the student is added to the list of enrolled students for the course.	During the test, the enrollStudentInCourse() method of the EnrollmentService class is executed, and the enrollmentSuccess flag is set to true. The enrollStudentInCourse() method in the mock database is also called, indicating a successful enrollment. Finally, the student is added to the list of enrolled students for the course.

	expected outcome using assertions.		

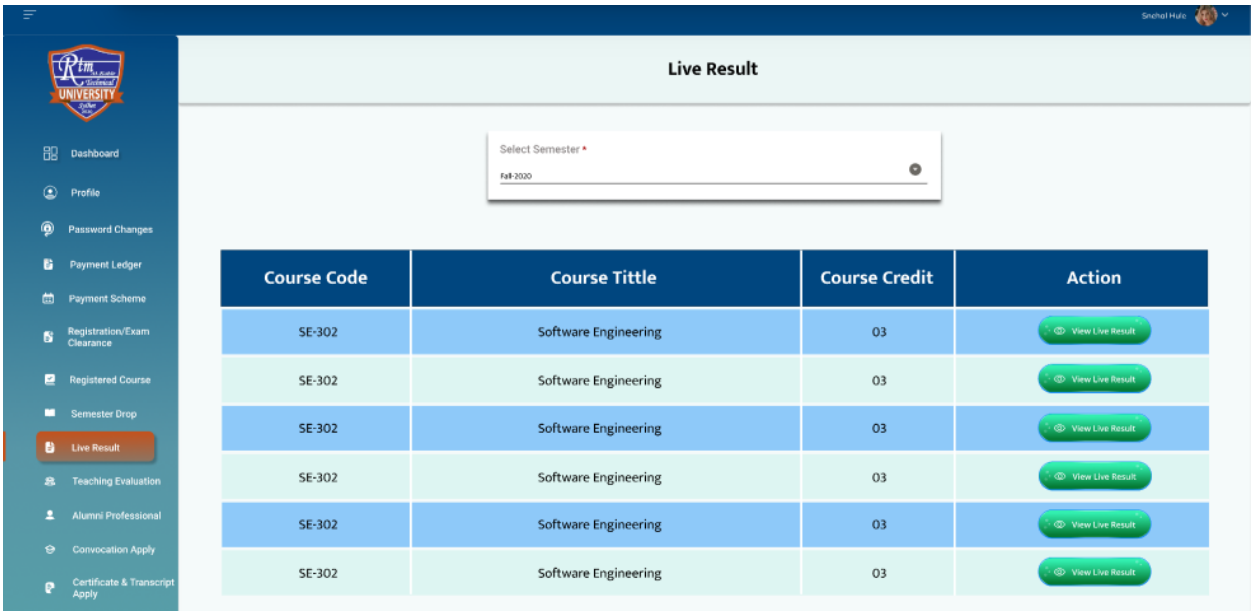


Figure 53: Integration Test two

11.6 Acceptance Testing

Acceptance Test one

Test Case

Name of the test case	**Acceptance Testing**
Test Class	EnrollmentAcceptanceTest
Description of the test	This test case aims to validate the enrollment process from the perspective of a user or stakeholder of the IUMS. It verifies that a student can successfully enroll in a course using the system's user interface.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**

For this example, we will use test data specifically created for the acceptance test.	Setup: Ensure that the IUMS is deployed and accessible with the necessary test data populated, including students and available courses. Execution: Simulate the actions of a user to enroll in a course. This involves navigating to the enrollment page, selecting a course, and submitting the enrollment request. Assertion: Compare the actual outcome with the expected outcome based on the system's response and the updated enrollment status.	The expected outcome of the testEnrollmentProcess() method is that the enrollment process is successful, indicated by a positive system response. Additionally, the enrollment status of the student for the selected course should be updated to "Enrolled."	During the test, the user navigates to the enrollment page, selects the course "Math101," and submits the enrollment request. The test asserts that the enrollment process is successful and verifies that the enrollment status for the "Math101" course is updated to "Enrolled."

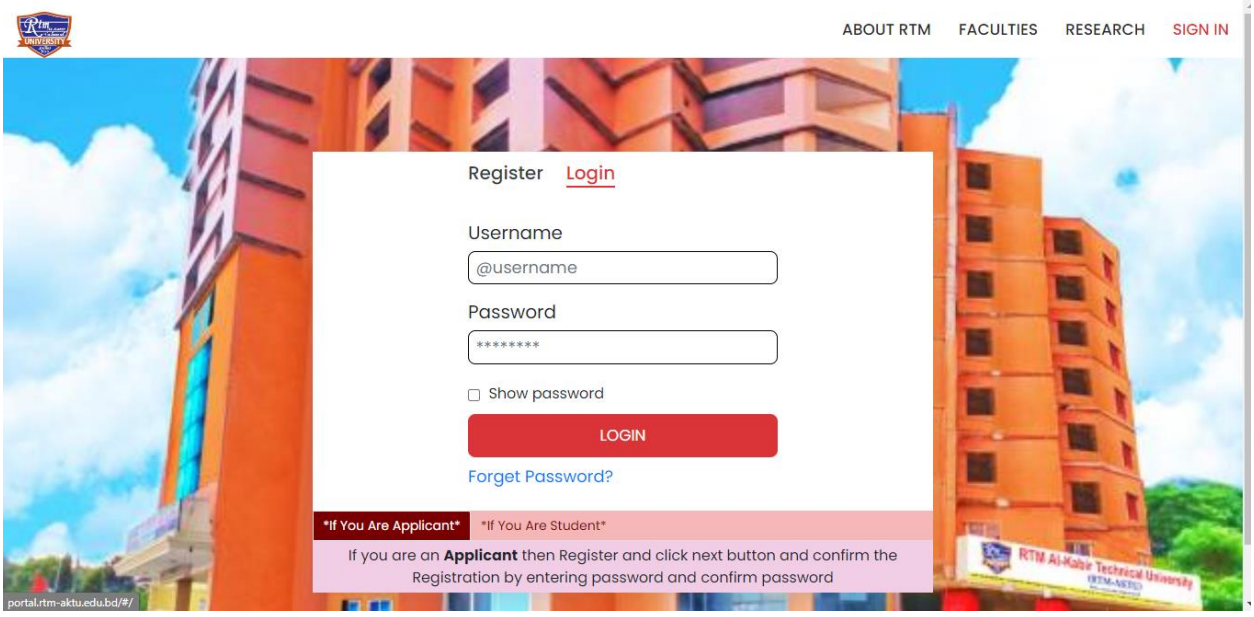


Figure 54: Acceptance Test

11.7 Security Testing

Security Test

Test Case

Name of the test case	**Security Testing**
Test Class	AuthenticationSecurityTest
Description of the test	This test case aims to verify the security of the authentication mechanism in the IUMS. It focuses on validating that only authorized users can access the system and perform specific actions.

Source of Data	**Steps in the Testing Process**	**Expected Outcome**	**Actual Outcome**
For this example, we will use test data specific to the security test, including valid and invalid user credentials.	Setup: Ensure that the IUMS is deployed and the authentication module is active with the necessary user accounts and access controls configured. Execution: Attempt to access the system using both valid and invalid user credentials. Perform actions that require authentication, such as accessing restricted pages or performing privileged operations Assertion: Compare the actual outcome with the	The expected outcome of the testAuthenticationSecurity() method is that authentication with valid credentials is successful, while authentication with invalid credentials fails. Additionally, privileged operations should only be successful for authorized users.	During the test, the user attempts to authenticate with both valid and invalid credentials. The test asserts that authentication with valid credentials is successful and authentication with invalid credentials fails. Additionally, the test verifies that privileged operations, such as changing the password, are only successful for authorized users.

	expected outcome based on the system's response and the access control mechanisms.		

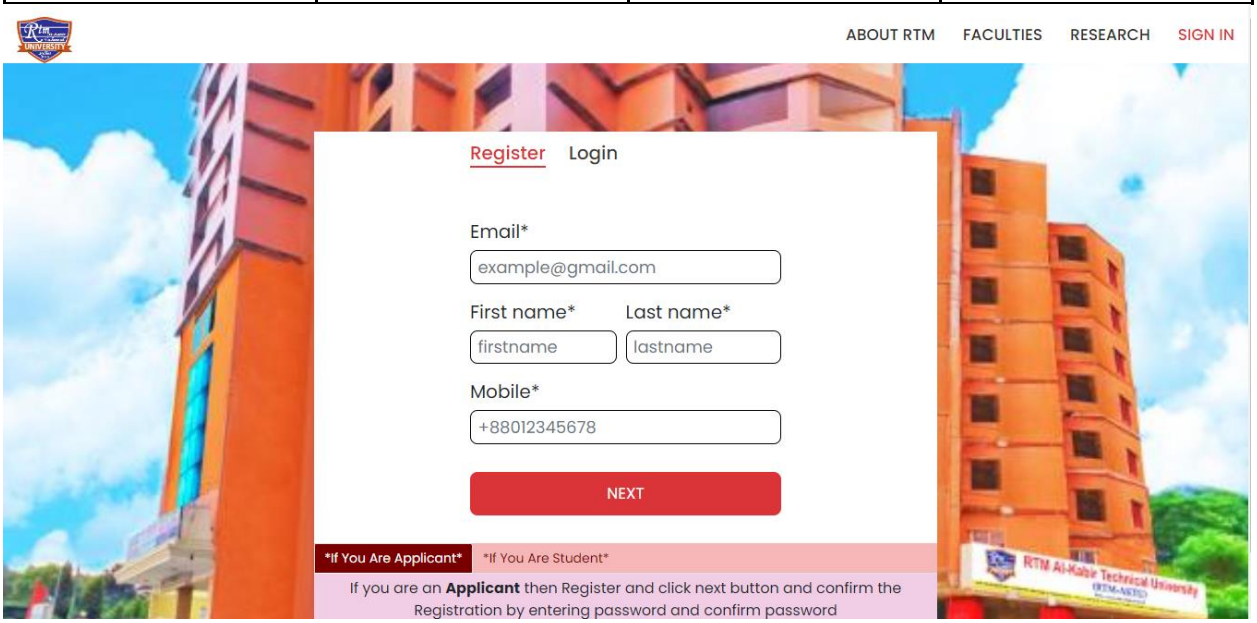


Figure 55: Security Testing

11.8 Accessibility Testing

Accessibility Test

Test Case

Name of the test case	**Accessibility Testing**
Test Class	AccessibilityTest
Description of the test	This test case aims to evaluate the accessibility of the IUMS user interface to ensure it is usable by individuals with disabilities. It focuses on verifying that the system conforms to accessibility standards and guidelines.

Source of Data	Steps in the Testing Process	Expected Outcome	Actual Outcome
For this example, we will use a combination of automated accessibility testing tools and manual evaluation based on accessibility guidelines.	Setup: Ensure that the IUMS is deployed and accessible for testing. Execution: Use automated accessibility testing tools, such as aXe or WAVE, to scan the user interface for potential accessibility issues. Additionally, perform manual evaluation based on recognized accessibility guidelines, such as the Web Content Accessibility Guidelines (WCAG).	The expected outcome of the testAccessibility() method is that the IUMS user interface is accessible and conforms to recognized accessibility standards and guidelines. The overall accessibility result should be positive, with no violations identified.	During the test, automated accessibility testing tools are used to scan the user interface for potential accessibility issues. Additionally, manual evaluation based on accessibility guidelines is performed. The test asserts that the overall accessibility result is positive and there are no violations identified.

Table 24: Accessibility Testing

Chapter 12 – Implementation

12.1 Training

Implementation training typically refers to the process of providing training and guidance to individuals or teams involved in implementing a system or software. It focuses on imparting the necessary knowledge and skills required to successfully deploy and configure the system or software in a specific environment.

SL No.	User	Training Scope	Time Period	Comment
01.	The user refers to the individuals or teams who will be responsible for implementing the system or software. They may include IT administrators, system integrators,	The training scope defines the specific areas and tasks that will be covered during the implementation training. It may include topics such as system	The time period for implementation training can vary depending on the complexity of the system or software and the specific needs of	Implementation training is crucial for ensuring a successful deployment of the system or software. It allows the user to gain the necessary skills and knowledge to effectively configure and operate the system in their specific environment. It is

	developers, or technical support staff.	installation, configuration, integration with other systems, data migration, security setup, and troubleshooting.	the participants. It may range from a few days for basic training to several weeks for in-depth and comprehensive training. The duration should be sufficient to cover all essential aspects of the implementation process and provide ample time for hands-on practice and knowledge assimilation.	important to design the training program with the user's needs in mind, providing a combination of theoretical concepts, practical exercises, and ongoing support. Regular assessment and feedback sessions can help identify areas for improvement and ensure that the training is meeting the user's requirements. Additionally, providing comprehensive documentation and resources as part of the training can serve as valuable references for users during and after the implementation process.

Table 25: User training

12.2 Implementation Scheme

Big Bang

The Big Bang approach is a software development and deployment strategy where all the components of a system are implemented and integrated simultaneously, followed by a full-scale deployment. In this approach, the entire system is developed, tested, and deployed as a single unit, rather than gradually implementing and integrating individual components.

12.3 Scaling

Scaling refers to the process of expanding or adjusting a system's capacity to handle increased workload, user traffic, or data volume. It involves making changes to various components of the system to ensure it can effectively accommodate growing demands without sacrificing performance or stability.

12.4 Load Balancing

Load balancing is a technique used in computer networks and distributed systems to distribute incoming workload or network traffic across multiple servers or resources. The goal of load balancing is to optimize resource utilization, improve system performance, maximize throughput, and ensure high availability and reliability.

Chapter 13 – Critical Appraisal and Evaluation

13.1 Objective could be Met

Efficient Administration: An IUMS can streamline administrative processes by automating tasks such as student enrollment, course registration, fee management, academic record management, and generating reports. This improves administrative efficiency, reduces manual effort, and enables faster decision-making.

Improved Communication: An IUMS facilitates effective communication between different stakeholders, including students, faculty, administrators, and parents/guardians. It provides channels for announcements, notifications, messaging, and discussion forums, enabling timely and transparent communication.

Enhanced Academic Management: With an IUMS, academic management becomes more streamlined. It allows for efficient course planning, scheduling, and curriculum management. Faculty members can use the system to manage class rosters, submit grades, and provide feedback to students. The system can also support the creation and management of academic resources such as e-learning materials and assessments.

Integrated Student Information: An IUMS acts as a centralized repository for student information, including personal details, academic records, attendance, and performance. This integration enables efficient access to student information, simplifies data management, and supports data-driven decision-making.

The specific objectives that can be met through an IUMS may vary depending on the institution's requirements and goals. Customization and adaptation of the system to meet the unique needs of the institution play a crucial role in achieving the desired objectives.

Objective-1

Achievement rate and others

Increased Achievement Rate: The primary objective is to improve the achievement rate of students, which can be measured by tracking metrics such as graduation rates, academic progression, and student success indicators. The IUMS can support this objective through various features:

Early Warning Systems: Implementing early warning systems within the IUMS to identify students who may be at risk of academic difficulties. This allows for timely interventions and support to help struggling students.

Academic Support Services: Integrating academic support services such as tutoring programs, study resources, and mentoring initiatives within the IUMS. This promotes student engagement and provides targeted assistance to improve academic performance.

Objective-2

Achievement rate and others

Enhanced Learning Analytics: Implementing learning analytics capabilities within the IUMS to gather data on student performance and engagement. This data can be analyzed to identify patterns, trends, and areas for improvement. Insights gained from learning analytics can inform teaching strategies, curriculum enhancements, and personalized learning interventions.

Objective-3

Achievement rate and others

Effective Student-Faculty Interaction: The IUMS can facilitate improved communication and interaction between students and faculty members, fostering an environment of collaboration and support. Features such as online discussion forums, virtual office hours, and messaging systems can enhance accessibility and promote meaningful engagement between students and faculty.

Objective-4

Achievement rate and others

Personalized Learning: The IUMS can enable personalized learning experiences by providing adaptive learning materials, personalized recommendations, and customized learning paths. This individualized approach can address students' unique needs, learning styles, and pace of learning, leading to improved academic performance.

13.2 Objective that totally don’t meet of touched

While the IUMS can provide various administrative and academic management features, it is not specifically designed or focused on addressing the cost or quality of cafeteria meals. The primary purpose of an IUMS is to streamline and automate administrative processes, enhance academic management, and improve communication within the university.

Therefore, reducing cafeteria meal costs and improving food quality would typically fall under the purview of the university's facilities management or food services department. These objectives would require specific initiatives such as negotiating contracts with food vendors, implementing cost-saving measures, sourcing high-quality ingredients, and ensuring compliance with food safety standards.

Although the IUMS may indirectly support some aspects related to cafeteria services, such as enabling online meal ordering or integrating payment systems, its core functionalities are not tailored towards directly addressing the cost or quality of cafeteria meals.

The reasons why it could not be touch

The reasons why certain objectives may not be directly addressed or touched upon in an Integrated University Management System (IUMS) can vary. Here are some possible reasons:

Scope and Focus: An IUMS is primarily designed to streamline administrative processes, enhance academic management, and improve communication within the university. It may have a specific set of functionalities and features that align with these core objectives. Objectives related to areas outside the system's scope and focus, such as cafeteria management, may not be included or prioritized.

Diverse Institutional Needs: Different universities have unique needs and priorities. While some institutions may place significant emphasis on cafeteria services and food quality, others may prioritize different areas such as research support, student recruitment, or alumni engagement. The IUMS may be customized to meet the specific needs of the institution, and cafeteria management may not be a primary focus for that particular institution.

Separate Systems and Departments: Some objectives, such as cafeteria management, fall under the purview of specific departments or systems within the university. Facilities management, food services, or campus dining may have their own dedicated systems, processes, and personnel to handle these aspects. Integrating cafeteria management directly into the IUMS may not be practical or feasible due to the existing infrastructure and workflows in place.

Specialized Systems: Objectives that require specialized systems or technologies may not be directly integrated into the IUMS. For example, optimizing cafeteria meal costs and quality may involve dedicated food service management software, inventory management systems, or supplier management solutions that are specifically designed for that purpose.

What could have been done?

Vendor Negotiations: The University can actively engage with food vendors or suppliers to negotiate better pricing and terms. This can include exploring bulk purchasing options, seeking competitive bids, or entering into long-term contracts that offer cost savings.

Menu Planning and Ingredient Sourcing: The University can work closely with food service providers to develop cost-effective and nutritious menus. Emphasizing seasonal and locally sourced ingredients can help reduce costs and improve the quality of food. Collaborating with culinary experts or nutritionists can ensure that the menus meet both financial and nutritional goals.

Efficient Inventory Management: Implementing robust inventory management systems can help minimize waste, control costs, and optimize ingredient usage. By tracking inventory levels, monitoring expiration dates, and adjusting procurement accordingly, the university can effectively manage food inventory and reduce unnecessary expenses.

Quality Assurance Programs: Establishing quality assurance programs and conducting regular inspections can help ensure that food service providers adhere to food safety standards, maintain cleanliness, and deliver high-quality meals. Implementing feedback mechanisms and conducting surveys among students and staff can help monitor satisfaction levels and identify areas for improvement.

Chapter 14 Lessons Learned

14.1 Pre-Project-Review-closing

The Pre-Project Review (PPR) closing in an Integrated University Management System (IUMS) marks the completion of the pre-planning phase and sets the stage for the project's implementation. It involves a comprehensive evaluation of the project's readiness and alignment with the institution's goals. Here's an overview of the Pre-Project Review closing process in an IUMS:

Documentation Review: The project team, stakeholders, and relevant experts review all the documentation related to the IUMS project. This includes the project charter, requirements documents, scope statements, and any other project-related artifacts. The purpose is to ensure that the project documentation accurately represents the project's objectives, scope, and deliverables.

Assessment of Requirements: The project team verifies that the requirements captured during the pre-planning phase are complete, well-defined, and aligned with the institution's needs. Any gaps or discrepancies are identified and addressed to ensure that the IUMS will meet the desired outcomes.

Resource Evaluation: The availability and allocation of resources necessary for the successful implementation of the IUMS are reviewed. This includes assessing the availability of skilled personnel, hardware, software, infrastructure, and budget. Any resource constraints or dependencies are identified and appropriate plans are put in place to address them.

14.2 What I have learnt

During the process of implementing an Integrated University Management System (IUMS), you may have learned several valuable lessons and gained insights. Here are some common learnings that individuals and organizations often experience:

Understanding of Institutional Processes: Implementing an IUMS involves a deep understanding of the existing institutional processes, workflows, and requirements. Through this process, you may have gained insights into how various departments and stakeholders interact, the challenges they face, and opportunities for streamlining and improving efficiency.

System Integration Challenges: Integrating different systems and databases within the IUMS can present technical challenges. You may have learned about the complexities involved in data migration, data synchronization, and ensuring seamless communication between different modules or components of the system.

Change management: Putting an IUMS into place frequently calls for major adjustments to the institution's procedures, division of labor, and job descriptions. To ensure the seamless adoption and acceptance of the new system, you may have heard about the significance of change management tactics including effective communication, training programs, and stakeholder involvement.

14.3 The problems I have faced

While implementing an Integrated University Management System (IUMS), you may have encountered several challenges or problems. Here are some common issues that organizations often face during the IUMS implementation process:

Complexity of System Integration: Integrating different systems, databases, and modules within the IUMS can be complex and technically challenging. Data migration, data synchronization, and ensuring seamless communication between different components of the system may have posed difficulties and required significant technical expertise.

Resistance to Change: Implementing an IUMS often involves significant changes in processes, roles, and responsibilities. Resistance to change from stakeholders, such as faculty, staff, and administrators, may have posed challenges and required effective change management strategies to overcome resistance and ensure successful adoption of the new system.

Limited IT Infrastructure: Insufficient or outdated IT infrastructure can hinder the implementation of an IUMS. Challenges such as outdated hardware, inadequate network infrastructure, or insufficient server capacity may have impacted system performance and required additional investments or upgrades.

14.4 What Solution Occurred

In an Integrated University Management System (IUMS), various solutions can be implemented to address the challenges and problems that arise during the implementation process. Here are some potential solutions that organizations may consider:

Engage Stakeholders: Actively involve stakeholders, including faculty, staff, and administrators, throughout the implementation process. Encourage their participation, address their concerns, and communicate the benefits of the IUMS to gain their support and cooperation.

Change Management Strategies: Develop and execute effective change management strategies to manage resistance to change. Provide comprehensive training programs, clear communication, and continuous support to ensure smooth transition and adoption of the new system.

Infrastructure Upgrades: Assess the existing IT infrastructure and identify areas that need improvement or upgrading. Invest in hardware, network infrastructure, and server capacity as necessary to ensure optimal system performance and reliability.

Chapter 15 – Conclusion

15.1 Summary of the Project

The Integrated University Management System (IUMS) is a comprehensive software solution designed to streamline and automate various administrative and academic processes within a university or educational institution. It aims to improve efficiency, enhance communication, and provide a centralized platform for managing student information, course registrations, academic records, financial transactions, and other related activities. The IUMS project involves the implementation of a system that integrates different modules, such as student management, faculty management, course management, finance management, and more. It encompasses the entire student lifecycle, from admission and enrollment to graduation, and facilitates smooth operation and collaboration among various departments and stakeholders.

15.2 Goal of the project

The goal of the Integrated University Management System (IUMS) project is to create a centralized and integrated software solution that streamlines and enhances various administrative and academic processes within a university or educational institution. The primary objectives of the project include:

Efficiency and Productivity: The IUMS aims to improve the efficiency and productivity of university operations by automating manual tasks, reducing paperwork, and streamlining administrative processes. This goal is achieved by providing a centralized platform that enables easy access to information, facilitates data sharing, and eliminates redundant processes.

Data Accuracy and Accessibility: The project aims to ensure the accuracy and accessibility of data within the university by centralizing information in a unified system. This helps eliminate data silos, minimize errors, and enable authorized users to access up-to-date and consistent information across different departments and functions.

Improved Decision-Making: The IUMS provides comprehensive reporting and analytics capabilities, enabling university administrators and decision-makers to access real-time data and insights. This empowers them to make informed decisions related to student enrollment, course offerings, resource allocation, budget planning, and strategic initiatives.

15.3 Success of the Project

The success of the Integrated University Management System (IUMS) project can be evaluated based on several key factors:

Achievement of Project Goals: The project's success can be determined by assessing whether it has achieved its intended goals and objectives. This includes improvements in operational efficiency, data accuracy and accessibility, decision-making capabilities, student experience, faculty and staff empowerment, institutional collaboration, and adaptability.

User Satisfaction: User satisfaction is a crucial measure of project success. Feedback from students, faculty, staff, and administrators regarding the usability, functionality, and overall experience of using the IUMS can help gauge its success. Regular surveys, user feedback sessions, and support requests can provide insights into user satisfaction levels.

Adoption and Utilization: The successful implementation of the IUMS relies on its adoption and utilization by the university community. High rates of system usage, active participation in system activities, and widespread acceptance and integration of the IUMS into daily workflows indicate the project's success.

15.4 What I have done in the documentation

Requirements Documentation: Documenting the functional and non-functional requirements of the IUMS based on stakeholder input and analysis. This includes capturing the desired features, system behavior, user interactions, and performance expectations.

Design Documentation: Creating design documents that outline the architectural structure, data models, user interfaces, and integration points of the IUMS. These documents help communicate the technical aspects of the system and guide the development process.

User Manuals and Guides: Developing user manuals and guides that provide instructions and information on how to use the IUMS effectively. These documents may include step-by-step instructions, screenshots, and explanations to help users navigate through different functionalities and perform tasks.

System Documentation: Documenting the technical aspects of the IUMS, such as system configurations, installation procedures, database schemas, API documentation, and system dependencies. This information helps IT teams understand and maintain the system.

The specific documentation activities and deliverables can vary depending on the project's scope, requirements, and methodologies followed. It's important to maintain organized and up-to-date documentation to ensure effective communication, knowledge transfer, and ongoing support for the IUMS.

15.5 Value of the Project

The Integrated University Management System (IUMS) project brings several significant values to a university or educational institution. These values include:

Improved Efficiency: The IUMS automates and streamlines various administrative and academic processes, reducing manual efforts, paperwork, and time-consuming tasks. This leads to increased operational efficiency, allowing staff and faculty to focus more on core responsibilities and strategic initiatives.

Enhanced Data Management: With a centralized system like IUMS, data accuracy, integrity, and accessibility are improved. It eliminates data duplication, reduces errors, and provides a single source of truth for student information, course data, financial records, and other relevant data. This ensures that accurate and up-to-date information is available to authorized users whenever needed.

Enhanced Decision-Making: The IUMS provides comprehensive reporting and analytics capabilities, enabling university administrators and decision-makers to access real-time data and insights. This empowers them to make informed decisions related to enrollment management, resource allocation, curriculum planning, and student support initiatives.

Improved Student Experience: The IUMS enhances the student experience by providing self-service functionalities, easy access to academic information, online registration and payment options, and streamlined communication channels. Students can conveniently manage their course registrations, access grades and academic records, communicate with faculty, and stay informed about important announcements.

Overall, the value of the IUMS project lies in its ability to streamline administrative processes, enhance data management, improve decision-making, and provide a better experience for students, faculty, and staff. It contributes to the overall efficiency, effectiveness, and competitiveness of the educational institution, ultimately leading to improved student outcomes and institutional success.

15.6 My Experience

I had to do a lot of research to put this system in place. I've worked on a number of online and offline research initiatives. The most difficult task was bug removal. Finally, after overcoming all obstacles, I was able to put my strategy into action. I have learned a lot through this venture. Deal of experience. I confronted and conquered a variety of hurdles, acquiring significant experience. I learnt how to manage a large project and accomplish all of its objectives in a short period of time, which gave me tremendous experience.

Appendices:

Test Script

User Guide:

1. User Enrollment Process
Brief Description: This appendix outlines the process for a student to enroll in courses using the IUMS.
Actors: Student: The user who wants to enroll in courses.
Preconditions: ● The student is logged into their IUMS account. ● The course registration period is open. ● The student meets any prerequisites or restrictions for the courses they intend to enroll in.
Basic Flow of Events: ● The student navigates to the "Course Enrollment" section in the IUMS. ● The system displays a list of available courses, along with relevant information such as course name, instructor, schedule, and available seats. ● The student searches or filters the course list to find the desired courses. ● The student selects the desired courses by clicking on the corresponding checkboxes. ● If there are any course conflicts or restrictions, the system notifies the student and provides alternative options.
Post-Conditions: ● The student is successfully enrolled in the selected courses. ● The student's course schedule is updated in the IUMS.

2. Faculty Grade Submission Process
Brief Description: This appendix provides an overview of the process for faculty members to submit grades for student assignments and assessments using the IUMS.
Actors: Faculty: The user who is responsible for assigning and submitting grades for student assessments.
Preconditions:

- The faculty member is logged into their IUMS account.
- The grading period for the course is open.
- The faculty member has completed grading the student assignments.

Basic Flow of Events:

- The faculty member navigates to the "Grade Submission" section in the IUMS.
- The system displays a list of courses assigned to the faculty member.
- The faculty member selects the course for which grades need to be submitted.
- The system presents a list of enrolled students in the selected course.
- The faculty member selects a student to input their grade.

Post-Conditions:

- The grades for the selected student's assignments and assessments are successfully submitted and recorded in the IUMS.
- The faculty member can view the submitted grades in their IUMS account.

3. Financial Aid Application Process

Brief Description: This appendix outlines the process for a student to apply for financial aid using the IUMS.

Actors: Student: The user who wants to apply for financial aid..

Preconditions:

- The student is logged into their IUMS account.
- The financial aid application period is open.
- The student has gathered the necessary documents and information required for the application.

Basic Flow of Events:

- The student navigates to the "Financial Aid" section in the IUMS.
- The system provides information and instructions regarding the financial aid application process.
- The student reviews the eligibility criteria and requirements for financial aid.
- The student completes the online financial aid application form provided by the system.
- The student uploads any required supporting documents, such as income statements, tax returns, or other requested information.

Post-Conditions:

- The student's financial aid application is successfully submitted and recorded in the IUMS.

- The financial aid office receives the application for review and evaluation.
- The student can track the status of their financial aid application in their IUMS account.

Works Cited

(DSDM), D. S. (18, 06 2018). *Dynamic Systems Development Method (DSDM)*. From GeeksforGeeks: https://www.geeksforgeeks.org/dynamic-systems-development-method-dsdm/

2, C. (n.d.). *Chapter 2*. From Developing Client/Server Applications: https://www.microfocus.com/documentation/net-express/nx31books/sgdevt.htm

chapter, P. M.-N. (2022, 12 25). *PMI Mid-Nebraska chapter*. From Welcome | PMI Mid Nebraska: https://pmimidnebraska.org/

history, N. U. (2023, 02 10). *Northern University > about nub and our history*. From Northern University > about nub and our history: https://www.nub.ac.bd/about/dj7mbril/about-nub-and-our-history

methodologies)?, W. i. (2022, 09 15). *What is Agile Software Development (agile methodologies)?* From Software Quality: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development

Project Management Methodology: Definition, t. e. (2023, 02 12). *Project Management Methodology: Definition, types, examples*. From Your Guide to Project Management Best Practices: https://mymanagementguide.com/basics/project-methodology-definition/

Project Management Methodology: Definition, t. e. (2023, 05 20). *Project Management Methodology: Definition, types, examples*. From Your Guide to Project Management Best Practices: https://mymanagementguide.com/basics/project-methodology-definition/

Rapid application development (RAD): Definition, S. &. (2023, 05 04). *Rapid application development (RAD): Definition, Steps & Full Guide*. From Kissflow: https://kissflow.com/application-development/rad/rapid-application-development/

University, S. (2023, 04 25). *Southeast University*. From UMS: https://ums.seu.edu.bd/

192-16-447_Tajul Islam Refath

ORIGINALITY REPORT

27%	23%	3%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	17%
2	Submitted to Daffodil International University Student Paper	2%
3	Submitted to University of Greenwich Student Paper	1%
4	Submitted to Tower Hamlets College Student Paper	1%
5	Panagiotis Leloudas. "Introduction to Software Testing", Springer Science and Business Media LLC, 2023 Publication	1%
6	Submitted to University of Derby Student Paper	<1%
7	Submitted to West Herts College Student Paper	<1%
8	Submitted to Victoria University Student Paper	<1%
	www.ijirem.org	

9	Internet Source	<1 %
10	Submitted to American Public University System Student Paper	<1 %
11	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1 %
12	Submitted to Westcliff University Student Paper	<1 %
13	fargea.blogspot.com Internet Source	<1 %
14	Submitted to University of Bedfordshire Student Paper	<1 %
15	Submitted to University of Kent at Canterbury Student Paper	<1 %
16	expediensolutions.com Internet Source	<1 %
17	Submitted to Luton Sixth Form College, Bedfordshire Student Paper	<1 %
18	Submitted to NCC Education Student Paper	<1 %
19	Submitted to University of Wales Institute, Cardiff	<1 %