



Daffodil
International
University

TITLE OF THE PROJECT

IUMS ADMISSION SERVICE

Submitted by

Habiba Islam

ID: 201-16-490

Supervised by

Md. Mehedi Hassan

Lecturer (DIU)

Course: **Internship – Fall’23**

Course Code: **CIS499**

Department of Computing and Information System (CIS)

Submission Date: **23 January 2024**

DAFFODIL INTERNATIONAL UNIVERSITY

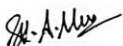
APPROVAL

This internship project titled "IUMS Admission Service" Submitted by **Habiba Islam, ID No: 201-16-490**, 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 was held on 23/01/2024.

BOARD OF EXAMINERS


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

Chairman


Dr. Shekh Abdullah-Al-Musa Ahmed
Assistant Professor
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner


Md. Nasimul Kader
Assistant Professor
Department of Computing & Information Systems
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner


Prof. Mohammad Shorif Uddin,
Professor
Department of Computer Science and Engineering
Jahangirnagar University

External Examiner

© Daffodil International University

DECLARATION

I, **Habiba Islam**, hereby declare that the work in this dissertation titled “**IUMS Admission Service**”, I have done this internship project under the 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 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


23.07.2024

Md. Mehedi Hassan

Lecturer

Department of Computing & Information System
Daffodil International University, Bangladesh

SUBMITTED BY

Habiba Islam

Habiba Islam

ID: 201-16-490

Dept. of Computing & Information System
Daffodil International University, Bangladesh

Acknowledgment

I would want to begin by deeply appreciating Almighty Allah for granting me the opportunity to go on this journey. The successful execution of this undertaking and all of its associated activities would not have been feasible without supernatural guidance and support. I would like to thank Daffodil Computers Ltd. (DCL) for providing me with an internship trainee position. I have gained invaluable experiences from this internship that have significantly benefited my professional and personal development.

Teachers play the role of a second set of parents, offering advice and insight to help students advance. During my internship program, I had the good fortune to work with two amazing teachers who really shaped my career. My deepest appreciation goes out to Md. Mazharul Islam, my intern trainer, and Md. Mehedi Hassan, my academic supervisor. Their resolute assistance, astute guidance, and motivation strengthened me during difficult assignments, and I am incredibly appreciative of their guidance. Not only do I appreciate their knowledge, but also their generosity, friendliness, and unwavering faith in my talents. I want to express my deep gratitude to them for all of their assistance, for teaching me how to deal with difficult situations, and for giving me the knowledge I need to take on significant projects in the future.

I am grateful to **Almighty Allah, DCL, Md. Mazharul Islam and Md. Mehedi Hassan** for their crucial contributions to my personal development.

Abstract

As a cutting-edge and all-inclusive web-based platform, IUMS ADMISSION SERVICE is ready to completely transform the way that educational institutions handle admissions. Numerous features are integrated into this advanced technology to optimize the admissions process. With the help of the Admission Circular Management module, educational institutions may produce and distribute comprehensive circulars that educate potential applicants of approaching deadlines and important dates. By giving institutions the ability to specify and control the prerequisites for admission to various courses, the Admission Requirement Tracking component increases transparency. The Admission Subject Configuration and Admission Center Setup modules highlight the system's customization capabilities, which enable educational institutions to customize the platform to meet their unique academic requirements and operational requirements. With the help of the seamless integration of the Fee Management system, administrators may effectively track, invoice, and reconcile entrance fees. Administrators may monitor applications in real-time while also improving the applicant experience with the user-friendly Application Process and strong Application Tracking features. IUMS ADMISSION SERVICE promises to redefine the efficacy and efficiency of admission management for educational institutions, ultimately fostering a more approachable and structured approach to student enrollment. It does this by transforming the admissions process into a streamlined, transparent, and user-centric experience.

Table of contents

Acknowledgment	iii
Abstract	iv
Chapter 1 - Introduction	1
1.1 Introduction	1
Chapter 2 – Initial Study	2
2.1 Project Proposal	2
2.2 Background of the project	4
2.3 Problem Areas	5
2.4 Possible Solutions.....	5
Chapter 3 – Literature Review	6
3.1 Discussion on the Problem Domain (IUMS).....	6
3.2 Discussion on the Problem Solution:.....	6
3.3 Comparison among the leading solutions.....	8
3.4 Recommended Approach	12
Chapter 4 – Methodology	13
4.1 What to Use	13
4.2 Why to Use Methodology.....	18
4.3 (IUMS) Sections of Methodology	19
4.4 Implementation Plans	20
Chapter 5 – Planning	22
5.1 Project Plan.....	22
5.1.1 Work Breakdown Structure.....	22
5.1.2 IUMS ADMISSION SERVICE Resource Allocation	23
5.1.3 Time Boxing.....	24
5.1.4 Gantt Chart of University Management System Working	25
5.2 Test Plan	25
5.2.1 IUMS ADMISSION SERVICE Against the Time Boxes	25
5.2.2 Required Test	26
5.2.3 Test Case of IUMS	28
5.2.4 User Acceptance Test Plan.....	28

5.3 Risk Management of University Management System	29
5.3.1 Risk Identification	29
5.3.2 Risk Assessment of IUMS	31
5.3.3 Risk Precaution / Action Plan	32
5.3.4 Steps Taken for Possible Risks of UMS	33
5.4 Change Management	35
5.4.1 Factors that Might Cause Change	35
5.4.2 DSDM Welcome Change of IUMS	35
5.4.3 Considering Business Priority	36
5.4.4 Change Workshop	36
5.4.5 Changes That Are Allowed	37
5.4.6 Key Decision Taker of Change	37
5.5 Quality Management	37
5.5.1 Rules Applied to Maintain Quality	38
5.5.2 DSDM Standard Quality Measures	38
5.5.3 Quality Plan and Measuring Meter	39
Chapter 6 – Feasibility	40
6.1 All Possible Types of Feasibility IUMS	40
6.2 Cost Benefit Analysis for IUMS ADMISSION SERVICE	44
6.3 Explain DSDM Good or Bad for this Project	46
Chapter 7 – Foundation	47
7.1 The Problem Area Identification	47
7.1.1 Interview	47
7.1.2 Questionnaires	47
7.2 Overall Requirement List	48
7.2.1 Functional Requirements	48
7.2.2 Non-Functional Requirements	49
7.3 Technology to Be Implemented	49
Chapter 8 – Exploration	53
8.2 Full System Use Case	54

8.2.1	Breakdown of Use Case diagram according to the functions of IUMS ADMISSION SERVICE.....	55
8.3	Full System Activity Diagram.....	59
8.4	Requirements Catalogue.....	60
8.6	Prototype of the new system.....	66
Chapter 9 – Engineering		71
9.1	Modules of the new system	71
9.2	Use Case for IUMS ADMISSION SERVICE.....	74
9.4	Component diagram for IUMS ADMISSION SERVICE	76
9.5	Deployment diagram for IUMS ADMISSION SERVICE.....	77
9.6	System Interface Design / Prototype	78
Chapter 10 – Deployment/Development		81
10.1	Core Module Coding Sample.....	81
10.2	Possible problem breakdown	94
10.3	Prioritization while developing.....	95
Chapter 11 – Testing		97
11.1	Test Plan Acceptance.....	97
11.2	Test Case.....	98
Chapter 12 – Implementation		101
12.1	Training.....	101
12.2	Big Bang	102
12.3	Scaling.....	103
12.4	Load balancing.....	103
Chapter 13 – Critical Appraisal and Evaluation		104
13.1	Objective Could be Met	104
13.2	How much better could have been done	105
13.3	How better is the features of the solution	105
13.4	Which features could not be touched.....	106
Chapter 14 – Lesson Learned		107
14.1	Pre Project – Review – Closing	107
14.2	What I have learned	107

14.3	The Problems I Have Faced.....	108
14.4	What Solution Occurred	108
Chapter 15 – Conclusion		110
15.1	Summary of the project.....	110
15.2	Goal of the project	111
15.3	Success of the project.....	112
15.4	What I have done in the documentation	113
15.5	Value of the project.....	113
15.6	My Experience	114
Appendix.....		116
	User guide.....	116
Works Cited.....		128
Plagiarism Report		130

List of Figures of IUMS ADMISSION SERVICE

Figure 1: Overview of SEU Admission	8
Figure 2: Overview of BOU Admission	10
Figure 3: Overview of BUBT Admission	11
Figure 4: DSDM Methodology	14
Figure 5: Waterfall Methodology	16
Figure 6: RAD Methodology	17
Figure 7: Gantt Chart	25
Figure 8: Client server application.....	50
Figure 9: Web based application.....	52
Figure 10: Activity diagram.....	53
Figure 11: Full system use case	54
Figure 12: Use case for full system.....	55
Figure 13: Use case for admission info.....	56
Figure 14: Use case for admission center	57
Figure 15: Use case for fees management	58
Figure 16: Full system activity diagram	59
Figure 17: Admission circular list view.....	66
Figure 18: Admission circular create	66
Figure 19: Admission subject list view.....	67
Figure 20: Admission subject create	67
Figure 21: Admission requirement list view.....	68
Figure 22: Admission requirement create	68
Figure 23: Admit card signatory create	69
Figure 24: Admit card signatory list view	69
Figure 25: Quota list view.....	70
Figure 26: Quota create.....	70
Figure 27: Use Case for IUMS ADMISSION SERVICE	74
Figure 28: Sequence diagram for IUMS ADMISSION SERVICE	75
Figure 29: Component diagram for IUMS ADMISSION SERVICE.....	76

Figure 30: Deployment diagram for IUMS ADMISSION SERVICE.....	77
Figure 32: Admission center list view	78
Figure 31: Admission center create view.....	78
Figure 34: Admission building list view.....	79
Figure 33: Admission building create.....	79
Figure 35: Admission center room list view.....	80
Figure 36: Admission center room create	80
Figure 37: Admission center code sample - 1.....	81
Figure 38: Admission center code sample - 2.....	82
Figure 39: Admission center code sample – 3.....	83
Figure 40: Admission circular create code sample - 1.....	84
Figure 41: Admission circular create code sample - 2.....	85
Figure 42: Admission circular create code sample - 3.....	86
Figure 43: Admission circular create code sample - 4.....	87
Figure 44: Admission circular create code sample – 5.....	88
Figure 45: Admission circular list code sample – 1.....	89
Figure 46: Admission circular list code sample – 2.....	90
Figure 47: Admission circular list code sample – 3.....	91
Figure 48: Admission circular list code sample – 4.....	92
Figure 49: Admission circular list code sample – 5.....	93

List of Tables of IUMS ADMISSION SERVICE

Table 1: IUMS ADMISSION SERVICE Work Breakdown Structure	22
Table 2: IUMS ADMISSION SERVICE Resource Allocation.....	23
Table 3: Time Boxing of IUMS ADMISSION SERVICE	24
Table 4: IUMS Sample of Testing Against the Timebox	26
Table 5: Test case of IUMS	28
Table 6: User acceptance sample of IUMS.....	29
Table 7: Risk identification of IUMS	31
Table 8: Risk assessment of IUMS	31
Table 9: Risk precaution of IUMS.....	33
Table 10: Steps Taken for Possible Risks of UMS.....	34
Table 11: Business priorities.....	36
Table 12: Web based application cost	42
Table 13: Desktop application cost.....	42
Table 14: Total cost of IUMS	44
Table 15: Total earning of IUMS.....	45
Table 16: Total revenue of IUMS	45
Table 17: Requirement catalog for login system	60
Table 18: Requirement catalog for the Admission Info.....	61
Table 19: Requirement catalog for the Admission Center.....	62
Table 20: Requirement catalog for the Fees Management	63
Table 21: Must have priorities	64
Table 22: Should have priorities	65
Table 23: Could have priorities.....	65
Table 24: Module for the login system	71
Table 25: Module for the Admission Configuration.....	72
Table 26: Module for the Admission Center Setup	72
Table 27: Module for the Fees Management Setup	73
Table 28: Prioritization of IUMS	96
Table 29: Unit test – test case	99
Table 30: Module test – test case	99

Table 31: Integration test – test case.....	100
Table 32: Training of Admission Admin.....	102

Chapter 1 - Introduction

1.1 Introduction

The IUMS ADMISSION SERVICE Module is a pivotal component within the broader framework of the comprehensive University Management System. This module has been meticulously designed to address the intricate processes associated with the admission lifecycle within a university setting. As educational institutions navigate the complexities of modern academia, the demand for streamlined, efficient, and technology-driven solutions becomes increasingly imperative. The IUMS ADMISSION SERVICE emerges as a powerful tool to meet these demands, aiming to revolutionize and optimize the admission procedures within the university. (ValuePLUS Computer Systems Ltd., n.d.)

This cutting-edge technology incorporates a ton of features to streamline the admissions procedure. The Admission Circular Management module enables educational institutions to create and disseminate thorough circulars that inform prospective applicants of upcoming deadlines and significant dates. The Admission Requirement Tracking component improves transparency by enabling institutions to define and manage the requirements for admission to different courses.

The system's customization features, which let educational institutions tailor the platform to match their particular operational and academic needs, are highlighted in the Admission Subject Configuration and Admission Center Setup modules. Entrance fee management is made easy for administrators with the aid of the Fee Management system's seamless connection. With the user-friendly Application Process and robust Application Tracking tools, administrators may enhance the applicant experience while simultaneously keeping an eye on applications in real-time.

Chapter 2 – Initial Study

2.1 Project Proposal

The goal of the Admission Module-focused Integrated University Management System is to improve and expedite the intricate process of student admissions in academic institutions. This all-inclusive solution is made to ensure efficiency, correctness, and transparency by automating and optimizing several parts of the admission workflow. The proposed system will feature user-friendly interfaces for both administrators and applicants, providing a seamless experience from application submission to enrollment. Key functionalities include online application submission, document verification, merit-based ranking, communication channels for applicants and administrators, and real-time status tracking. Additionally, the system will integrate with existing university databases and management systems to ensure data consistency and eliminate redundancy. By implementing the University Management System with a dedicated Admission Module, educational institutions can expect to reduce administrative burdens, minimize errors, and ultimately enhance the overall experience for both applicants and university staff. This project aligns with the ongoing digital transformation in education, contributing to the modernization and efficiency of university administration.

Background Study

Higher education has evolved significantly with the advent of technology, necessitating a modernized approach to university administration. Traditional methods of handling student admissions are often time-consuming, prone to errors, and lack the efficiency required to meet the growing demands of today's educational institutions. Recognizing these challenges, the proposed "University Management System" with a specialized focus on the "Admission Module" aims to address these issues and bring about a transformative change in the admission process. The shift towards digital solutions is paramount in enhancing the accessibility, speed, and accuracy of admissions. By conducting a comprehensive background study, it becomes evident that the integration of a dedicated Admission Module within the University Management System is not only a practical necessity but also aligns with the global trend of leveraging technology to

streamline administrative processes in the education sector. This project seeks to contribute to the ongoing efforts to modernize university administration, making it more efficient and responsive to the needs of both academic institutions and prospective students.

Description of the proposed system

The "IUMS ADMISSION SERVICE" is a comprehensive and integrated solution that aims to transform the admissions process for students in educational institutions. This comprehensive system will encompass a range of features aimed at automating and optimizing every stage of the admission lifecycle. Prospective students will experience a user-friendly online application platform, allowing for the seamless submission of admission documents. The system will incorporate robust document verification mechanisms, ensuring the authenticity and accuracy of submitted information. The Admission Module will implement merit-based ranking algorithms to facilitate fair and efficient selection processes. Real-time communication channels will be integrated to keep applicants informed about their application status, and administrators will have access to intuitive dashboards for effective management and decision-making. Furthermore, the system will seamlessly integrate with existing university databases and management systems, promoting data consistency and eliminating redundancy.

Exploration & Engineering

The Prioritized Requirements List (PRL) serves as the foundation for the final criterion:

- Design a clean and intuitive user interface for both applicants and administrative staff.
- Develop an online application form that captures essential information.
- Implement a validation mechanism to ensure accurate and complete information.
- Integrate a document verification process to validate the authenticity of submitted documents.
- Establish connections with educational institutions for seamless verification of academic records.

- Integrate APIs or establish partnerships with standardized testing services for test score validation.

Iterative development – Timeboxing

The tasks must be divided into smaller parts with a time estimate for each in an iterative development process. The feasibility and foundation phases of the project will take 15-20% of the overall time, real development would take 50–60%, and documentation will take 10-15%.

In three months, the task must be finished, requiring 280–320 hours of labor. The project is divided into seven time periods, each lasting 20–40 hours. I'm going to use the "T-Shirt" estimating method to estimate the time-boxes. The story points will be computed using the XS, S, M, L, and XL t-shirt sizes. Compared to the smaller ones, the greater size will require more labour and time.

2.2 Background of the project

The IUMS ADMISSION SERVICE is a pivotal component within the broader framework of an integrated academic management platform. This project stems from the recognition of the evolving needs of modern educational institutions, necessitating a sophisticated and streamlined approach to the admission process. With the ever-increasing number of prospective students and the dynamic nature of admission criteria, this module aims to provide a comprehensive solution to universities seeking efficiency, transparency, and enhanced user experience in their admission procedures. By combining cutting-edge technology with a user-centric design, the system addresses the complexities of application management, document verification, and communication, ultimately contributing to the overall effectiveness of university administration. The project aligns with the broader goal of fostering a technologically advanced and accessible educational environment, where the admission process becomes a seamless and empowering experience for both applicants and administrators alike.

2.3 Problem Areas

The IUMS ADMISSION SERVICE seeks to address several critical problem areas inherent in traditional university admission processes. One of the primary challenges is the manual and time-consuming nature of admissions, leading to inefficiencies, delays, and increased workload for administrative staff. Paper-based documentation and verification processes contribute to errors and can hinder the overall transparency of the admission system. Furthermore, variations in the assessment and handling of applications are frequently the consequence of the absence of a centralized and uniform method. The existing systems may struggle to handle the growing volume of applications efficiently, leading to potential bottlenecks during peak admission periods. Furthermore, outdated communication methods can result in delayed notifications and a lack of real-time updates for applicants. The Admission Module aims to mitigate these challenges by introducing automation, digitalization, and enhanced communication channels to streamline the admission process, reduce errors, and provide a more user-friendly experience for both applicants and university administrators. (Hassan, 2021)

2.4 Possible Solutions

The IUMS ADMISSION SERVICE suggests creative fixes for the noted issues with university admissions. To streamline and expedite the application process, the module incorporates a user-friendly online application system that enables applicants to submit their materials digitally. Integration with educational institutions and standardized testing services ensures swift and accurate verification of academic records and test scores, reducing manual effort and potential errors. The implementation of a secure online payment gateway facilitates seamless processing of application fees, enhancing financial transactions and reducing administrative burdens.

Technical feasibility requires a thorough assessment of the institution's technical resources and competencies. Investing in resource development or seeking external expertise can help ensure the virtual platform's success. The platform's efficiency depends on whether it is user-friendly, secure, and capable of providing mentor ship, networking, and communication features.

Chapter 3 – Literature Review

The literature assessment for the IUMS ADMISSION SERVICE project comprises a review of pertinent studies and current research that aid in understanding and developing initiatives that bridge students' and alumni's educational trajectories. Several academic papers and publications stress the value of fostering close bonds between current students and well-known alumni. Studies demonstrate the beneficial effects of these programs on students' academic and professional paths. (Fink, 2014)

3.1 Discussion on the Problem Domain (IUMS)

The opportunities and problems of connecting students' and alumni's educational paths are at the center of the University Management System project's problem domain. This conversation explores important facets of the issue domain and the reasoning behind the project's goals.

- **Manual Admission Processes:** Traditional university admission processes often involve manual data entry and documentation, leading to inefficiencies, errors, and delays.
- **Complex Documentation and Verification:** Managing and verifying a diverse range of documents, including transcripts, recommendation letters, and standardized test scores, can be complex and time-consuming.
- **Scalability Challenges:** Universities, especially during peak admission seasons, may struggle with the scalability of their systems to handle a large volume of applications.
- **Data Security Concerns:** Security is a paramount concern, particularly when dealing with sensitive applicant information and academic records. (Musotsi, n.d.)

3.2 Discussion on the Problem Solution:

The proposed solutions for the IUMS ADMISSION SERVICE offer a comprehensive approach to address the identified challenges within the problem domain. By transitioning from manual processes to a streamlined online application system, the module aims to significantly reduce the

time and effort involved in handling admissions. The integration with educational institutions and standardized testing services introduces a layer of automation, ensuring swift and accurate verification of academic records and test scores. This not only minimizes the potential for errors but also enhances the overall reliability of the admission process. The following problems need to be taken into account in the solution:

- **Safety and security**

Since the system would be storing personal data, data security should be the primary focus of the solution. Robust security features and role-based data access constraints are fundamental elements of the system.

- **Trust**

Trust is at the core of the University Management System's Admission Module, underpinning its design, functionality, and user interactions. Establishing trust is paramount, as the module deals with sensitive and critical information related to applicants and their academic journey.

- **Privacy**

Privacy is a paramount consideration within the design and implementation of the University Management System's Admission Module. Acknowledging the sensitivity of personal information handled during the admission process, the module places a high premium on ensuring stringent privacy measures. Stringent data protection protocols, including robust encryption for both data transmission and storage, are integrated into the system to safeguard the confidentiality and integrity of applicant data.

- **Server Down or 24/7 Availability in Integrated University Management System**

The University Management System project's success depends on ensuring 24/7 uptime and dealing with potential server downtimes. The platform's constant availability is critical for providing ongoing support, networking opportunities, and resources to both current students and alumni.

- **Backup**

- The system needs to make sure that frequent backups are carried out because it manages a significant volume of data. A proxy server and data storage have to be accessible at all times in case the system needs to be backed up. (Radel, n.d.)

3.3 Comparison among the leading solutions

New technologies appear to stay up with the times and simplify living as the environment changes over time. These days, the world is smaller than a hamlet thanks to the Internet. There are hundreds of different ways to solve the same issue. Although AMS is a unique solution, there are a number of other systems in the same field. I'll analyze and contrast some of the most well-known and well-liked modern websites in this section, outlining their best attributes, advantages, and disadvantages. (Wikipedia, 2021) They are as follows:

- <https://ums.seu.edu.bd/sign-in?redirectURL=%2Fdashboard> (Bangladesh)
- <https://www.ums.bdu.ac.bd/> (Bangladesh)
- <https://gradnet.io/>

SEU Admission

With beautifully designed dashboards filled with comprehensive and eye-catching modules, UMS helps every Southeast University student complete a range of academic and extracurricular tasks. Our main goal is to make this university paperless. I hope you enjoy your time at Southeast University.

Southeast University
Online Admission

Admission Id
Enter the admission id

Program Initiate Basic Info Education Address Confirmation

Please select the department you want to enroll into:
Department of Computer Scienc

PROGRAM NAME	SESSION	DEADLINE	DATE OF ADMISSION TEST	APPLY
BSc in CSE	Spring 2024	N/A	N/A	Apply now
BSc in CSE (Weekend)	Spring 2024	N/A	N/A	Apply now

Figure 1: Overview of SEU Admission

Best Features

- Intuitive and easy-to-navigate interface for both applicants and administrators.
- Streamlined and automated online application process for prospective students, reducing paperwork and manual efforts.
- Capability for applicants to upload necessary documents, and automated verification processes to ensure accuracy and authenticity.

Limitations

- System outages, security vulnerabilities, and internet connectivity issues can disrupt the online application process and create inconvenience for both students and administrators.
- Not everyone has access to reliable internet or digital devices, potentially excluding certain demographics from applying online. Institutions must consider alternative methods for these individuals.

BOU:

A university or other educational institution might use a University Management System (UMS) as a complete platform to integrate different administrative, academic, and operational tasks. Its main objectives.

1) Simplifying Tasks Related to Administration

2) Education Administration

3) Management of Student Information

4) Allocating and Managing Resources

5) Interaction and Cooperation

6) Reports and Data Analysis

7) Openness and Availability

8) Assistance with Distance Education

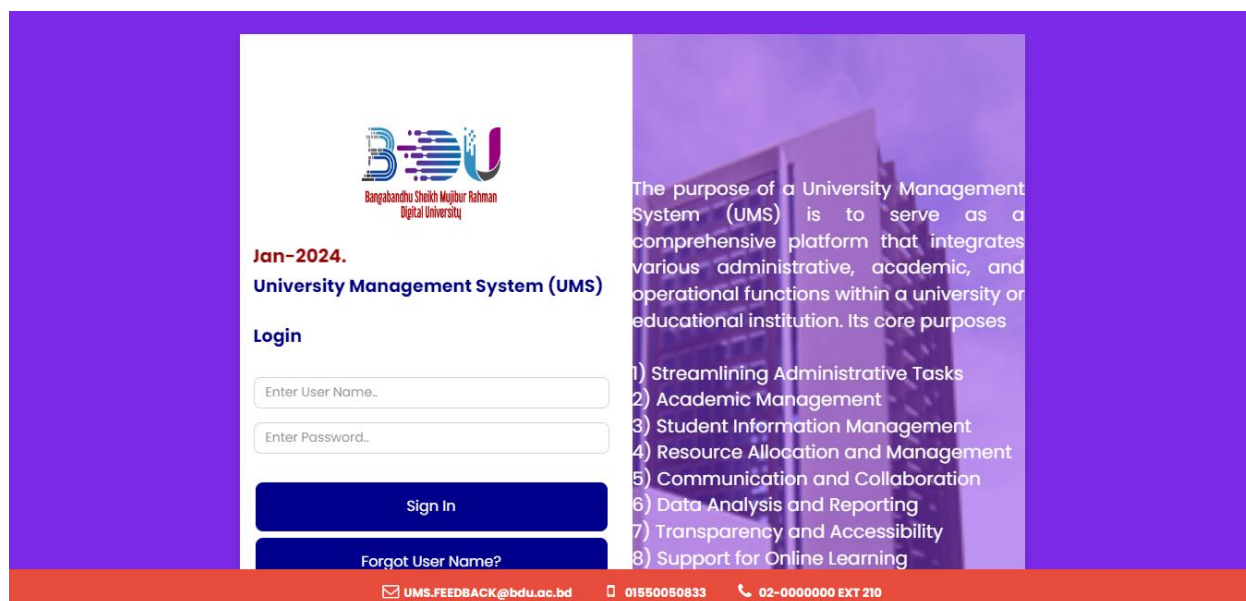


Figure 2: Overview of BOU Admission

Best Features

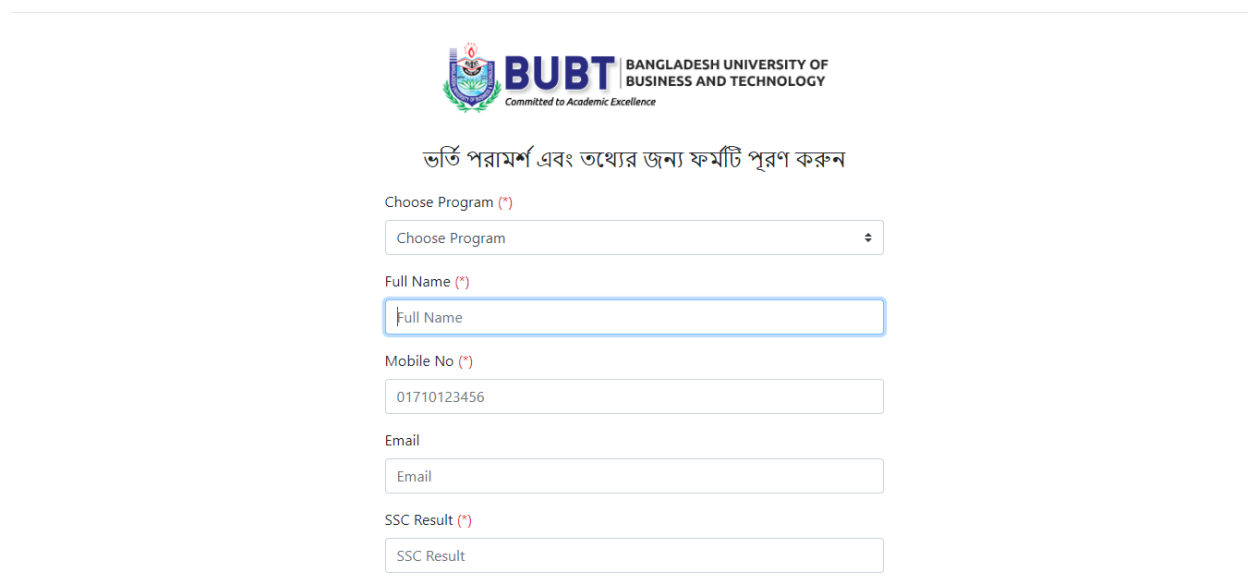
- Seamless integration with academic databases to fetch and verify academic records, making the admission process efficient and reliable.
- Integrated communication system (email or messaging) for sending notifications, updates, and requests for additional information to applicants.
- Customizable multi-stage approval workflow for application processing, involving various stakeholders such as admission officers, department heads, and administrative staff.

Limitations

- During the initial setup, there may be complexities in integrating the system with existing university databases and workflows.
- Adequate training and technical support may be required for users, including administrators, faculty, and applicants, to ensure smooth adoption and operation.

BUBT

The institution already has full automation software produced by BUBT's software development and maintenance staff. We are in charge of the admission and registration software, payroll, HRM, accounting software, and examination software. The group also looks after the attendance software. They also use NXBUBT. Students can view their grades, dues, fees, and other information online with this module. The group is always working to complete the campus automation system.



The screenshot shows the BUBT (Bangladesh University of Business and Technology) admission form. At the top is the BUBT logo with the tagline 'Committed to Academic Excellence'. Below the logo is the text 'ভর্তি পরামর্শ এবং তথ্যের জন্য ফর্মটি পূরণ করুন' (Fill out the form for admission advice and information). The form contains several fields: 'Choose Program (*)' with a dropdown menu showing 'Choose Program'; 'Full Name (*)' with a text input field; 'Mobile No (*)' with a text input field containing '01710123456'; 'Email' with a text input field; and 'SSC Result (*)' with a text input field containing 'SSC Result'.

Figure 3: Overview of BUBT Admission

Best Features

- Online application submission with built-in validation checks.
- Automated document verification system to ensure accuracy and authenticity.

Limitations

- Not easily navigable.
- No choices for real-time searching.

3.4 Recommended Approach

To ensure the success of developing an IUMS ADMISSION SERVICE, a methodical and comprehensive strategy is necessary. The following points outline the recommended approach:

- Initiate the project by conducting comprehensive requirements gathering sessions involving key stakeholders, including admission officers, department heads, administrators, and IT staff. Clearly define the specific needs and objectives of the admission module.
- Prioritize a user-centric design by considering the perspectives of both administrators managing the admission process and applicants navigating the system.
- Adopt an Agile development methodology to facilitate iterative development, allowing for continuous feedback and adjustments.
- Foster collaboration between the IT department and academic departments to ensure seamless integration with existing university databases, academic records, and other relevant systems.
- The method emphasizes building a supportive community through networking events, mentoring programs, and a dynamic virtual platform.
- Prioritize security measures to safeguard sensitive data and ensure compliance with relevant regulations, such as data protection laws.

Chapter 4 – Methodology

The software team that acts in the best interests of the project selects software development approaches to ensure project success and good management. A project management methodology is a collection of concepts, guidelines, and methods developed by professionals in a certain industry. Not only are the greatest strategies designed differently, but they also require particular outcomes, actions, and even the creation of project management software. The chosen methodology and the rationale for implementing the plan after adequate research and comprehension are covered in detail in this chapter. (Nikolaieva, 2018)

4.1 What to Use

Software methods are vital approaches used in the design, construction, and development of systems across the software development life cycle. They are essential in guiding the entire process of developing software. There are numerous methods accessible, and each one approaches the growth process in a different way. A few models include waterfall, spiral, dynamic systems development, agile software development, rapid application development, and collaborative application development. I'll go over the benefits and drawbacks of three solutions below:

Dynamic System Development Method (DSDM)

DSDM employs an incremental and iterative process in contrast to the Waterfall methodology. Throughout the development process, it encourages active user participation, which makes it easier to adjust to changing requirements. For projects where adaptability to changing requirements is essential, DSDM is perfect.

The Dynamic Systems Development Methodology (DSDM), an Agile methodology, takes into account each stage of the project life cycle. The alignment of the project with the strategic goals of the company is a crucial aspect of the DSDM approach. A DSDM project's dual goals are to solve business requirements and produce real financial gains. DSDM makes ensuring that the foundations are in place, the solution is workable, and the advantages are obvious before launching a project. DSDM is an Agile approach that is both powerful and adaptable, allowing it to adapt to

changing business needs. It simultaneously offers the necessary project management and governance tools.

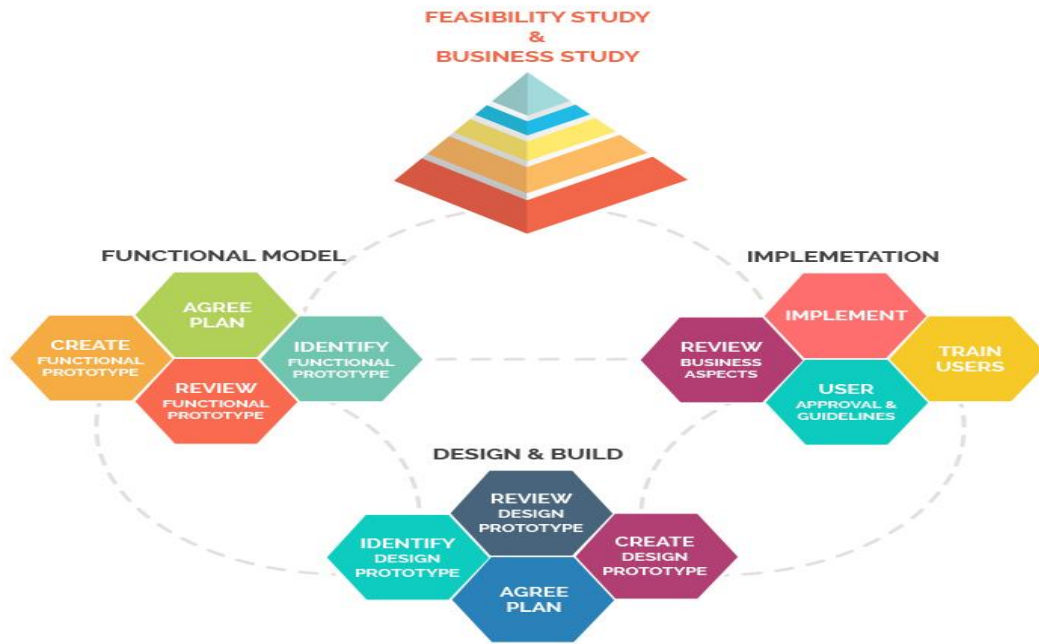


Figure 4: DSDM Methodology

Benefits of DSDM

- The incremental and iterative nature of DSDM enables the timely delivery of high-priority, functional features.
- Significant emphasis on involving end-users throughout the development process.
- Early risk identification and mitigation are made possible by the iterative nature of DSDM. Project failure is reduced by conducting regular reviews and gathering input at each iteration.
- Cooperation and communication between development teams, project managers, and end users, among other parties.
- The continuous testing and validation processes in DSDM contribute to maintaining high-quality deliverables.

- DSDM provides a structured framework that allows for better project control and predictability.(Ltd}, 2021)
- DSDM encourages the delivery of business value at each iteration. This guarantees that the project team is always developing features that complement the strategic objectives of the company.

Drawbacks of DSDM

- DSDM may be perceived as too elaborate for small projects with straightforward requirements.
- The success of DSDM relies heavily on continuous and active involvement of end-users throughout the development process.
- The framework may struggle to provide sufficient guidance and structure for handling extensive requirements and intricate project landscapes.
- In organizations with a hierarchical or siloed culture, achieving the required level of collaboration and communication can be a significant challenge. (Ltd}, 2021)

Waterfall Model or Structured System Analysis and Design Method (SSADM)

The Waterfall model, sometimes referred to as the Waterfall technique, is a methodical approach to project development that entails finishing each phase of a project—such as analysis, design, programming, and testing—before proceeding to the next. The waterfall technique is one project management strategy that emphasizes the project's linear flow from beginning to end. Engineers are typically the ones who use this front-loaded method, which is predicated on extensive preparation, voluminous documentation, and consistent execution. (Anon., 2021)

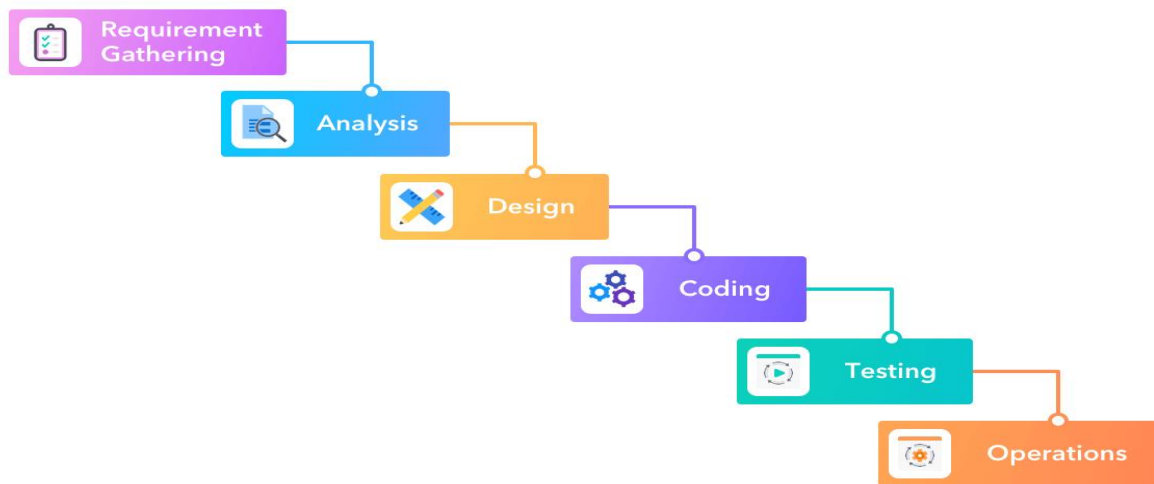


Figure 5: Waterfall Methodology

Benefits of Waterfall

- The Waterfall model provides a structured and easily understandable framework with distinct phases, including requirements, design, implementation, testing, deployment, and maintenance
- In the Waterfall model, the requirements are typically gathered and documented thoroughly at the beginning of the project.
- Since the Waterfall model follows a sequential and phased approach, it is easier to estimate project timelines and costs. (Sherman, 2015)

Drawbacks of Waterfall

- Waterfall's sequential, phase-gated approach struggles to adapt to changing requirements or market conditions. Any adjustments often require backtracking, leading to delays and rework.
- Testing is relegated to later stages, potentially revealing critical flaws only after significant time and resources have been invested. This makes corrections costly and time-consuming. (Sherman, 2015)

Rapid Application Development

Prototyping and iteration take precedence over meticulous preparation in the Rapid Application Development (RAD) paradigm. In the RAD method to software development, planning is typically deferred until development and prototyping. While the RAD technique lays more attention on demands that vary continuously as development progresses, the waterfall paradigm places a strong emphasis on planning and precise specification. (Idesis, 2020)

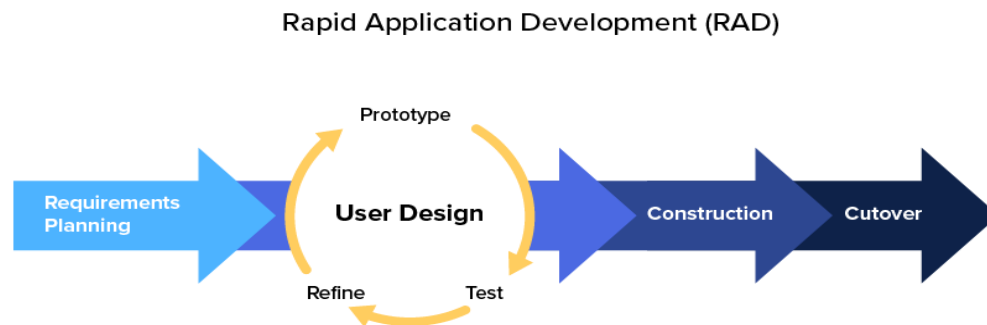


Figure 6: RAD Methodology

Benefits of RAD

- It is adaptable and versatile, and it may help you lower the project's total risk.
- Both flexibility and adaption.
- Deliverables can be transferred easily thanks to intermediary programs, high-level abstractions, and scripts.
- Code generators and reuse have reduced the amount of manual coding, and prototyping is predicted to lower mistake rates. (Martin, 2021)

Drawbacks of RAD

- It need strong collaboration to succeed.
- I have trouble working in large groups.
- There is a need for developers with advanced skills.

- Only applications that are intended to be modularized over the course of the product's life cycle may be created fast (Martin, 2021).

Choosing Methodology

The provided content explains the "Integrated University Management System" initiative, which attempts to connect alumni's and students' educational paths. With a focus on objectives, aims, and key components, it outlines the project's original research, feasibility study, and core elements. Additionally, covered are the iterative development-timeboxing and MoSCoW prioritizing methodologies.

The methods used in software development for a farm management system are covered in the paragraph that follows. It affirms the occasional use of the waterfall model but asserts the superiority of the Dynamic Systems Development Method (DSDM). Since the DSDM strategy routinely produces high-quality goods and projects on schedule, it is the best option for this project. It also highlights how flexible the methodology is to different businesses and institutions. The flexibility of DSDM, which enables clients to modify their needs during the development phase, is highlighted in the concluding paragraph.

In project development, selecting a methodology is crucial, and the paper suggests that the DSDM approach is the best because of its flexibility, timeliness, adaptability, and client-centeredness.

4.2 Why to Use Methodology

The "University Management System" project's effectiveness depends on a methodical approach. A methodology guides the team through every phase of the project, from early research and feasibility studies to implementation and deployment, providing a rigorous approach to project management. The method chosen in the context of this project will be crucial to ensuring productivity and achieving project goals within the given constraints. A technique functions as a road map, offering a clear course to follow while dealing with the complexity of tasks like creating an alumni database, planning networking events, and putting in place a virtual platform. It aids in

reducing any roadblocks, encouraging teamwork, and maintaining focus on the project's main objectives.

4.3 (IUMS) Sections of Methodology

The development team is required to adhere to certain established DSDM methodology elements. Among them are the following:

Phase one of the project

The fundamental needs, budget, schedule, and project idea are all created at this phase.

Feasibility Study Phase

The project budget, technological solutions, business strategy, and business case are all supported at this phase.

Requirement Gathering Phase

A range of approaches are used in this section to determine the functional and non-functional project demands.

Requirement Analysis and Prioritization Phase

This step uses a prioritizing technique like MoSCoW to analyze and rank the acquired needs.

Using MoSCow prioritization:

A tool for prioritizing has been created by MoSCoW to help with system development. The following is a list of the parts:

Must have:

This part is essential for identifying the primary tasks that the system needs to do; without these tasks, the users won't be able to utilize the system.

Should have:

This section aids in the identification of critical parameters and the creation of an all-encompassing key.

Could Have:

A few demands that are unnecessary and unrelated to the system may be found with the help of this section.

Won't Have:

This system element aids in locating demands that the system is unable to satisfy.

Exploration and Engineering Phase

Engineering: Iterative testing, system increments, system design, and making sure that user demands are satisfied in line with system requirements are all part of the engineering process.

Exploration: To determine the functional and non-functional needs of the system, this phase employs MoSCoW prioritizing.

The requirements were analyzed and the solution was designed iteratively using the primary iterative component of the process.

Post- project:

This stage assists in establishing the ultimate desired solution for the system and estimating the project's benefits.

Review Phase

Users test the created deliverables in this step, and if any adjustments are required, they go back to the earlier stage. (Sigma X, 2021).

4.4 Implementation Plans

The project is nearly complete, and users are able to access and operate complex applications. If there are any issues with this identification and resolution, utilize the new system. This section

contains a full description of the planning, configuration, and release criteria. The new system is made available after everything is in working condition.

Chapter 5 – Planning

5.1 Project Plan

This section covers the planning process for the project's completion. Essentially, the project is divided into many independent sections, and if everything proceeds as planned, all of the work will be completed within the designated time limit. (Managing Project, 2015)

5.1.1 Work Breakdown Structure

During this phase, the project will be divided into smaller components in order to be finished more quickly, effectively, and within the allocated time frame. We can estimate the time and tasks with this approach. It could be harder to finish the job without this structure. As a consequence, the WBC has divided the suggested system into groups and subcategories, as shown in the chart below:

No.	Task Name	Start Date	Finish Date	Duration (Days)
1	Introduction	01/07/2023	10/07/2023	10
2	Initial Study	11/07/2023	25/07/2023	15
3	Literature Review	26/07/2023	15/08/2023	21
4	Methodology	16/08/2023	05/09/2023	21
5	Planning	06/09/2023	19/09/2023	14
6	Feasibility Study	20/09/2023	03/10/2023	14
7	Foundation	04/10/2023	17/10/2023	14
8	Exploration & Engineering	18/10/2023	31/10/2023	14
9	Deployment	01/11/2023	14/11/2023	14
10	Testing	15/11/2023	28/11/2023	14
11	Implementation	29/11/2023	12/12/2023	14
12	Critical Appraisal & Evaluation	13/12/2023	26/12/2023	14
13	Lesson Learned	27/12/2023	02/01/2024	7
14	Conclusion	03/01/2024	09/01/2024	7
15	Total	01/07/2023	10/01/2024	194

Table 1: IUMS ADMISSION SERVICE Work Breakdown Structure

5.1.2 IUMS ADMISSION SERVICE Resource Allocation

To guarantee the project is completed efficiently and on time, every resource and asset is assigned and controlled. The distribution of resources is an essential component of project management. As this is an independent intellectual undertaking, I will assume varying responsibilities at different times. To meet the job delivery date, the University Management System project will allocate its resources as follows. (2020, Dopson)

Task Name	Duration	Resources
Introduction	10 days	Analyst
Initial Study	15 days	Analyst
Literature Review	21 days	Analyst
Methodology	21 days	Analyst
Planning	14 days	Analyst
Feasibility Study	14days	Analyst
Foundation	14 days	Designer
Exploration & Engineering	14 days	Developer
Deployment	14 days	Developer
Testing	14 days	Tester
Implementation	14 days	Developer
Critical Appraisal & Evaluation	14 days	Analyst
Lesson Learned	7 days	Analyst
Conclusion	7 days	Analyst

Table 2: IUMS ADMISSION SERVICE Resource Allocation

5.1.3 Time Boxing

The division of tasks into time boxes to guarantee that the objectives are fulfilled by the deadline is another crucial component of DSDM project planning. All jobs are grouped into time boxes in this section with predetermined durations. The iterative technique's time constraints must be followed in order to complete these tasks.

Time Box	Task Name	Duration	Resources
TB1	Introduction	10 days	Analyst
TB1	Initial Study	15 days	Analyst
TB1	Literature Review	21 days	Analyst
TB2	Methodology	21 days	Analyst
TB2	Planning	14 days	Analyst
TB2	Feasibility Study	14days	Analyst
TB3	Foundation	14 days	Designer
TB4	Exploration & Engineering	14 days	Developer
TB5	Deployment	14 days	Developer
TB5	Testing	14 days	Tester
TB6	Implementation	14 days	Developer
TB6	Critical Appraisal & Evaluation	14 days	Analyst
TB7	Lesson Learned	7 days	Analyst
TB7	Conclusion	7 days	Analyst

Table 3: Time Boxing of IUMS ADMISSION SERVICE

5.1.4 Gantt Chart of University Management System Working

An activity schedule for a project is represented graphically by a Gantt chart. Instead of displaying the length in days, it does so as a progress bar from the start to the finish date. Below is a Gantt chart showing the University Management System:

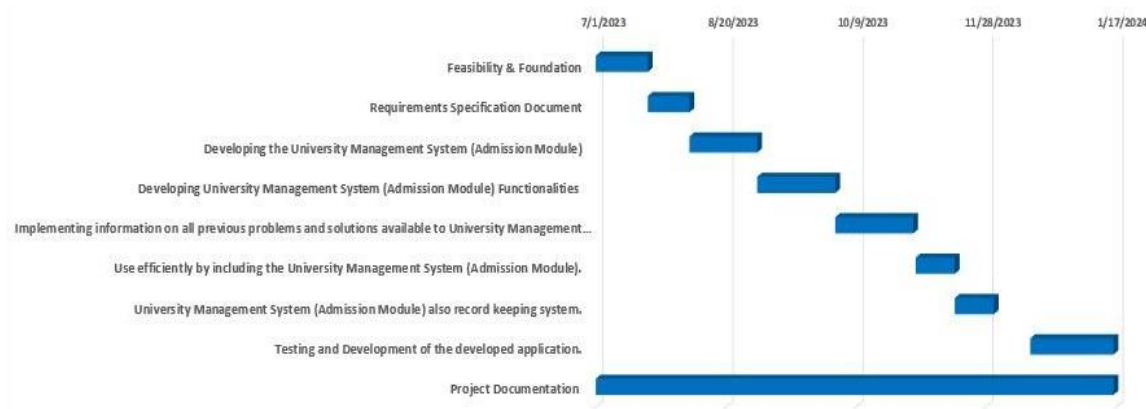


Figure 7: Gantt Chart

5.2 Test Plan

A test plan is an all-inclusive document that lists the objectives, schedule, budget, materials, and testing methodology needed to complete a project. Consider it as a guide for doing the required tests, overseen by test managers, to confirm that the software is operating as intended. The test plan was created in response to discrepancies between the input received and the desired output of the system. Two types of testing that occurred during the software development process were validation and verification.

5.2.1 IUMS ADMISSION SERVICE Against the Time Boxes

For a given segment, a fixed and maximum unit of time was estimated using the time box approach. Time constraints are being examined.

User Name for testing purpose	Example for testing	Role for testing	Example for tes
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				

Table 4: IUMS Sample of Testing Against the Timebox

5.2.2 Required Test

There are numerous testing modules available, however the two most popular types of system testing are functional and non-functional testing:

Functional Testing:

- **Unit Testing:** Each software unit's suitability for unit testing is evaluated. Use processes, operational procedures, and sets of computer coded modules are a few examples of these. The developer examines each module independently using this testing method to check for issues. This is determined by how well the various components work together.
- **Integration Testing:** Analyzing the relationships between two software modules or components is the discipline of integration testing. Making sure the interface is operating correctly is its main goal. Integrity testing is one way of finding gaps in the communication between connected elements. Unit testing is followed by integration testing for every module.
- **System Testing:** Software is tested on an integrated system as a whole to see if it satisfies the requirements. This technique is known as system testing. System testing uses components as input that pass integration testing. Integration testing looks for differences between the connected components. System testing looks for issues with the system overall as well as its specific parts. The observable behavior of a system or component during testing is produced by system testing.
- **Acceptance Testing:** Formal testing of this type is carried out in accordance with business protocols, user requirements, and expectations in order to determine if a system satisfies acceptance criteria and to allow users, clients, or other authorized organizations the choice to accept or reject the system (GeeksforGeeks, 2019).

Non-Functional Testing:

- **Security Testing:** Web application security testing is done in addition to external security testing. It prevents risks from the inside as well as the outside, such as SQL injections, by validating unauthorized users or access.
- **Usability Testing:** The main advantage of usability testing is its ability to assist designers in identifying usability issues early on in a design, enabling them to be resolved prior to approval or production implementation. Because of this, usability testing typically uses prototypes—varying in fidelity depending on the stage of production—rather than finished things. . (GeeksforGeeks, 2019).

Testing for Reliability: This testing allows users to communicate with it directly. It guarantees steady system operation. It evaluates and verifies the many kinds of system faults.

5.2.3 Test Case of IUMS

The test case included the test number, test type, test description, and test methods.

No. of Test Case			
Type of Test			
Description of the test			
Steps of testing	Expected outcome	Actual outcome	Comment

Table 5: Test case of IUMS

5.2.4 User Acceptance Test Plan

The last stage of the testing segment is user acceptability. It refers to and verifies the user's performance during the test.

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: User acceptance sample of IUMS

5.3 Risk Management of University Management System

The process of locating, evaluating, and reducing any risks that can arise throughout the development of a project is known as risk management. It's used in project tracking to make sure goals and objectives are met. Project performance, budget, and schedule are a few examples of variables that could be at risk. There are several tactics that can be used to control risk.

- Identification of risks
- Evaluation of risks
- Proactive steps and a plan of action
- Safeguards against possible dangers

5.3.1 Risk Identification

Risk identification is one of the most crucial components and duties of a risk management approach. Making sure that all potential risks that could affect the project are recognized, recorded, and that each of their components is recognized and recorded is the first stage in the risk identification process. In light of this investigation, the following risk identification criteria will be considered reasonable:

- Possible hazards should be identified and recorded.

- The risks associated with the highlighted dangers should be evaluated in light of their real causes, as well as any potential future consequences and effects.

Detailed description of possible risk identification and risk monitoring for the project, including the following risks:

Types of dangers	Causes	Consequences & Impact
Technical Risks	Lack of experienced developers	Delays in project timeline, poor system performance, and potential technical failures during deployment
Requirement Changes	Changes in university admission policies or regulations	Scope creep, increased development time, and potential conflicts with stakeholders
Resource Constraints	Insufficient budget or human resources	Delays in development, compromised project
System Technical Problem Hardware	- Not travelling at the required speed - Due to a major performance limitation or lack of necessary setup.	One can hope that the delivery of the goods would go smoothly.
Project Backup for Security	The project is not backed up on GitHub or any other similar platform.	Cost implications: The project is completely lost.

Failure of the Network	Internet connectivity is required for a number of integrated API and email management features.	It is prohibited to send emails to former students and alumni.
------------------------	---	--

Table 7: Risk identification of IUMS

5.3.2 Risk Assessment of IUMS

Risk assessment involves with quantitative or qualitative value of risks included threats of a project

Type of risks	Likelihood	Impact	Restore time
Technical Risks	5	6	8
Requirement Changes	5	5	5
Resource Constraints	6	6	6
System Technical Problem ➤ Hardware	4	5	5
Project Backup for Security	5	6	4
Failure of the Network	4	4	8

Table 8: Risk assessment of IUMS

5.3.3 Risk Precaution / Action Plan

The dangers were identified and assessed, and then a risk action plan was created. There are several ways to prepare for risks.

- To lessen the hazards that might arise in the future as well as those that have already happened;
- To deal with the problems that have surfaced and any possible risks;
- To abstain from taking chances that might result in anything or have already happened.

Type of risks	Action	Taking Action	Action Required
Technical Risks	Using a Gantt chart and other relevant tools, create precise time estimates. Using technology: Asana and Microsoft Excel	A developer who takes an active part in	Before we start working on the project, make sure everything we need is present.
Requirement Changes	➤ To complete the necessary normalization ➤ To establish the exact relationship that is required on the table,	A developer who takes an active part in	Concerning the addition, modification, and removal of data,
Resource Constraints	➤ To write code without sacrificing a	A developer who takes an active part in	After the components for authentication have been configured

	pleasing aroma. ➤ Provide a multi-authentication mechanism for this project.		
Following the configuration of the authentication components			
Security Project Backup	Project backups may be made using any cloud service, such as GitHub.	A developer who takes an active part in	Update
The Network's Failure	to engage in fruitful dialogue with network service providers.	A developer who takes an active part in	In order to do the work in time.

Table 9: Risk precaution of IUMS

5.3.4 Steps Taken for Possible Risks of UMS

The responsible person adheres to the necessary procedures after identifying and evaluating any potential dangers. They are listed in the following order:

Type of risks	Description	Likelihood	Impact	Taking Step
Technical Risks	One of the biggest risks is not finishing	Rare	High	The project time boxing was

	the assignment by the deadline.			completed, along with the creation of the Gantt chart and task breakdown structure.
Requirement Changes	It's possible that more than one entry will be made.	Likely	High	The data dictionary was created using the entity relationship diagram and third normal form normalization.
Resource Constraints	Entry into the restricted network.	Unlikely	High	Developed a practical system for access permission and authentication.
Technical Issue with the System ➤ Hardware	We cannot rule out the possibility of a malfunctioning equipment.	Frequent	Medium	In addition to continuous maintenance and oversight, backup is ensured.
Project Backup for Security	GitHub must be used to create a backup of each project file.	Rare	High	monitoring the project file securely.
Failure of the Network	An internet problem or a lost connection.	Frequent	Low	The proxy server has been set up, and high-bandwidth access has been authorized.

Table 10: Steps Taken for Possible Risks of UMS

5.4 Change Management

5.4.1 Factors that Might Cause Change

There are several types of changes needed for this system which are included:

- New regulations or updates to existing ones related to university admissions might necessitate changes in the system to ensure compliance.
- Changes in admission policies, regulations, or stakeholder expectations may lead to modifications in system requirements.
- Emergence of new technologies or updates to existing ones might necessitate adjustments to the system for improved functionality or security.
- New technologies or updates in existing platforms may present opportunities to enhance system capabilities, prompting changes for improved efficiency and performance.
- Security handling has changed in some ways.

5.4.2 DSDM Welcome Change of IUMS

The project requirements should guide any necessary modifications to the project development life cycle. These factors influence our decision to select a method that, given the constraints of the system, can handle a variety of issues. As a result, "DSDM" is a contemporary, trustworthy, and user-friendly method. This method is used by this system to manage and apply changes to the development time.

It operates in a variety of ways:

- The project is divided into increments, and each increment delivers a portion of the required functionality.
- DSDM uses time-boxing to allocate fixed time periods (time-boxes) to various project activities
- It allows for the emergence of solutions over time, incorporating feedback and making adjustments as the project progresses.
- DSDM supports the concept of empowered and self-organizing teams.

5.4.3 Considering Business Priority

To achieve the intended purpose and vision of the organization, modifications have been looked into. Changes in functionality and requirements that may materially benefit consumers must be continuously monitored.

Priorities were as follows:

Topical Area	Priority Level
Delivering Segments	6
Functional requirements	8
Non-Functional requirements	6
Segments for University Management	5
Segments for security handling	8

Table 11: Business priorities

5.4.4 Change Workshop

In order to discuss and assess suggested system modifications, important stakeholders, project team members, and pertinent specialists gather in a Change Workshop for the IUMS ADMISSION SERVICE project. This dynamic session fosters open communication, idea sharing, and decision-making, ensuring that changes align with the project's objectives and stakeholder expectations. The workshop typically begins with a comprehensive review of the proposed changes, highlighting their potential impact on project timelines, budget, and overall scope. Stakeholders are encouraged to express their perspectives, providing valuable insights into the necessity and implications of the proposed modifications.

- Give a clear explanation of the Change Workshop's aims, highlighting the necessity of coordinating suggested changes with project objectives and stakeholder expectations.

- Make sure important parties are represented, such as end users, IT specialists, admissions officials, and university administrators, to encourage a variety of viewpoints on the modifications that are being suggested.

5.4.5 Changes That Are Allowed

In the dynamic landscape of the IUMS ADMISSION SERVICE project, a structured approach is adopted to accommodate changes that enhance the system's functionality and align it with evolving requirements. The project team understands the value of adaptability and that, with proper management, adjustments are expected and can even improve the project's final outcome.

5.4.6 Key Decision Taker of Change

Change agents will get assistance from specialists in several sectors. In this project, several decision-makers for modifications were involved.

- Analyst for System Requirements
- System Developer
- System Tester

Both the analyst and the developer are able to modify the functional requirement and the relevant code as needed.

5.5 Quality Management

Maintaining the standard that the user accepted at the start of the project is the aim of quality management. Quality management is built on a number of components, including.

- Begin with a thorough validation of system requirements, ensuring they are clear, complete, and aligned with the needs of university administrators, admission officers, and applicants.

- Adhere to industry best practices and quality standards, such as ISO 9001, to establish a framework for quality management processes throughout the project lifecycle.
- Establish a thorough user acceptance testing (UAT) procedure that involves end users to confirm that the system satisfies their needs and to find and fix any inconsistencies or problems prior to deployment.

5.5.1 Rules Applied to Maintain Quality

The rules for upholding quality differ based on the user and the system. Control and quality assurance are two ideas that must be followed in order to maintain quality.

By evaluating farmers' issues, product buyer and seller service quality control has been preserved.

- Incomplete or ambiguous specifications might cause miscommunication, mistakes, and more effort later on in the project.

Quality Assurance: By examining the justifications for both good and incorrect approaches, quality assurance has been preserved.

- Make a comprehensive testing plan that includes system, integration, unit, and user acceptability testing. This ensures that every component of the system is thoroughly examined for accuracy and functionality.
- Install automated testing frameworks and technologies to accelerate the testing process.

5.5.2 DSDM Standard Quality Measures

Solution quality: By monitoring user needs in light of anticipated business demands and user expectations, solution quality has been guaranteed. Two different priority strategies were applied in order to finish this procedure.

- Time Boxing

➤ MoScow

The timetable has been adhered to and both of these approaches have been finished.

The chosen organization has guaranteed the process's quality. Two tactics have been developed and put into use to guarantee the caliber of these procedures.

➤ CMMI

➤ DSDM

5.5.3 Quality Plan and Measuring Meter

As of right now, a number of high-quality plans have been maintained, including

- The quality objectives and benchmarks for the project must to be well-defined and disseminated.
- Ensure that the development team adheres to design principles, industry best practices, and coding standards throughout the project.
- Implement a continuous monitoring system to track and assess the system's functionality, user experience, and code quality.

Chapter 6 – Feasibility

6.1 All Possible Types of Feasibility IUMS

In order to ensure the successful implementation of the IUMS project, a number of feasibility considerations are made. Operational feasibility focuses on how practical the idea is to implement and highlights collaborating with real industries to give new students relevant work culture experiences. A comprehensive assessment of the institution's technical capabilities is required to determine technological viability before establishing a virtual platform for networking and mentoring. Establishing mentorship programs and sustaining the online platform require a thorough assessment of the project's financial requirements and the organization's funding capacity. We call this technique financial feasibility. Market research analysis looks at competitor activity and target audience need in order to effectively meet alumni and student expectations. (THE INVESTOPEDIA TEAM, 2021)

Technical Feasibility:

A few elements of technological viability include:

The IUMS project's technological feasibility is a critical component for the effective implementation of the recommended initiatives. Since the project is focused on creating a virtual platform for networking, communication, and mentorship between students and alumni, a thorough evaluation of the institution's technical resources and competencies is essential. This evaluation is required to guarantee that the platform and its features are implemented without a hitch. Utilizing HTML, CSS, Bootstrap, Angular, and Java demonstrates a robust and contemporary technology stack that enables the development of an interactive and dynamic application. Furthermore, a well-designed database and JavaScript integration into the backend code ensure data integrity and system validity. (ogbebor, 2011)

So, clearly in this project perspectives the technical aspects are given below:

Hardware:

- HP laptop (Configuration)
- Wi-fi Router(Tp-link)

Software:

- Microsoft office(MS)
- Microsoft Excel
- Google chrome browser
- Windows 11 Pro (operating system)
- VS Code

Database:

- MySql

Technology:

Design Slide:

- Html
- CSS
- Angular
- Java

Server Side:

- Java

Economic Feasibility:

There are other approaches to construct the proposed system, such as creating desktop and online applications.

- The cost of a web-based application:

All of the data and the complete application are stored on a remote server that is always and everywhere reachable over the Internet.

Equipment	Cost per unit	Cost
VPN enabled extranet network	₹2000 per/m	₹2,000
Laptop (core i5, 2.50-271 GHz processor, 4 GB DDR4 RAM, 1 TB HDD)	₹ 85,000	₹ 85,000
Web, File and Email servers	₹17,000 per/m	₹17,000
Total		₹ 1,04,000

Table 12: Web based application cost

Equipment	Cost per unit	Cost
Laptop (core i5, 2.50-271 GHz processor, 4 GB DDR4 RAM, 1 TB HDD)	₹ 85,000	₹ 85,000
Web, File and Email servers	₹17,000 per month	₹17,000
Total		₹ 1,02,000

Table 13: Desktop application cost

You will need a domain name and hosting company in order for the system to function. Because the system has to continuously solve issues and track the whereabouts of the objects in order to meet the expectations of the problem solver and delivery search. To operate the remainder of the system, we will create a web application.

Market Research Analysis Based on the Feasibility Factors:

The feasibility of implementing a University Management System with a dedicated Admission Module is a critical aspect that requires thorough market research analysis. Several factors contribute to the feasibility of such a system within the higher education landscape. Firstly, assessing the technological infrastructure of universities is paramount. Institutions with robust IT infrastructure and a willingness to embrace modern solutions are more likely to seamlessly integrate a sophisticated Admission Module. Additionally, the existing market for similar systems should be analyzed to identify potential gaps or areas of improvement that the proposed module could address. To make sure that the University Management System stands out and satisfies the particular requirements of various educational institutions, it is necessary to comprehend the functions, features, and user experience offered by rival systems.

6.2 Cost Benefit Analysis for IUMS ADMISSION SERVICE

Determining the profit and spending assumptions is the main objective of a cost-benefit analysis. The benefit may be found by comparing the total cost and earnings. (HAYES, 2021) The following is the project's cost-benefit analysis:

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	₹2,00,000
5	Other Cost	₹20,000	₹20,000	₹20,000	₹20,000	₹20,000	₹1,00,000
6	Total Cost	₹ 291,000	₹ 85,000	₹ 85,000	₹ 85,000	₹ 85,000	₹ 631,000

Table 14: Total cost of IUMS

Total Earning:

SL 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. Tax two	₹2,00,000	₹2,10,000	₹2,40,000	₹2,60,000	₹3,00,000	₹1,210,000
	Total	₹3,50,000	₹3,90,000	₹4,50,000	₹5,00,000	₹5,70,000	₹2,260,000

Table 15: Total earning of IUMS

Total Revenue of project:

SL No.	Sectors	First Year	Second Year	Third Year	Fourth Year	Five Year	Total
1	Total Earning	₹3,50,000	₹3,90,000	₹4,50,000	₹5,00,000	₹5,70,000	₹2,260,000
2	Total Equipment Cost	₹4,02,000	₹80,000	₹80,000	₹80,000	₹80,000	₹7,22,000
	Total Revenue	₹4,02,000	₹3,10,000	₹3,70,000	₹4,20,000	₹4,90,000	₹1,538,000

Table 16: Total revenue of IUMS

Thus, the government would be able to obtain a sizable sum of money annually from several Agriculture Departments, including Farmer, by supporting this method. Funds will be raised annually. As so, both the government and the general public will gain from this effort.

6.3 Explain DSDM Good or Bad for this Project

The feasibility of implementing a University Management System with a dedicated Admission Module is a critical aspect that requires thorough market research analysis. Several factors contribute to the feasibility of such a system within the higher education landscape. Firstly, assessing the technological infrastructure of universities is paramount. Institutions with robust IT infrastructure and a willingness to embrace modern solutions are more likely to seamlessly integrate a sophisticated Admission Module. Additionally, the existing market for similar systems should be analyzed to identify potential gaps or areas of improvement that the proposed module could address. To make sure that the University Management System stands out and satisfies the particular requirements of various educational institutions, it is necessary to comprehend the functions, features, and user experience offered by rival systems.

Furthermore, regulatory compliance is a key consideration. The system must align with prevailing educational policies and adhere to data protection regulations to guarantee legal feasibility. Understanding the financial implications is crucial as well, as universities may have varying budgetary constraints. A cost-benefit analysis should be conducted to demonstrate the system's economic viability and showcase the potential return on investment through improved efficiency and reduced administrative overhead.

Chapter 7 – Foundation

7.1 The Problem Area Identification

The majority of software is created in reaction to an issue. Therefore, the true problem needs to be found in order to design software that is usable. It is hard to devise a specific solution without first knowing what the problem is. Because of this, figuring out the problem areas is essential to building the system. The software company receives information from users in this domain. Since the users will be utilizing the solution and are aware of the particular issue for which they need the solution, their information is more appropriate for building the solution. There are several methods for identifying the problem. The two that are offered provide the basis for this response:

7.1.1 Interview

A face-to-face discussion about a specific problem or possible solution between two or more persons is called an interview. It's one of the best techniques for figuring out issues.

7.1.2 Questionnaires

The questionnaire is an additional data collection approach. We refer to this kind of interview as a written interview. In this process, interviewers utilize a series of questions to get data from the consumer.

Note: The issue area identification part has been completed by the user and the DCL analyst. My internship at DCL (Daffodil Computers Limited) has given me extensive knowledge of the used approach.

7.2 Overall Requirement List

The criteria are mostly divided into two sections:

- I. Functional Requirements
- II. Non-Functional Requirements

7.2.1 Functional Requirements

The function that a piece of software must do is specified in the Functional Requirement (FR). This describes a software system or a portion of it. The software system's operations and outputs are all that make up a function. Stated differently, a system's purpose is irrelevant as long as it has the ability to compute, manage data, and communicate with humans. The functional specification is a different kind of functional requirement used in software engineering. (Martin, What is a Functional Requirement in Software Engineering? Specification, Types, Examples, 2021)

The system should support secure user authentication mechanisms for administrators, admission officers, and applicants.

- Different user roles (e.g., admin, admission officer, applicant) should have specific access rights and permissions.
- Applicants should be able to create accounts, providing necessary personal and academic information.
- Applicants should be able to submit online applications through the system.
- The system should capture and store application data, including supporting documents.
- Admission officers should have tools to review and process applications.
- The system should facilitate the evaluation of academic credentials, test scores, and other relevant criteria.

7.2.2 Non-Functional Requirements

The specification of the quality of a software system is defined by the Non-Functional Requirement (NFR). These non-functional criteria are evaluated according to their portability, security, usability, and responsiveness. They are essential to the software system's success. What is the website loading speed? is a nonfunctional demand, for instance. Systems may fall short of user expectations if they do not meet non-functional criteria. (Martin, What is Non-Functional Requirement in Software Engineering? Types and Examples, 2021)

- The system should respond to user interactions within a maximum of 2 seconds for normal operations.
- The system should be scalable to handle a 20% increase in user load over the next five years.
- The system should be capable of running continuously for a minimum of 30 days without critical failures.

7.3 Technology to Be Implemented

Numerous technology options are available for implementing the IUMS ADMISSION SERVICE system. To be successful, choosing the right technology is essential. Web technology, also referred to as web application technology, will be employed. Anybody with an internet connection can access this server-based system. Because it can be accessed from anywhere in the globe without the need to download or install extra software, today's web technology is incredibly advantageous. There are two parts to the online system: an administrator component and a client component. The interface design is reasonable and the information is sufficient in both situations.

Client-server Application

"Client-server" refers to a connection between collaborating programs that, within the framework of an application, function as both clients and servers. Through the use of a computer network or

the Internet, the client-server paradigm, also known as "client-server architecture," is an application framework that distributes work between servers and clients. By submitting a request to another piece of software, the client asks the server for access to a service. One or more applications on the server distribute and share resources with the clients. (ApacheBooster, 2020)

Benefits of client-server application

- Having all the necessary data on one server makes it easier to protect and manage user access and authentication.
- When a client-server network is expanded with more computers, servers, or network segments, there aren't any significant disruptions.
- Customers and servers do not need to be in close proximity for efficient data access to occur.
- Because each node in a client-server system is completely autonomous and only communicates with the server to acquire data, nodes may be updated, replaced, or transferred with ease.

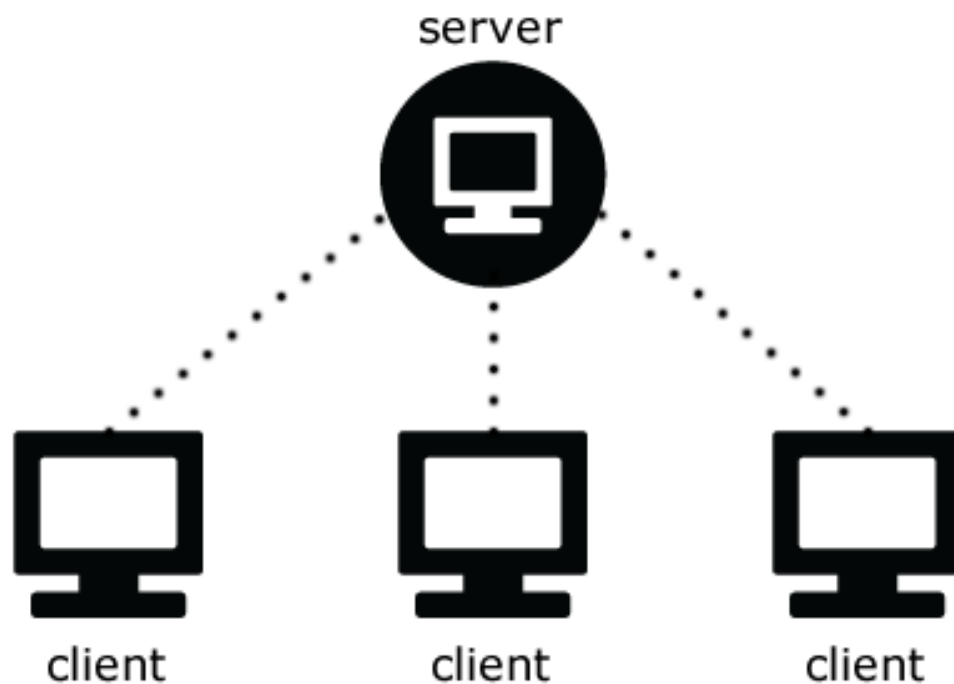


Figure 8: Client server application

The Web-based Application

Web-based applications are computer programs that carry out specific functions using web browsers and the internet, according to computer science. Web applications store and retrieve data using server-side languages like JavaScript and HTML, and client-side languages like HTML. Customers can interact with the company in this fashion by completing online forms, updating information, making purchases, and more. A further advantage of the program is that it allows employees to create, share, and collaborate on documents from several devices and places. (contributors, 2021)

Benefits of the Web application:

- Web applications can be accessed and utilized on multiple platforms and operating systems provided that the browser is compatible.
- Since everybody has access to a single version, there are no incompatibilities.
- They don't occupy any hard drive space, so there are no limitations.
- Because of its implementation, subscription-based web apps are using less pirated software.
- Both the business and the end user save money since there is less demand for the end user's PC and less support and maintenance needed from the company.

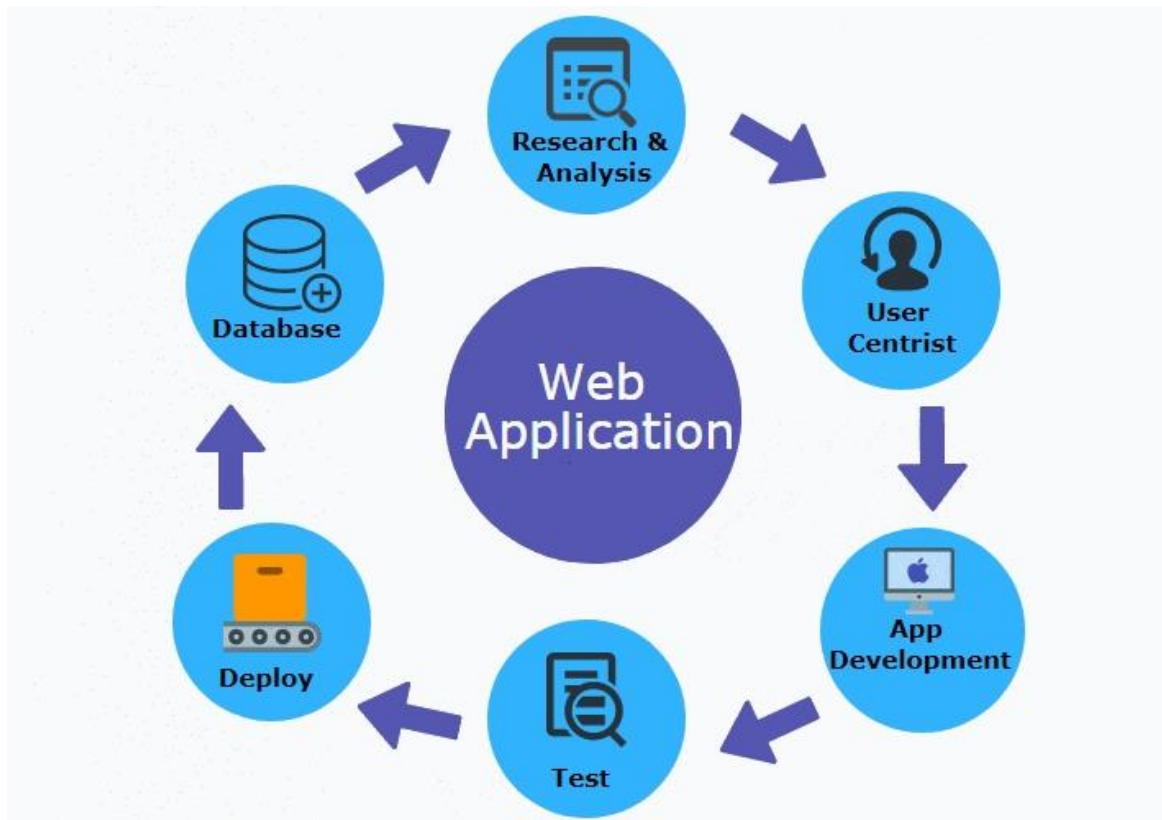


Figure 9: Web based application

Chapter 8 – Exploration

8.1 Activity Diagram

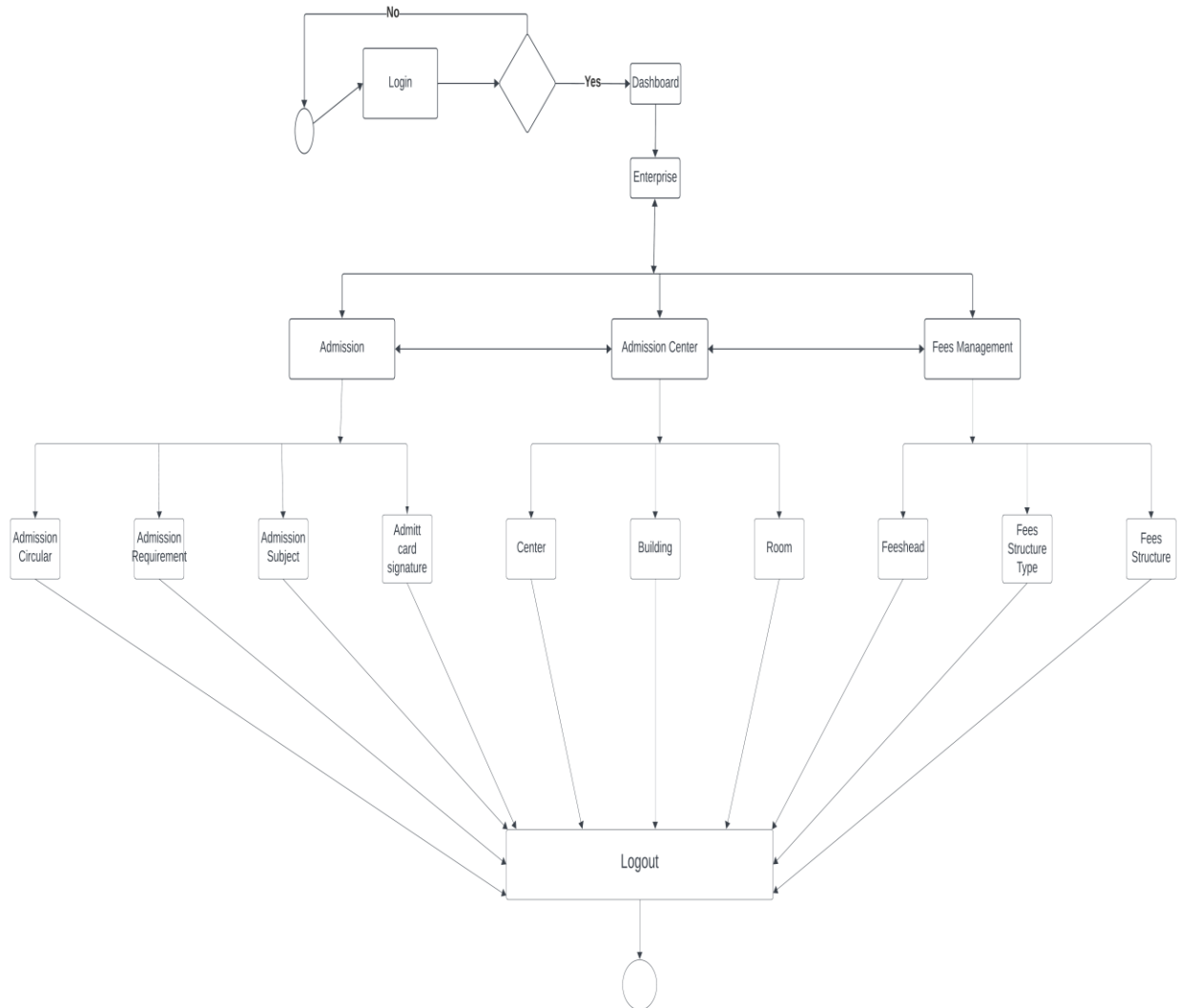


Figure 10: Activity diagram

8.2 Full System Use Case



Figure 11: Full system use case

8.2.1 Breakdown of Use Case diagram according to the functions of IUMS ADMISSION SERVICE

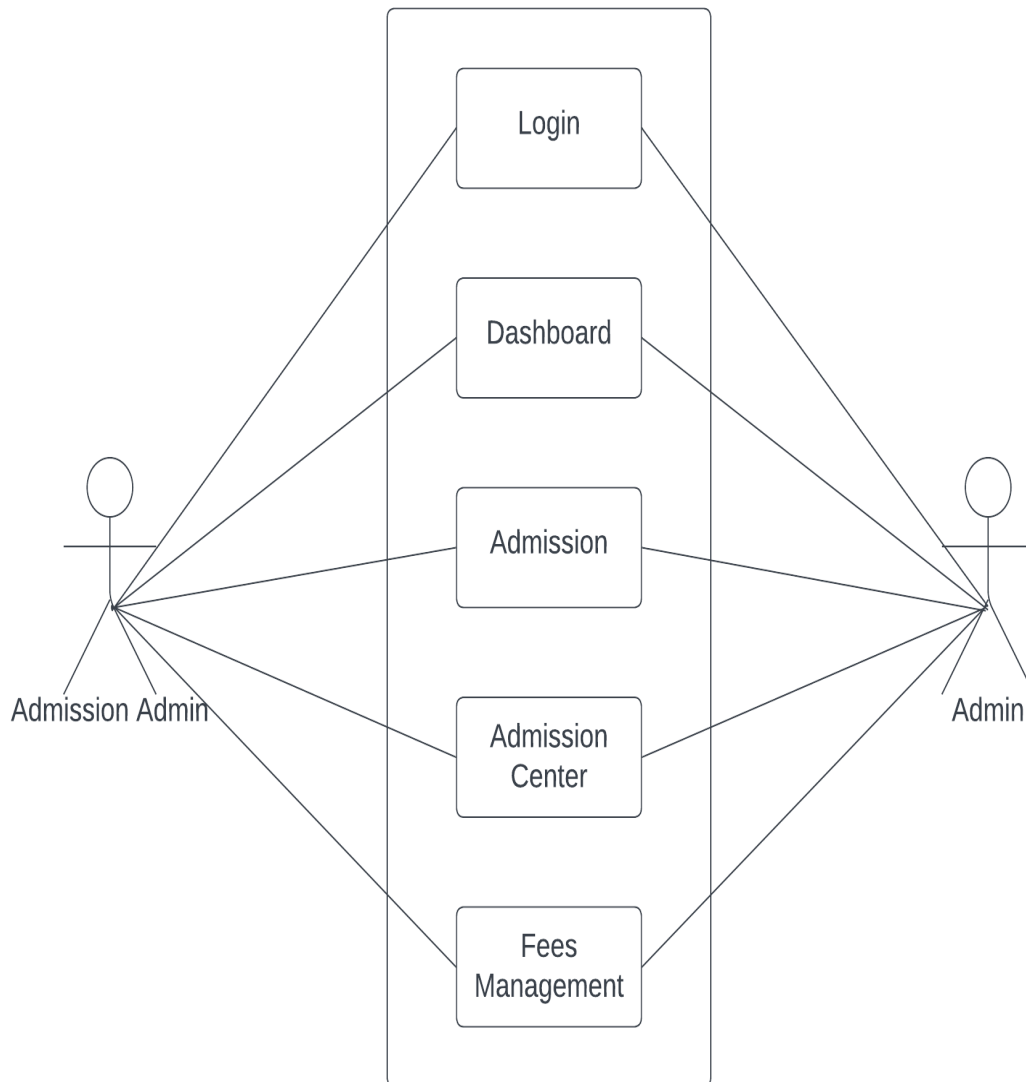


Figure 12: Use case for full system

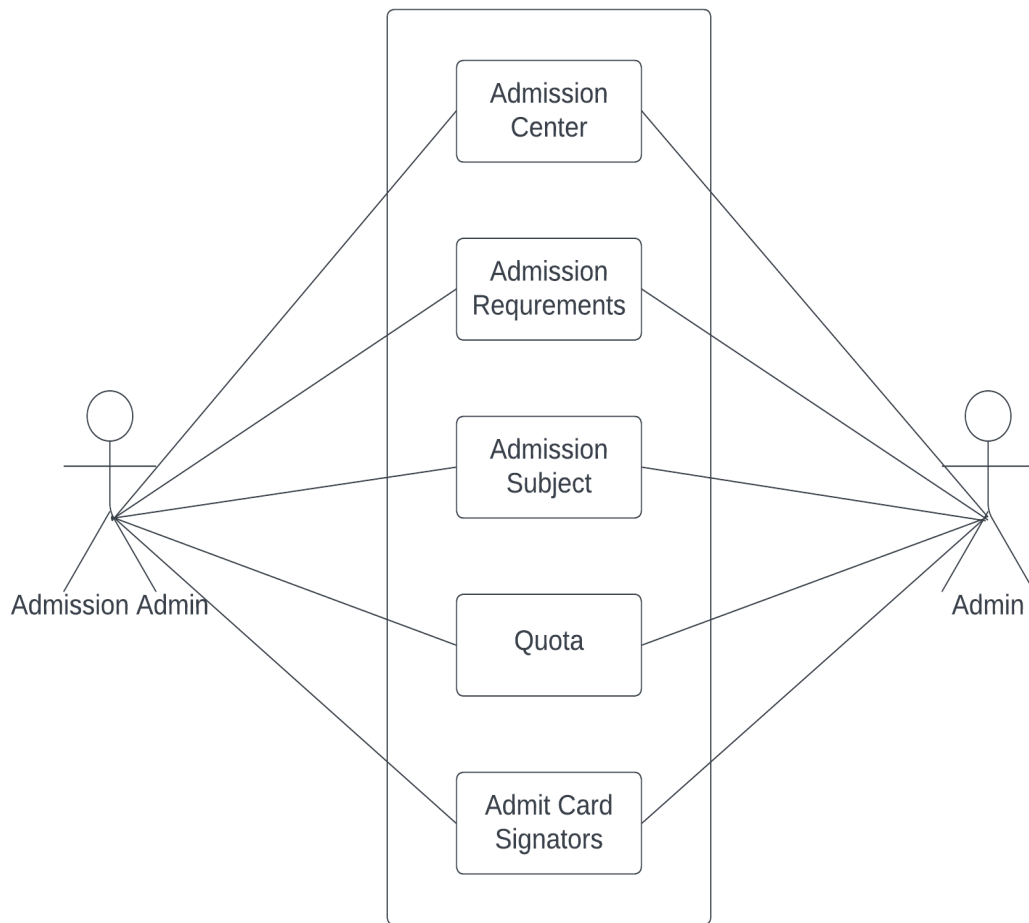


Figure 13: Use case for admission info

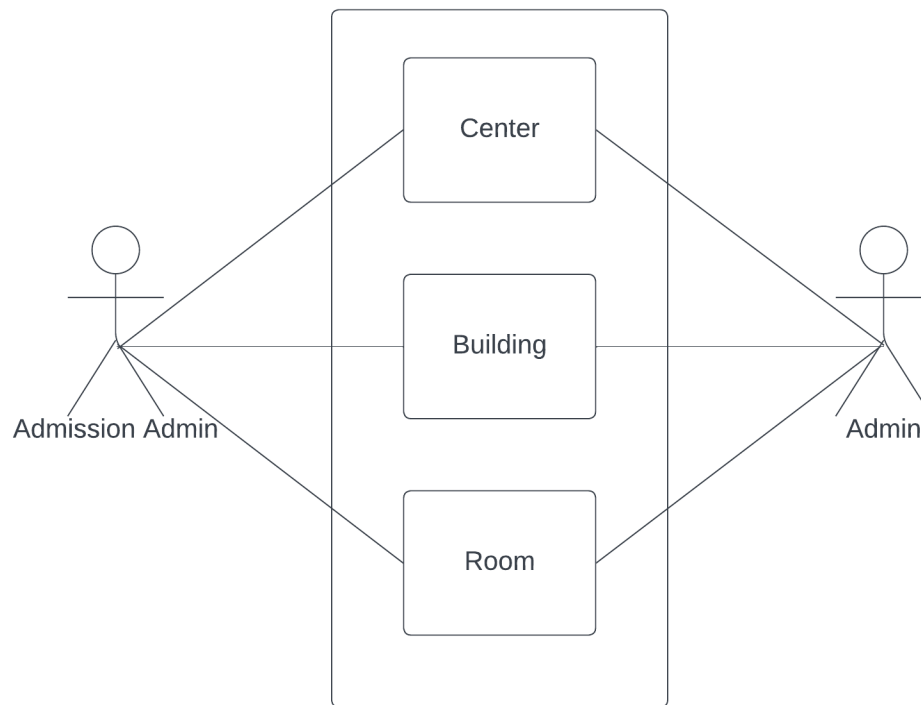


Figure 14: Use case for admission center

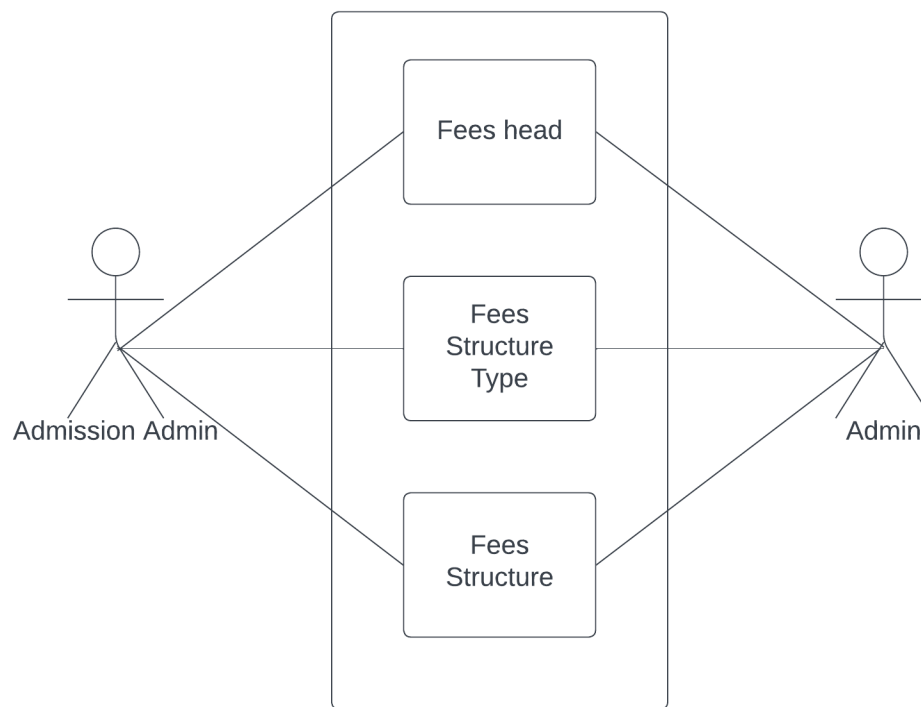


Figure 15: Use case for fees management

8.3 Full System Activity Diagram

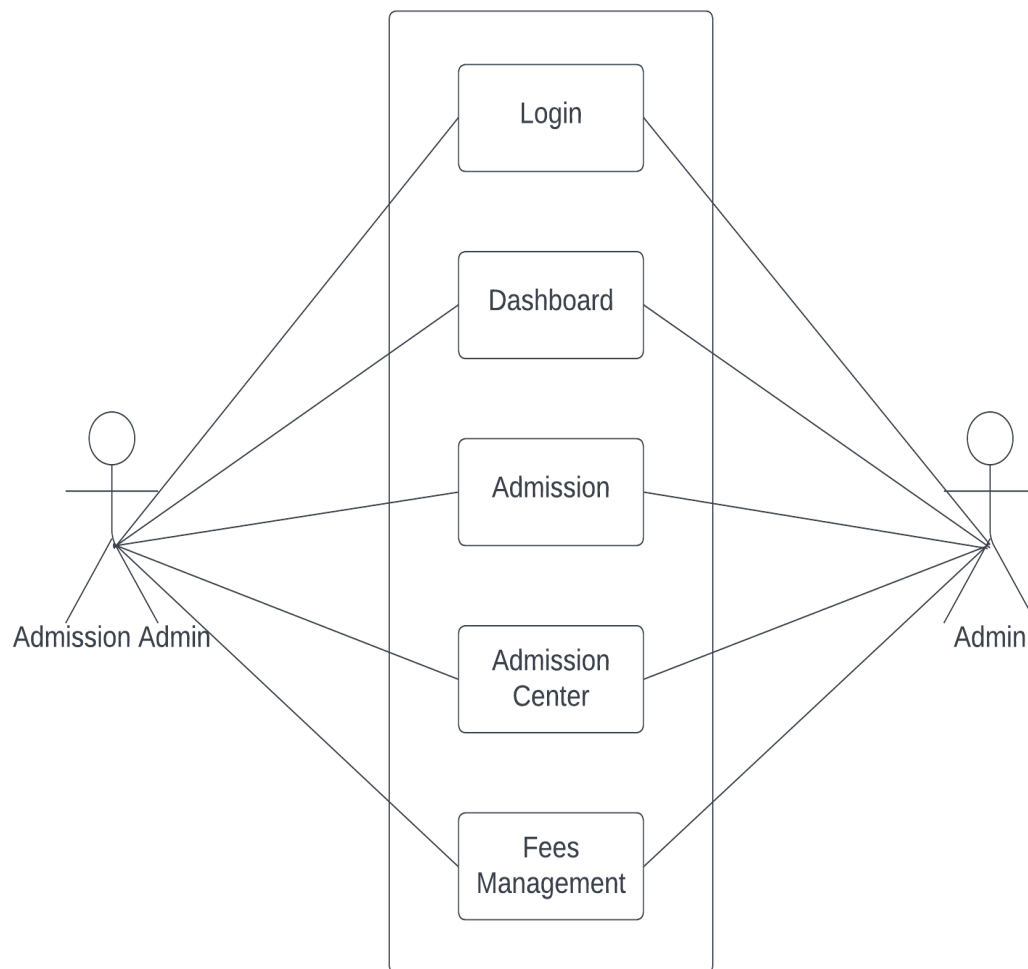


Figure 16: Full system activity diagram

8.4 Requirements Catalogue

To begin a project, it is necessary to identify the needs. These requirements should be maintained in a list called the requirement catalog. The requirements catalog's standard format will be followed for the addition of each found need to the list:

Requirement catalog for the login system

Requirement Type	Functional Requirement		
Requirement Name	Access system for administrators, admissions administrators, and applicants		
Requirement Description	The person who will be in charge of this system must register at the time the program is installed. After that, the administrator can log in to the system, create an admission admin, and the admission admin will have access to the system.		
Requirement ID	Date Source	Priority	Sign off
M-01	Admission Admin and Admin	Must	Admission Admin and Admin
Requirement Type	Non-Functional Requirement		
Requirement Name	The daily attempt to log in		
Description	Target Value	Acceptance Value	Comment
	2500	2200	Login is restricted to the admission admin and the admin.

Table 17: Requirement catalog for login system

Requirement catalog for the Admission Info

Requirement Type	Functional Requirement		
Requirement Name	Admission admin and admin will create admission information such as admission circular, admission requirements etc.		
Requirement Description	Admission portion contains admission circular, admission subject, admission requirement, admit card signatory etc. The creating opportunities of them must have for admission admin and admin		
Requirement ID	Date Source	Priority	Sign off
M-02	Admission Admin and Admin	Must	Admission Admin and Admin
Requirement Type	Non-Functional Requirement		
Requirement Name	The usability of these things for application		
Description	Target Value	Acceptance Value	Comment
	2500	2400	Admission Admin and Admin can add them to use

Table 18: Requirement catalog for the Admission Info

Requirement catalog for the Admission Center

Requirement Type	Functional Requirement		
Requirement Name	Admission admin and admin will setup admission center.		
Requirement Description	Admission center contains center, building and rooms. These things are used for setup centers for admission test. The creating opportunities of them must have for admission admin and admin.		
Requirement ID	Date Source	Priority	Sign off

M-02	Admission Admin and Admin	Must	Admission Admin and Admin
Requirement Type	Non-Functional Requirement		
Requirement Name	Number of centers created per year		
Description	Target Value	Acceptance Value	Comment
	500	450	Admission Admin and Admin can setup the center

Table 19: Requirement catalog for the Admission Center

Requirement catalog for the Fees Management

Requirement Type	Functional Requirement		
Requirement Name	Admission admin and admin will set the fees for several tasks of admission.		
Requirement Description	Fees management contains fees head, fees structure type and fees structure. By using this they can set the fees of admission related amount. The creating opportunities of them must have for admission admin and admin.		
Requirement ID	Date Source	Priority	Sign off
M-02	Admission Admin and Admin	Must	Admission Admin and Admin
Requirement Type	Non-Functional Requirement		

Requirement Name	Fees generation per year		
Description	Target Value	Acceptance Value	Comment
	500	500	Admission Admin and Admin can set the fees

Table 20: Requirement catalog for the Fees Management

8.5 Prioritized Requirement List (PRL)

The MoSCow priority technique is the best way to create a project's priority list. I will be employing this method to make a prioritized list of every criterion for the IUMS ADMISSION SERVICE that has been found. The list is as follows:

Priority Type	Must-Have
Serial No.	Requirement Name
1.	Role-based access control and secure user authentication.
2.	Flexible admissions requirements that are managed dynamically.
3.	Easy-to-use application submission and tracking of status.
4.	System for managing fees that is adaptable and based on programs or categories.
5.	An integrated communication system that facilitates effective teamwork and automated notifications.

Table 21: Must have priorities

Priority Type	Should-Have
Serial No.	Requirement Name
1.	Strong data encryption to protect the privacy of applicants.
2.	Features that are accessible to users with different needs.
3.	For data integrity, regular automatic backups are necessary.
4.	Scalability to manage different numbers of admissions.

5.	Compliance to privacy laws and data protection guidelines.
----	--

Table 22: Should have priorities

Priority Type	Could-Have
Serial No.	Requirement Name
1.	Integration for expedited verification with external academic databases.
2.	Gamified elements to improve the engagement of applicants.
3.	Chatbot powered by AI for immediate applicant assistance.
4.	Blockchain technology for safe verification of documents.
5.	Resource planning and admission patterns using predictive analytics.

Table 23: Could have priorities

8.6 Prototype of the new system

Admission Circular List

+ New Search

Name	Application start Date	Application start Time	Exam Date	Exam Time	Application end Date	Application end Time	Status	Action
« Previous 1 Next »								

Figure 17: Admission circular list view

ADMISSION CIRCULAR CREATE

Name
Enter name

Exam Date
mm/dd/yyyy

Exam Time
--:--:--

Application Start Date
mm/dd/yyyy

Application Start Time
--:--:--

Application Closing Date
mm/dd/yyyy

Application Closing Time
--:--:--

Payment Closing Date
mm/dd/yyyy

Payment Closing Time
--:--:--

SSC Passing Year
Select year

HSC Passing Year
Select year

Status
Select status

Instruction

Figure 18: Admission circular create

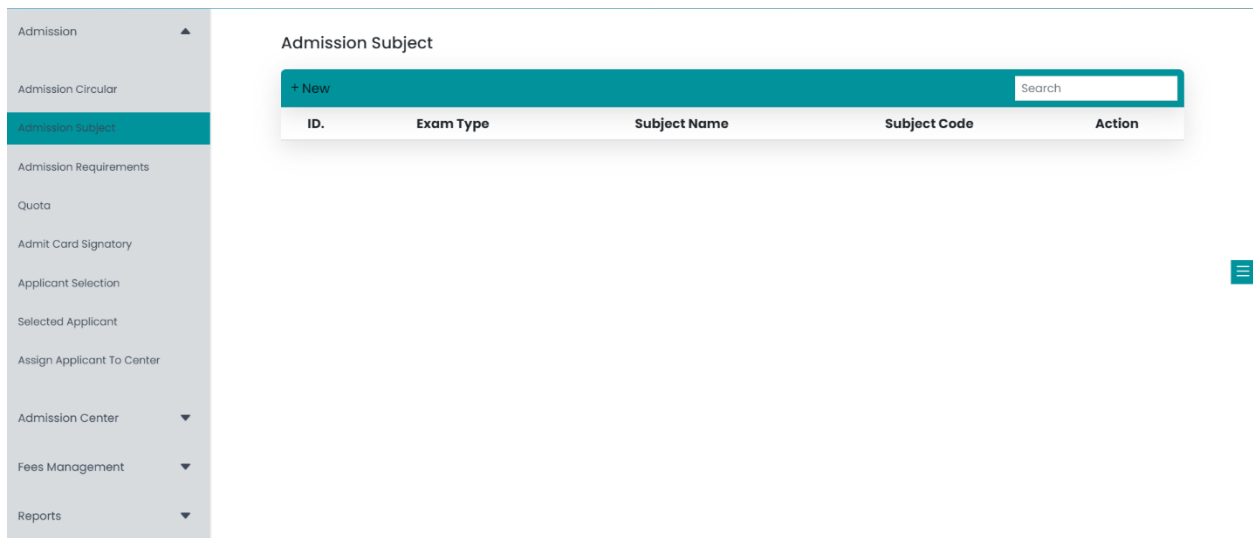


Figure 19: Admission subject list view

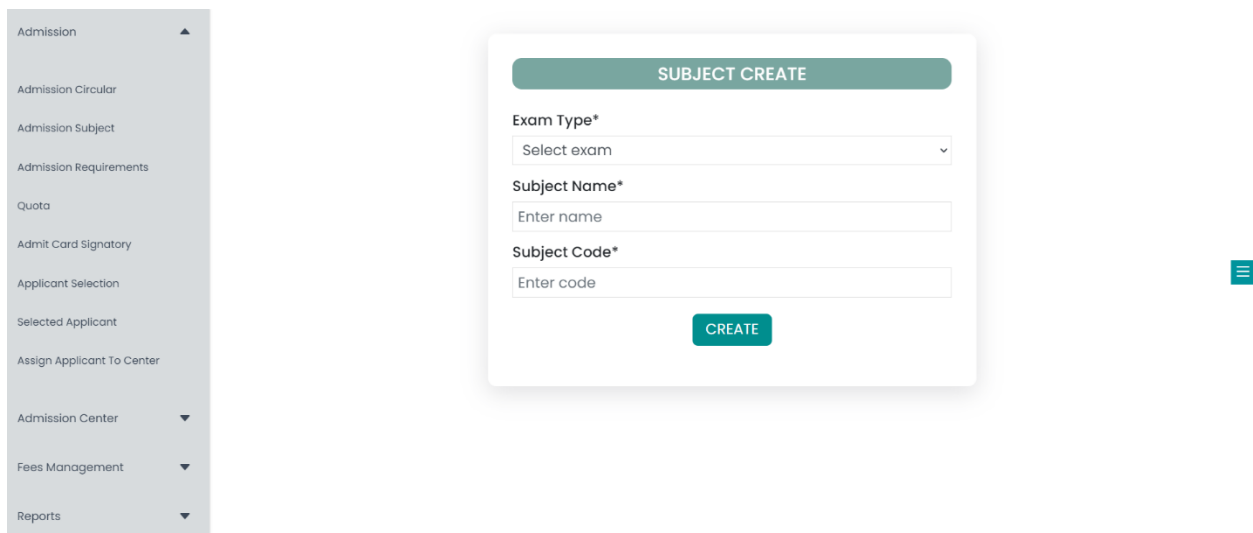


Figure 20: Admission subject create

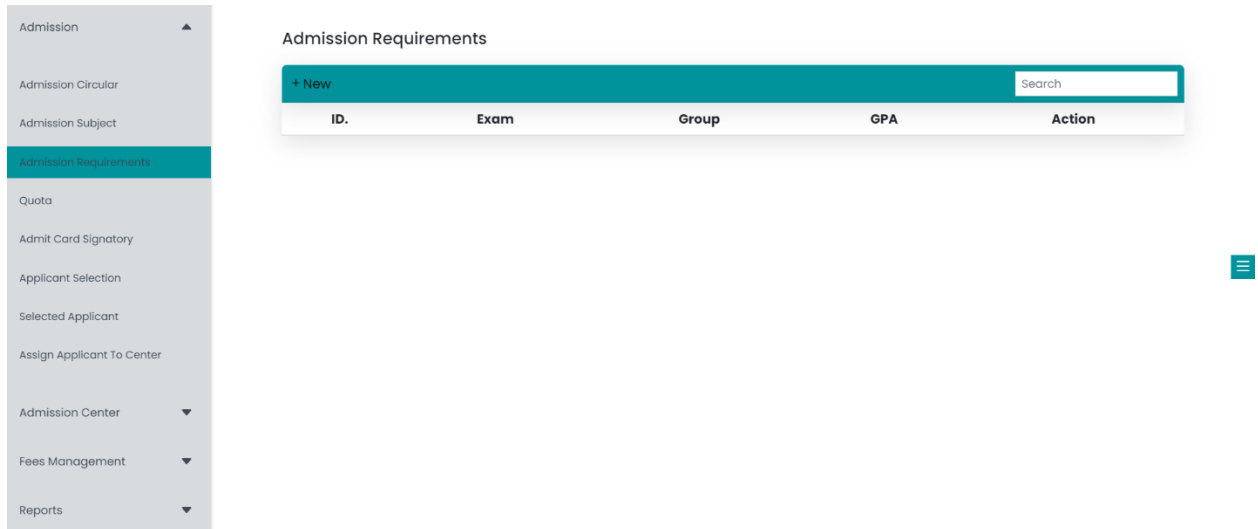


Figure 21: Admission requirement list view

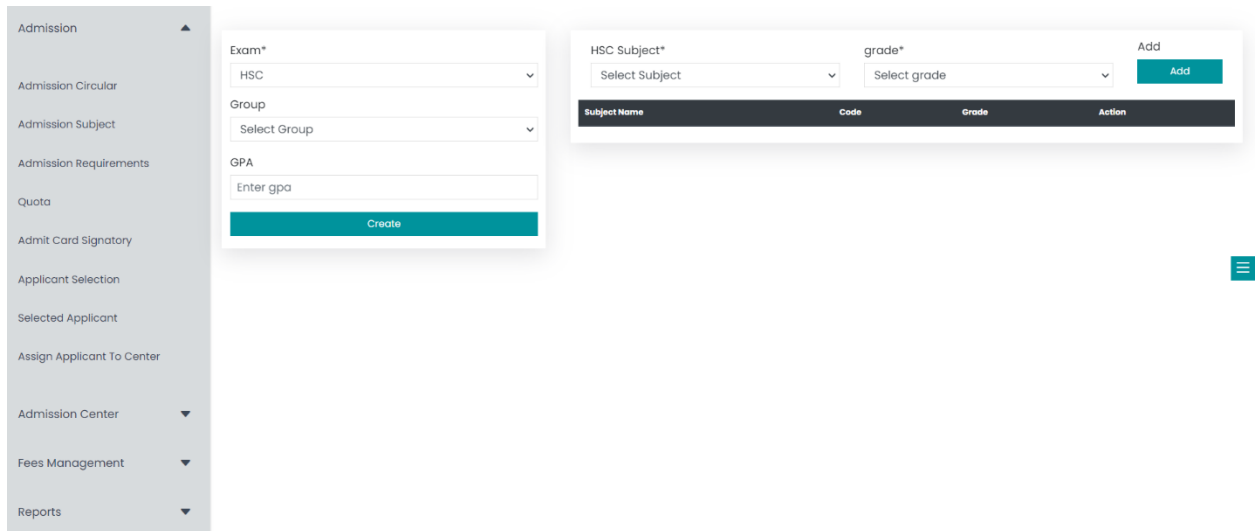


Figure 22: Admission requirement create

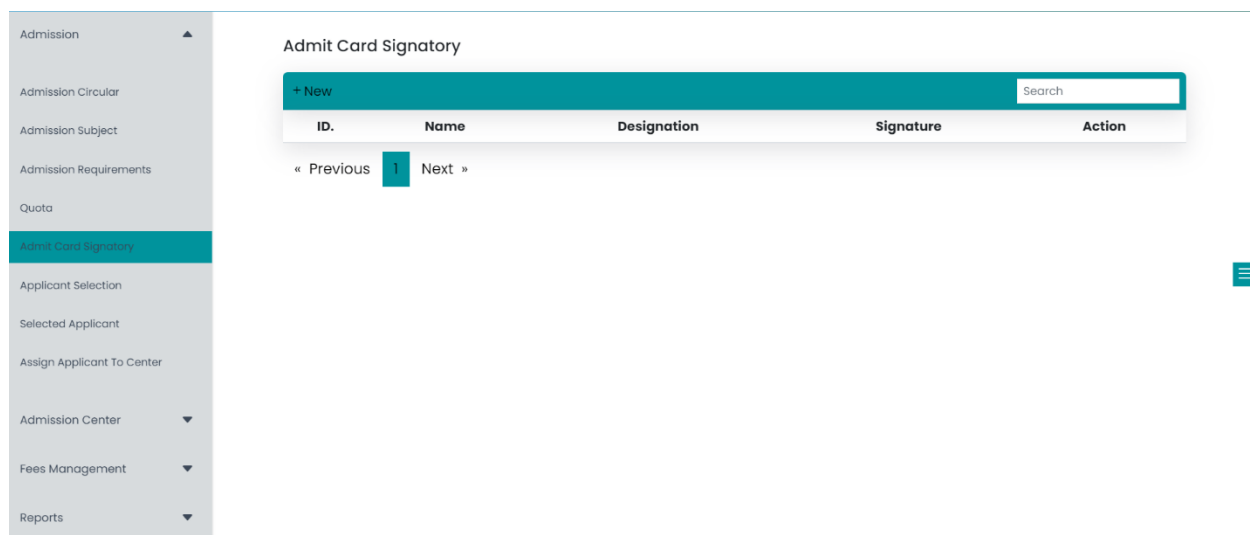


Figure 24: Admit card signatory list view

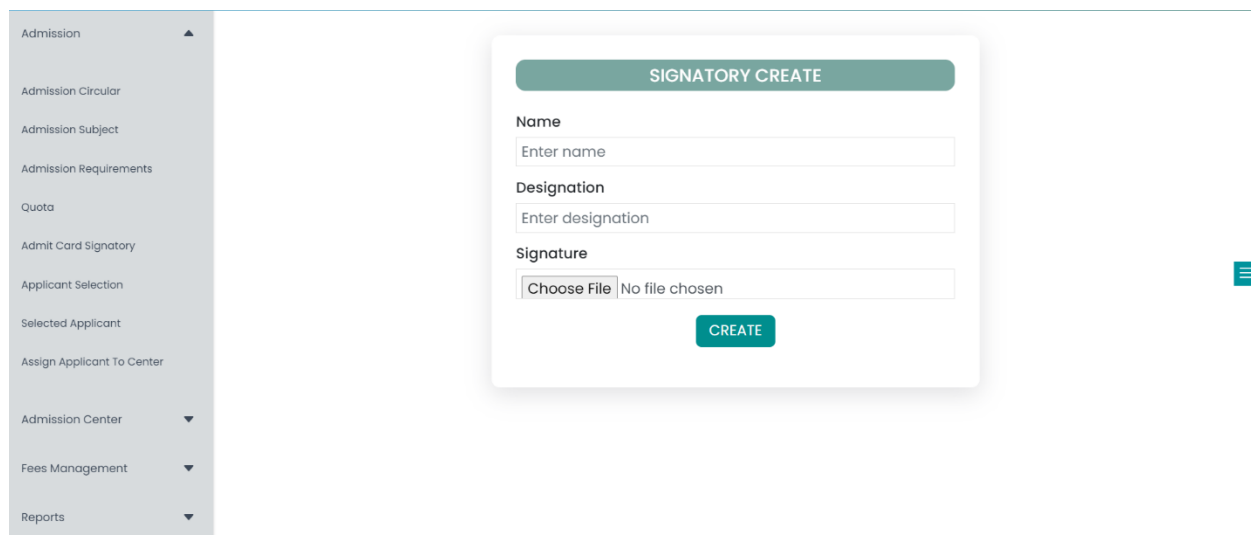


Figure 23: Admit card signatory create

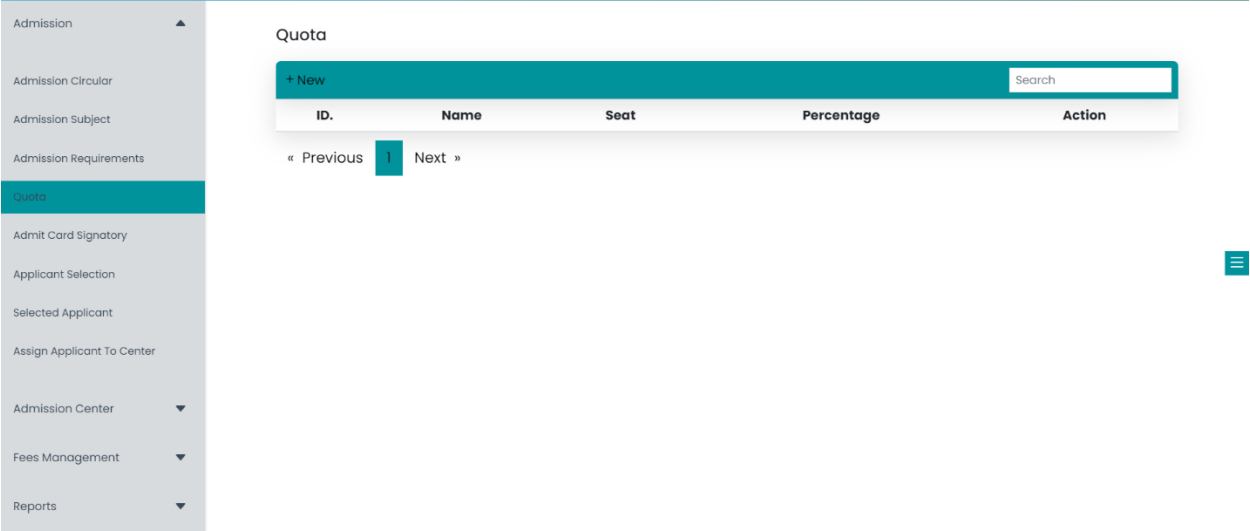


Figure 25: Quota list view

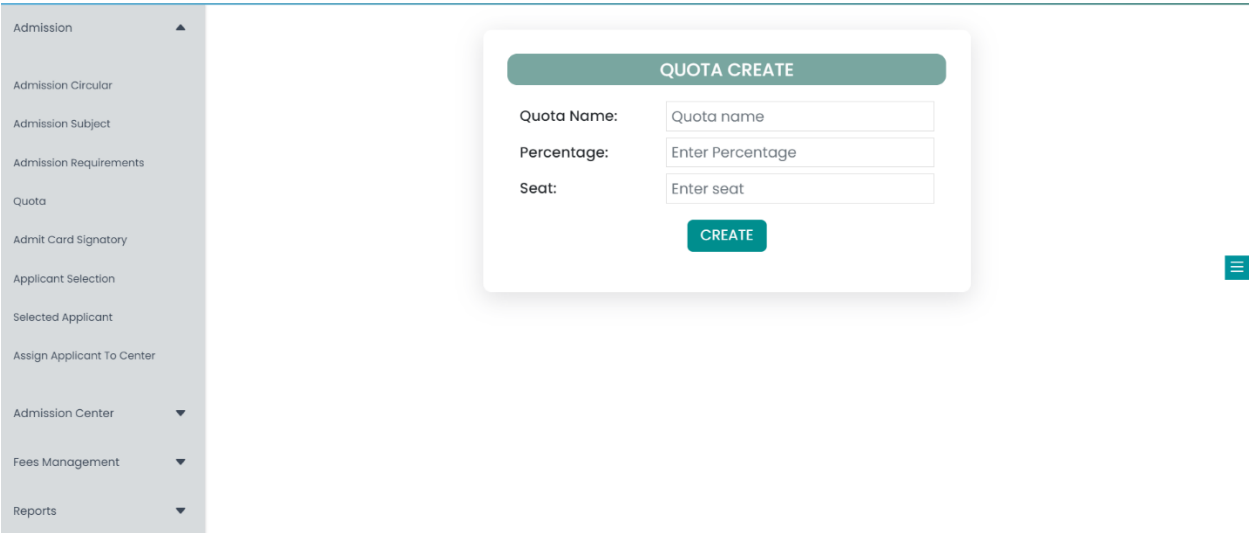


Figure 26: Quota create

Chapter 9 – Engineering

9.1 Modules of the new system

Since the IUMS ADMISSION SERVICE is a comprehensive system for university admission service, it is brimming with new modules. There will therefore be a ton of modules to cover. I'll demonstrate and go over a few of the key modules here:

Note: I'll demonstrate how the system interacts with itself if a user takes a certain action in this section.

Module for the login system

Serial of Action	The action of User	The interaction of System
1.	Once the admin clicks the sign in button, they can browse the sign in URL.	The system will direct the user to the login page and show a login form if the admin hasn't logged in yet; if not, it will direct the user to the dashboard.
2.	As soon as the login form is displayed, the admin will provide the necessary credentials.	After getting the user's credentials, the system will determine whether or not the user submitted data in a proper format.
3.	It's time to sign in now. The login or sign-in button will be clicked by the admin.	The system will compare the credentials with the database's contents once it has received the user's credentials. If the credentials are correct, the user will either be directed to the dashboard or get an error warning.

Table 24: Module for the login system

Module for the Admission Configuration

Serial of Action	The action of User	The interaction of System
1.	The admission option will be clicked by the admin and admission admin.	There will be another choice for admission configuration once the system opens the admission option.
2.	The function will be chosen by the administrator in accordance with his needs.	The system will route the user to a specific page in response to the request.
3.	Admin requests may be made to add, amend, or remove information based on need.	The user's instructions will be followed by the system, and it will react in line with their desire.

Table 25: Module for the Admission Configuration

Module for the Admission Center Setup

Serial of Action	The action of User	The interaction of System
1.	The selection for the admission center will be clicked by the admin and admission admin.	There will be another option for setting up an entrance center when the system opens the admission center option.
2.	The function will be chosen by the administrator in accordance with his needs.	The system will route the user to a specific page in response to the request.
3.	Admin requests may be made to add, amend, or remove information based on need.	The user's instructions will be followed by the system, and it will react in line with their desire.

Table 26: Module for the Admission Center Setup

Module for the Fees Management Setup

Serial of Action	The action of User	The interaction of System
1.	The option to control fees will be clicked by the admissions admin and admin.	The system will launch the option for managing fees, and under the area pertaining to entry costs, there will be an additional choice.
2.	The function will be chosen by the administrator in accordance with his needs.	The system will route the user to a specific page in response to the request.
3.	Admin requests may be made to add, amend, or remove information based on need.	The user's instructions will be followed by the system, and it will react in line with their desire.

Table 27: Module for the Fees Management Setup

9.2 Use Case for IUMS ADMISSION SERVICE

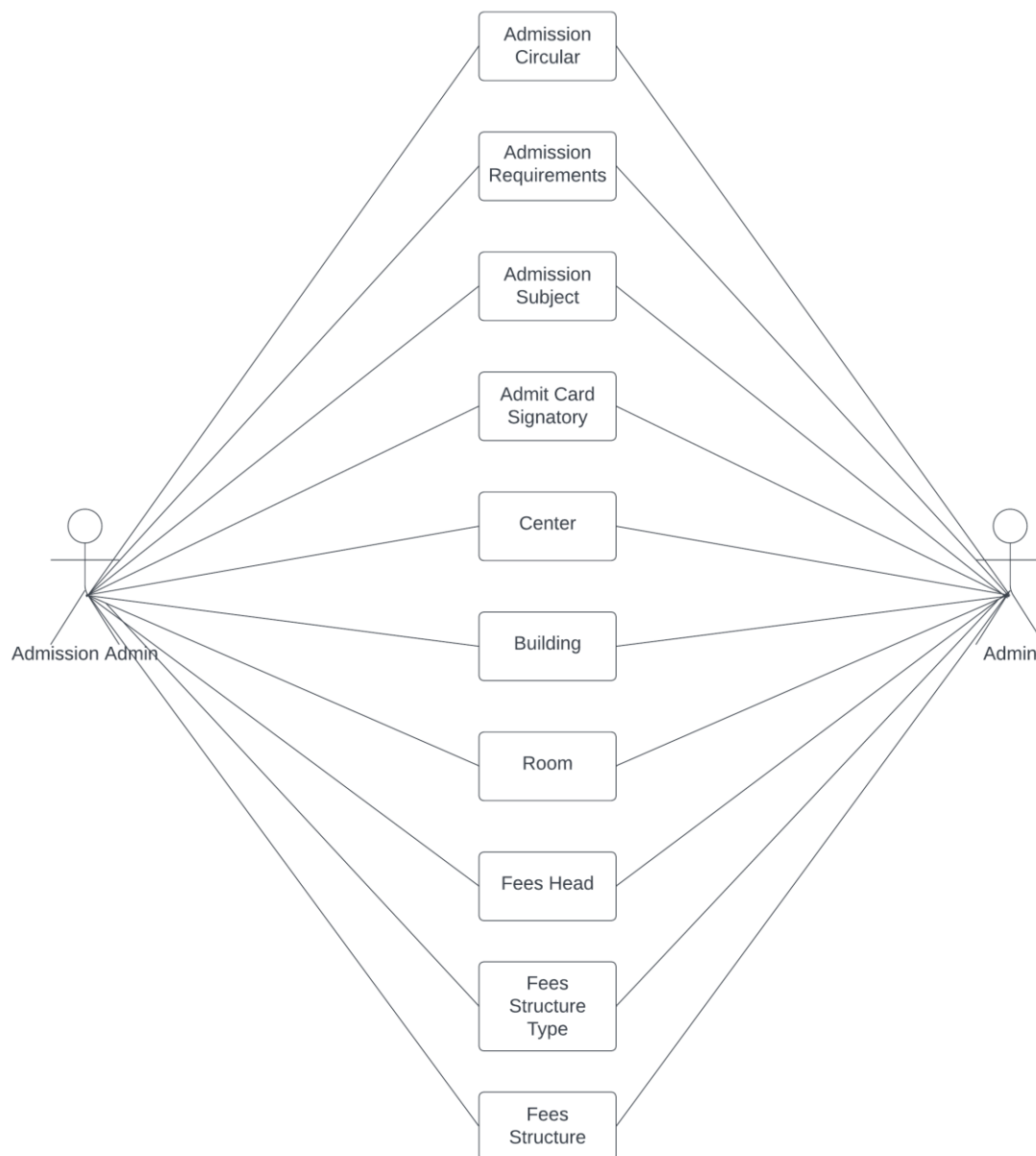


Figure 27: Use Case for IUMS ADMISSION SERVICE

9.3 Sequence diagram for IUMS ADMISSION SERVICE

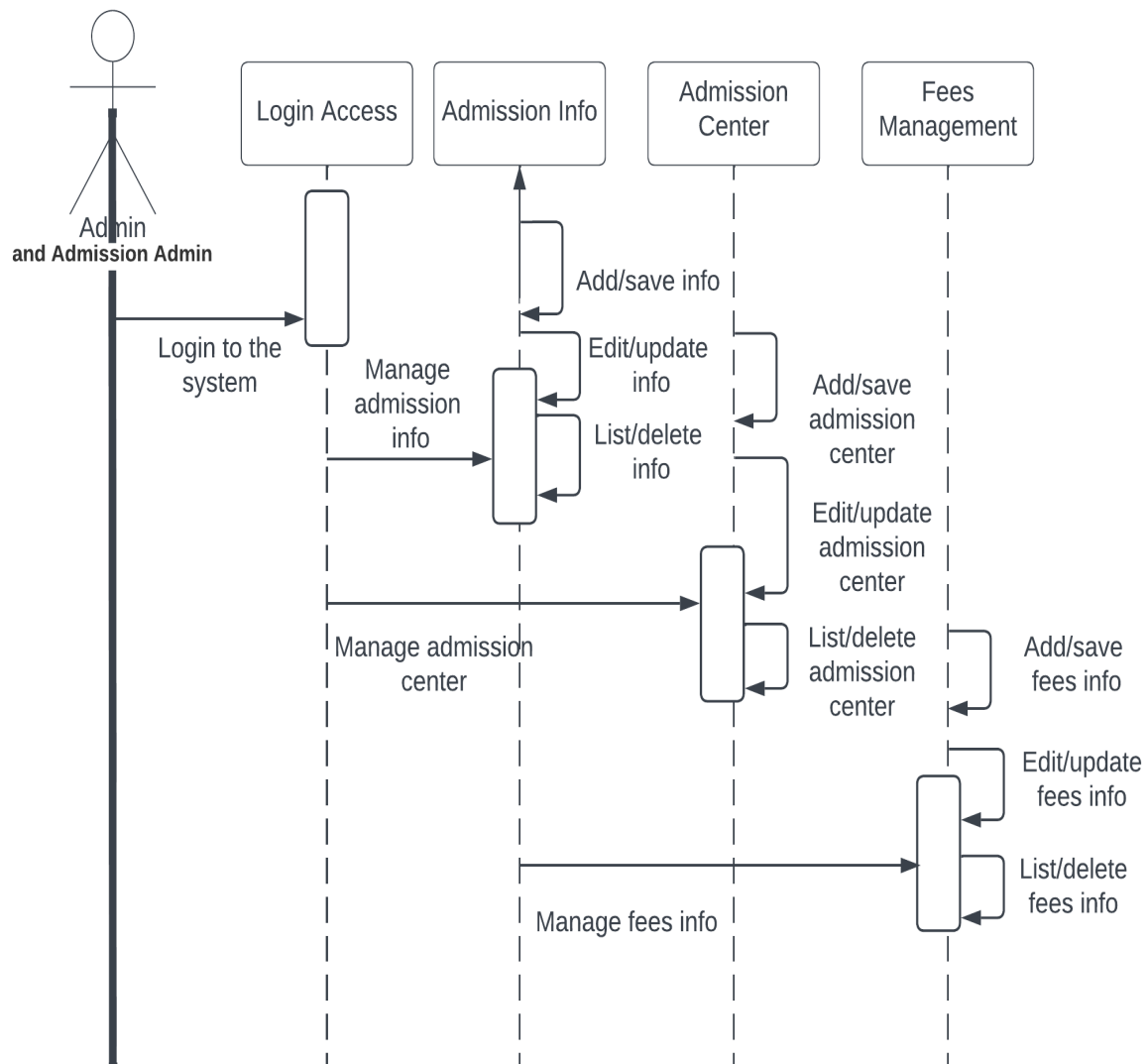


Figure 28: Sequence diagram for IUMS ADMISSION SERVICE

9.4 Component diagram for IUMS ADMISSION SERVICE

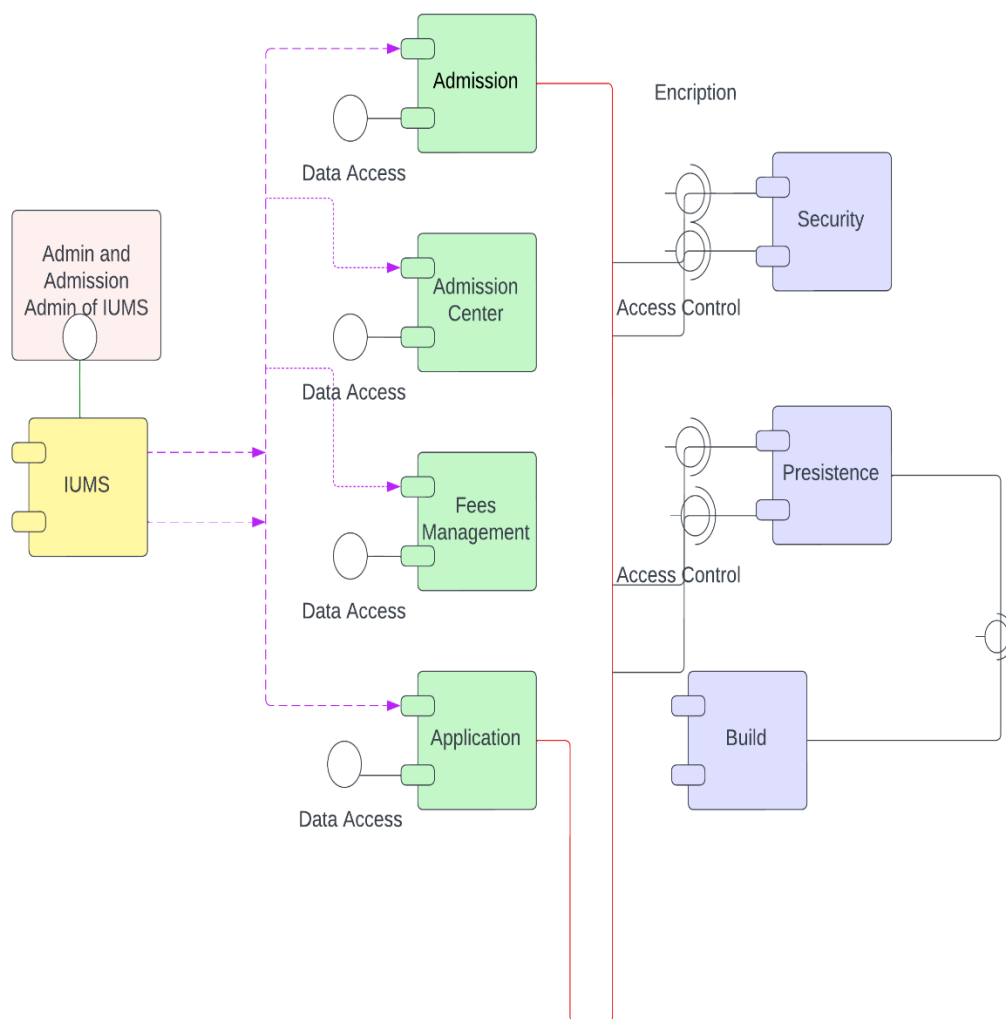


Figure 29: Component diagram for IUMS ADMISSION SERVICE

9.5 Deployment diagram for IUMS ADMISSION SERVICE

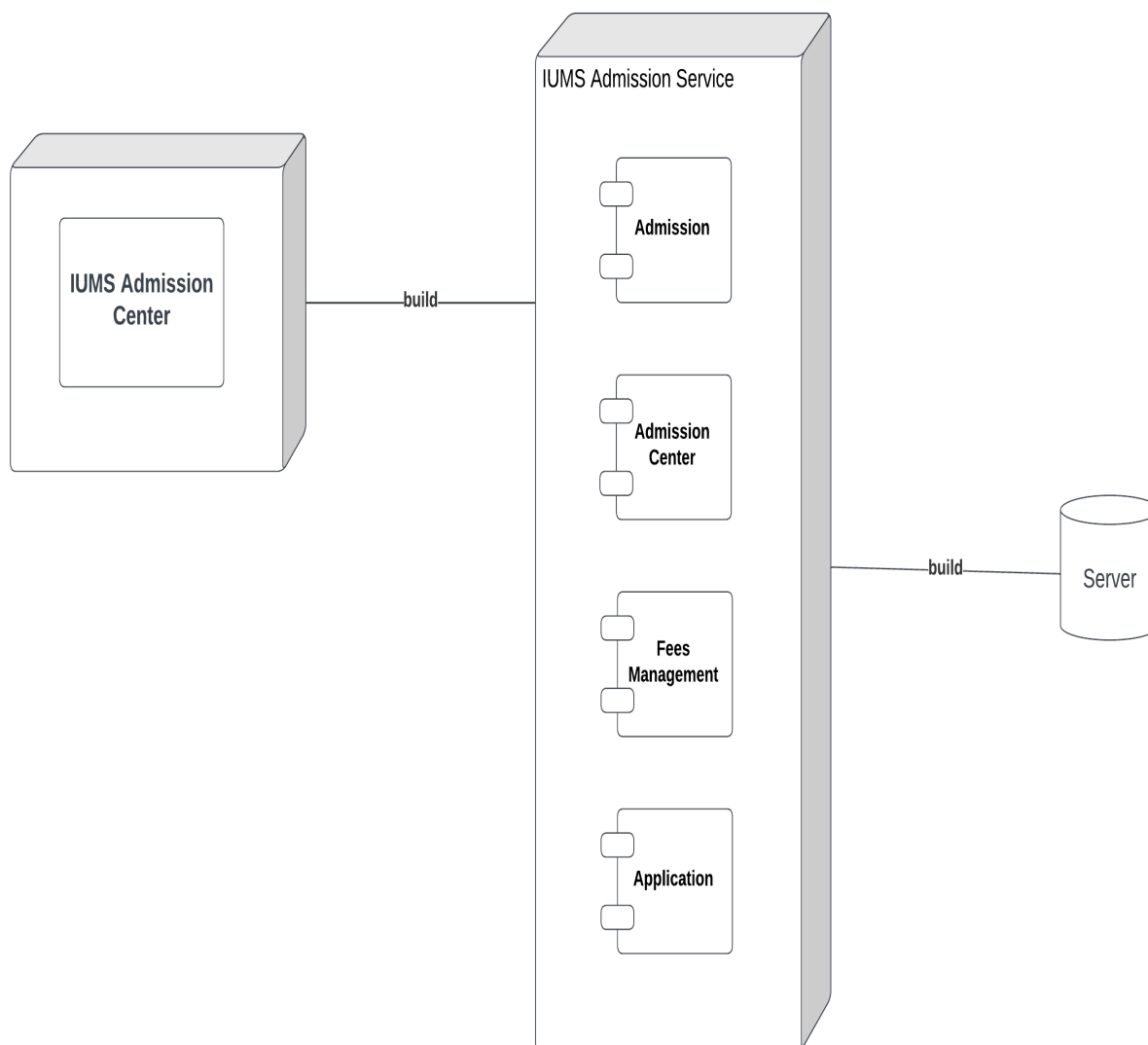


Figure 30: Deployment diagram for IUMS ADMISSION SERVICE

9.6 System Interface Design / Prototype

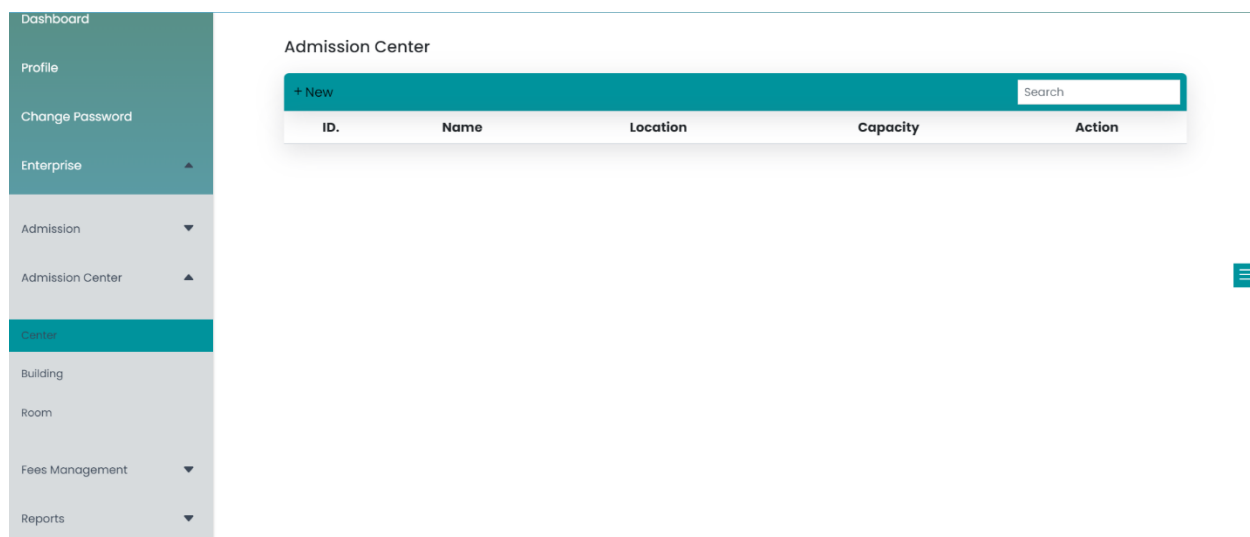


Figure 32: Admission center list view

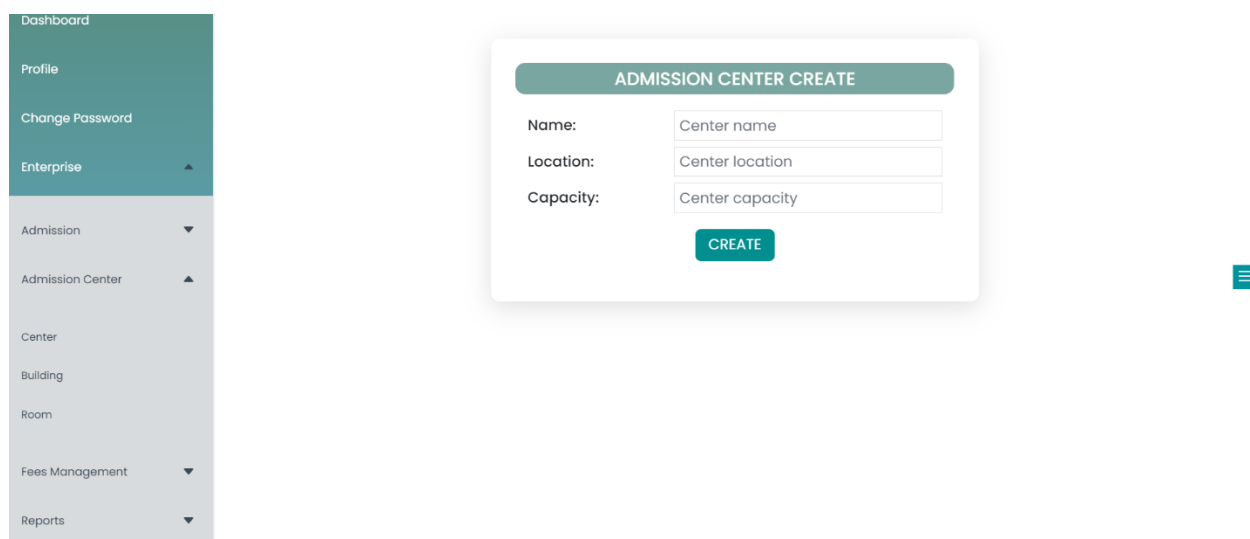


Figure 31: Admission center create view

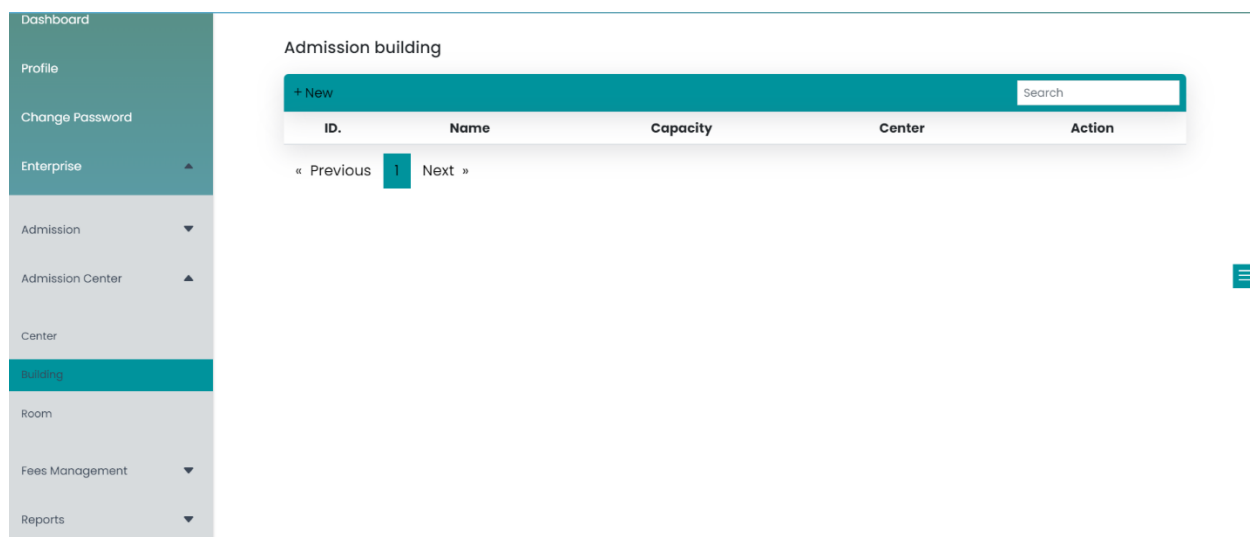


Figure 34: Admission building list view

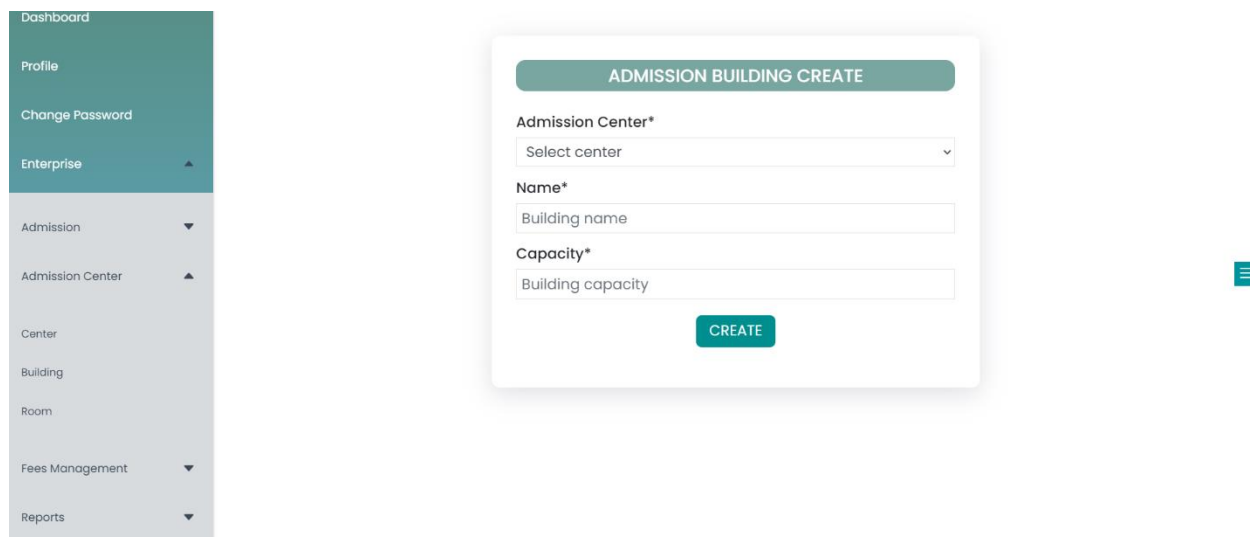


Figure 33: Admission building create

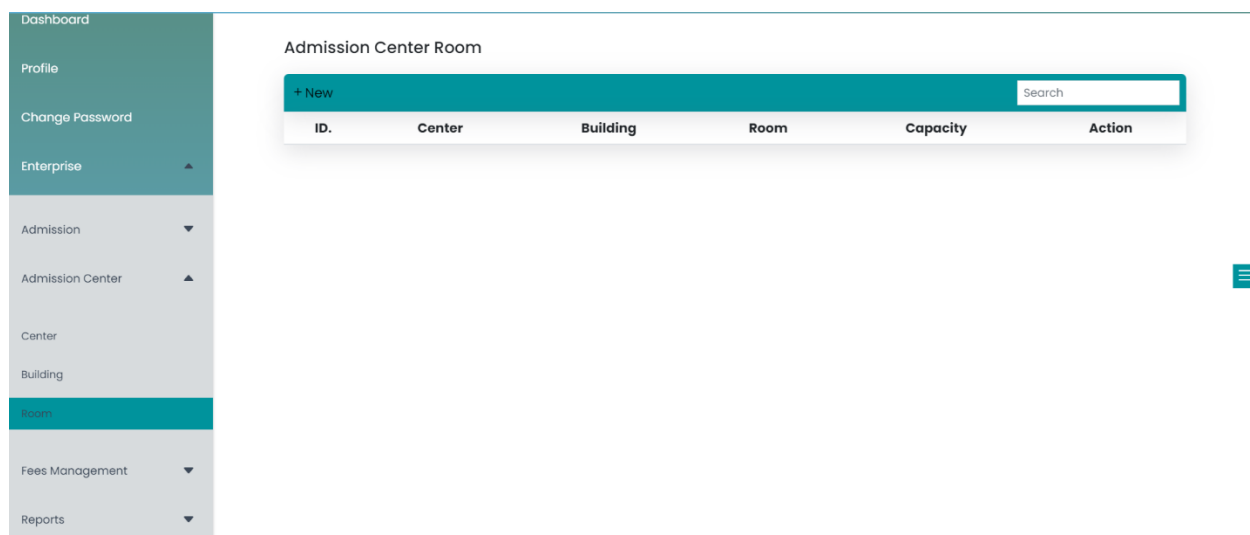


Figure 35: Admission center room list view

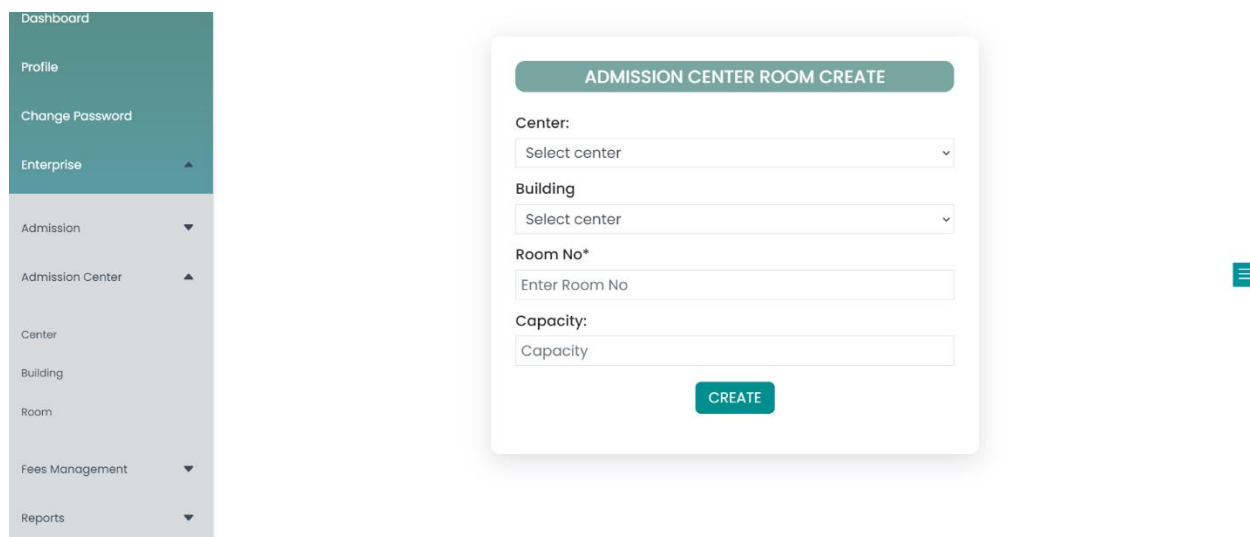


Figure 36: Admission center room create

Chapter 10 – Deployment/Development

10.1 Core Module Coding Sample

```
<div class="showList rtm__padding-top-bottom">
  <div class="container">
    <h2 class="showList__title">Admission Center</h2>

    <div class="rtm__table__container">
      <div class="rtm__table__topbar">
        <button routerLink="/dashboard/admin/admission-center-create">
          <i class="bi bi-plus"></i>
          <span>New</span>
        </button>
        <input
          type="text"
          name=""
          id=""
          class="form-control rtm__input"
          placeholder="Search"
          aria-describedby="helpId"
        />
      </div>
      <table class="table rtm__table">
        <thead class="rtm__table__thead">
          <tr>
            <th scope="col">ID.</th>
            <th scope="col">Name</th>
            <th scope="col">Location</th>
            <th scope="col">Capacity</th>
            <th scope="col">Action</th>
          </tr>
        </thead>
        <tbody>
          <tr *ngFor="let center of admissionCenters">
            <th scope="row">{{ center.id }}</th>
            <td>{{ center.name }}</td>
            <td>{{ center.location }}</td>
            <td>{{ center.capacity }}</td>
            <td class="action">
              <button
```

Figure 37: Admission center code sample - 1

```

@import "../../../styles/abstracts/mixins";
@import "../../../styles/base/variables";

.showList{
  width: 100%;
  min-height: 100vh;

  .rtm__table__topbar {
    display: flex;
    justify-content: space-between;
    button {
      border: none;
      background-color: transparent;
      font-size: 16px;
      i {
        font-size: 18px;
      }
    }
    input{
      width: 200px;
      font-size: 14px;
    }
  }

  .my-pagination ::ng-deep .ngx-pagination {
    li {
      font-size: 18px;
      padding: 8px 14px;
      color: $admin-text-secondary-clr;
      @include break(small) {
        font-size: 12px;
        padding: 4px 4px;
      }
    }
    li.current {
      background: $admin-primary-clr;
      color: $admin-text-secondary-clr;
    }
  }
}

```

Figure 38: Admission center code sample - 2

```

import { Component, OnInit } from "@angular/core";
import { ToastrService } from "ngx-toastr";
import { AdmissionService } from "../../../../../services/admission.service";
import { SweetAlertService } from "../../../../../core/services/sweet-alert.service";
@Component({
  selector: "app-admission-center",
  templateUrl: "./admission-center.component.html",
  styleUrls: ["./admission-center.component.scss"],
})
export class AdmissionCenterComponent implements OnInit {
  admissionCenters = [];

  constructor(
    private admissionService: AdmissionService,
    private toast: ToastrService,
    private swtalt : SweetAlertService
  ) {}

  ngOnInit(): void {
    this.getAllCenter();
  }

  getAllCenter(): void {
    this.admissionService
      .getAllAdmissionCenter()
      .subscribe((response: any) => {
        if (response.errors != undefined) {
        } else {
          this.admissionCenters = response.payload;
        }
      });
  }

  async deleteCenter(id: string) {
    let res = await this.swtalt.confirm("Are you sure?", "", "warning");
    if (res) {
      this.admissionService
    }
  }
}

```

Figure 39: Admission center code sample – 3

```

<div class="circular-create rtm_padding-top-bottom">
  <form class="rtm_form" (ngSubmit)="createNewCircular()">
    <h2 class="rtm_form_title">Admission Circular Create</h2>

    <div class="form-group">
      <label for="name" class="col-form-label"> Name </label>
      <input
        type="text"
        class="form-control rtm_input"
        id="name"
        name="name "
        placeholder="Enter name"
        [(ngModel)]="circular.name"
      />
    </div>

    <div class="row">
      <div class="col-md-6">
        <div class="form-group">
          <label for="examDate" class="col-form-label">
            >Exam Date
          </label>
          <input
            type="date"
            class="form-control rtm_input"
            id="examDate"
            name="examDate "
            [(ngModel)]="circular.examDate"
          />
        </div>
      </div>
      <div class="col-md-6">
        <div class="form-group">
          <label for="examTime" class="col-form-label">
            >Exam Time
          </label>
          <input

```

Figure 40: Admission circular create code sample - 1

```

<div class="col-md-6">
  <div class="form-group">
    <label for="examTime" class="col-form-label">
      Exam Time
    </label>
    <input
      type="time"
      class="form-control rtm__input"
      id="examTime"
      name="examTime "
      step="1"
      [(ngModel)]="circular.examTime"
    />
  </div>
</div>

<div class="col-md-6">
  <div class="form-group">
    <label for="applicationStartDate" class="col-form-label">
      Application Start Date
    </label>
    <input
      type="date"
      class="form-control rtm__input"
      id="applicationStartDate"
      name="applicationStartDate "
      [(ngModel)]="circular.applicationStartDate"
    />
  </div>
</div>
<div class="col-md-6">
  <div class="form-group">
    <label for="applicationStartTime" class="col-form-label">
      Application Start Time
    </label>
    .
  </div>
</div>

```

Figure 41: Admission circular create code sample - 2

```

@import "../../../styles/abstracts/mixins";
.circular-create {
  width: 100%;
  display: flex;
  justify-content: center;
  .rtm__form {
    width: 70% !important;
    @include break(small) {
      width: 100% !important;
      margin: 0 6px;
    }
  }
}

.editor {
  border: 2px solid rgba(0, 0, 0, 0.2);
  border-radius: 4px;

  .NgxEditor__MenuBar {
    border-top-left-radius: 4px;
    border-top-right-radius: 4px;
    border-bottom: 1px solid rgba(0, 0, 0, 0.2);
    font-size: 16px;
  }

  .NgxEditor {
    border-top-left-radius: 0;
    border-top-right-radius: 0;
    border: none;
    height: 300px;
    overflow-y: scroll;
    padding-left: 20px;
  }

  .NgxEditor__Content {
    font-size: 16px;
  }
}

```

Figure 42: Admission circular create code sample - 3

```

import { Component, OnInit, ViewEncapsulation } from "@angular/core";
import { ToastrService } from "ngx-toastr";
import { Editor, Toolbar } from "ngx-editor";

import { AdmissionService } from "../../../../../services/admission.service";
import { ErrorService } from "../../../../../services/error.service";
import { SweetAlertService } from "../../../../../core/services/sweet-alert.service";
@Component({
  selector: "app-admission-circular-create",
  templateUrl: "./admission-circular-create.component.html",
  styleUrls: ["./admission-circular-create.component.scss"],
  encapsulation: ViewEncapsulation.None,
})
export class AdmissionCircularCreateComponent implements OnInit {
  editor: Editor;
  toolbar: Toolbar = [
    ["bold", "italic"],
    ["underline", "strike"],
    ["code", "blockquote"],
    ["ordered_list", "bullet_list"],
    [{ heading: ["h1", "h2", "h3", "h4", "h5", "h6"] }],
    ["link", "image"],
    ["text_color", "background_color"],
    ["align_left", "align_center", "align_right", "align_justify"],
  ];

  years: number[] = [];
  circular = {
    name: "",
    examDate: "",
    examTime: "",
    applicationStartDate: "",
    applicationStartTime: "",
    applicationClosingDate: "",
    applicationClosingTime: "",
    paymentClosingDate: "",
    paymentClosingTime: "",
  };

```

Figure 43: Admission circular create code sample - 4

```

        hscPassingYear: "",
        status: "",
        instruction: "",
    };
    constructor(
        private toastService: ToastrService,
        private admissionService: AdmissionService,
        private errorService: ErrorService,
        private sweetAlertService: SweetAlertService
    ) {}

    ngOnInit(): void {
        this.editor = new Editor();

        let currentYear: number = getFullYear();
        for (let i = currentYear - 7; i < currentYear + 7; i++) {
            this.years.push(i);
        }
    }

    ngOnDestroy(): void {
        this.editor.destroy();
    }

    async createNewCircular(): Promise<void> {
        let confirm = await this.sweetAlertService.confirm(
            "Are you sure?",
            "",
            "question"
        );
        if (confirm) {
            this.admissionService.createNewCircular(this.circular).subscribe(
                (response) => {
                    this.toastService.success(response.payload);
                    this.circular = {
                        name: "",
                        ...

```

Figure 44: Admission circular create code sample – 5

```

<div class="showList rtm_padding-top-bottom">
  <div class="container-fluid">
    <h2 class="showList__title">Admission Circular List</h2>

    <div class="rtm_table_container">
      <div class="rtm_table_topbar">
        <button routerLink="/dashboard/admin/admission/circular/new">
          <i class="bi bi-plus"></i>
          <span>New</span>
        </button>
        <input
          type="text"
          name=""
          id=""
          class="form-control rtm_input"
          placeholder="Search"
          (keyup)="applyFilter($event.target.value)"
        />
      </div>
      <table class="table rtm_table">
        <thead class="rtm_table__thead">
          <tr>
            <th scope="col">Name</th>
            <th scope="col">Application start Date</th>
            <th scope="col">Application start Time</th>
            <th scope="col">Exam Date</th>
            <th scope="col">Exam Time</th>
            <th scope="col">Application end Date</th>
            <th scope="col">Application end Time</th>
            <th scope="col">Status</th>
            <th scope="col">Action</th>
          </tr>
        </thead>
        <tbody>
          <tr
            *ngFor="
              let circular of circularList

```

Figure 45: Admission circular list code sample – 1

```

| paginate
| : { itemsPerPage: 15, currentPage: page }
"
>
<th scope="row">{{ circular.id }}</th>
<td>
|   {{ formatDate(circular.applicationStartDate) }}
</td>
<td>
|   {{ (circular.applicationStartTime) }}
</td>
<td>{{ formatDate(circular.examDate) }}</td>
<td>{{ (circular.examTime) }}</td>
<td>
|   {{ formatDate(circular.applicationClosingDate) }}
</td>
<td>
|   {{ (circular.applicationClosingTime) }}
</td>
<td>{{ circular.status }}</td>
<td class="action">
|   <button
|     routerLink="/dashboard/admin/admission/circular/update/{{
|       circular.id
|     }}"
|     class="rtm__btn__icon--edit"
|   >
|     <i class="bi bi-pencil-square"></i>
|   </button>
|   <button
|     class="rtm__btn__icon--delete mt-2"
|     (click)="deleteAdmissionCircular(circular.id)"
|   >
|     <i class="bi bi-trash"></i>
|   </button>
| </td>
..

```

Figure 46: Admission circular list code sample – 2

```

@import "../../../styles/abstracts/mixins";
@import "../../../styles/base/variables";

.showList{
  width: 100%;
  min-height: 100vh;

  .rtm__table__topbar {
    display: flex;
    justify-content: space-between;
    button {
      border: none;
      background-color: transparent;
      font-size: 16px;
      i {
        font-size: 18px;
      }
    }
    input{
      width: 200px;
      font-size: 14px;
    }
  }

  .my-pagination ::ng-deep .ngx-pagination {
    li {
      font-size: 18px;
      padding: 8px 14px;
      color: $admin-text-secondary-clr;
      @include break(small) {
        font-size: 12px;
        padding: 4px 4px;
      }
    }
    li.current {
      background: $admin-primary-clr;
      color: $admin-text-secondary-clr;
    }
  }
}

```

Figure 47: Admission circular list code sample – 3

```

import { Component, OnInit } from "@angular/core";
import { ToastrService } from "ngx-toastr";
import { AdmissionService } from "../../../../../services/admission.service";
import { ErrorService } from "../../../../../services/error.service";
import { SweetAlertService } from "../../../../../core/services/sweet-alert.service";
import { FilterDataService } from "../../../../../core/services/filter-data.service";
@Component({
  selector: "app-admission-circular-list",
  templateUrl: "./admission-circular-list.component.html",
  styleUrls: ["./admission-circular-list.component.scss"],
})
export class AdmissionCircularListComponent implements OnInit {
  public circularList = [];
  public page = 1;
  constructor(
    private toastService: ToastrService,
    private admissionService: AdmissionService,
    private errorService: ErrorService,
    private sweetAlertService: SweetAlertService,
    private filterService: FilterDataService
  ) {}

  formatDate(date: any): any {
    return new Date(date).toLocaleDateString();
  }
  formatTime(date: any): any {
    return new Date(date).toLocaleTimeString();
  }

  applyFilter(filterValue: string) {
    filterValue = filterValue.toLocaleLowerCase();
    let rows = Array.from(document.querySelectorAll("tbody tr"));
    this.filterService.filterRowsData(rows, filterValue);
  }

  getAdmissionCircularList(): void {
    this.admissionService.getAdmissionCircularList().subscribe(

```

Figure 48: Admission circular list code sample – 4

```

        this.circularList = response.payload;
    },
    (error) => {
        this.errorService.handleError(
            error,
            "retrieving",
            "circular's"
        );
    }
);
}

ngOnInit(): void {
    this.getAdmissionCircularList()
}

async deleteAdmissionCircular(id: string): Promise<void> {
    let confirm = await this.sweetAlertService.confirm(
        "Are you sure?",
        "",
        "warning"
    );
    if (confirm) {
        this.admissionService.deleteAdmissionCircular(id).subscribe(
            (response) => {
                this.toastService.success(response.payload);
                this.getAdmissionCircularList();
            },
            (error) => {
                this.errorService.handleError(
                    error,
                    "deleting",
                    "circular"
                );
            }
        );
    }
}
}

```

Figure 49: Admission circular list code sample – 5

10.2 Possible problem breakdown

It is difficult to take on large projects at once and finish them correctly. Additionally, it will take a long time. On the other hand, the work will be easier to do and need less time if it can be broken down into manageable portions. The task will be finished effectively and correctly. Thus, the following is a summary of potential issues with IUMS ADMISSION SERVICE:

- Design and development of admission information
- Design and development of admission centers
- Design and development of the fees management component

Admission info design and development

- ❖ Admission circular design for create, view and update
- ❖ Admission circular develop for create, view, update and delete
- ❖ Admission requirement portion design for create, view and update
- ❖ Admission requirement portion develop for view, create, update and delete
- ❖ Admission subject option design for create, view and update
- ❖ Admission subject option develop for create, view, update and delete

Admission center design and develop

- ❖ Center design for create, view and update
- ❖ Center develop for create, view, update and delete
- ❖ Building portion design for create, view and update
- ❖ Building portion develop for view, create, update and delete
- ❖ Room option design for create, view and update
- ❖ Room option develop for create, view, update and delete

Fees management portion design and development

- ❖ Fees head design for create, view and update
- ❖ Fees head develop for create, view, update and delete
- ❖ Fees structure type portion design for create, view and update
- ❖ Fees structure type portion develop for view, create, update and delete
- ❖ Fees structure option design for create, view and update
- ❖ Fees structure option develop for create, view, update and delete

10.3 Prioritization while developing

Setting demand as a top priority is essential while creating a system. I've previously used the MoSCoW standard to rank the needs list according to relevance. The list of demands I will give priority to while creating the IUMS ADMISSION SERVICE is now visible.

Priority Serial	Requirement
1	Admission circular design for create, view and update.
2	Admission circular develop for create, view, update and delete.
3	Admission requirement portion design for create, view and update.
4	Admission requirement portion develop for view, create, update and delete.
5	Admission subject option design for create, view and update.
6	Admission subject option develop for create, view, update and delete .
7	Center design for create, view and update.
8	Center develop for create, view, update and delete.

9	Building portion design for create, view and update.
10	Building portion develop for view, create, update and delete.
11	Room option design for create, view and update.
12	Room option develop for create, view, update and delete.
13	Fees head design for create, view and update,
14	Fees head develop for create, view, update and delete.
15	Fees structure type portion design for create, view and update.
16	Fees structure type portion develop for view, create, update and delete,
17	Fees structure option design for create, view and update.
18	Fees structure option develop for create, view, update and delete.

Table 28: Prioritization of IUMS

Chapter 11 – Testing

11.1 Test Plan Acceptance

A test plan documents the resources needed to complete the project as well as the test strategy, objectives, timetable, budget, and deadlines. If you think of this as a testing template, you will understand what we are talking about. They support non-QA employees by giving detailed explanations of the testing process to business managers, engineers, and customer-facing teams. For quality assurance engineers, they act as a guide while performing their testing responsibilities. They go into considerable detail about topics such as the exam's scope, its estimated length, and its overall methodology. We've gathered all of this data into one place to make it easier for managers to assess and utilize. Primarily, there are two types of testing:

❖ **Functional Testing**

The method used by professionals in quality assurance to determine if a piece of software meets its stated requirements is called functional testing. Black-box testing techniques are used in functional testing since the tester is not aware of the reasoning behind the system at all. Finding out if a system operates as intended is the only objective of functional testing. The three categories of functional testing are listed below:

✓ **Unit Testing:**

- Input field validation.
- Filtering for specific types.
- For processing, pending, in realization, future contact, erroneous, and unobtained conversion filtering in the list.

✓ **Module Testing:**

- Complete the login form without providing your login information.
- The registration information or login credentials are invalid.
- Provide true and correct information while filling out a form.

✓ **Integration Testing:**

- Please enter the right login details.
- The process of adding and updating personal data was successful.

❖ **Non-Functional Testing:**

Non-functional testing examines the software's functionality and performance. Conversely, functional testing seeks to confirm a piece of software's entire functionality. Both functional and non-functional testing are equally vital in software development. Both help to guarantee that your product is performing as planned. Testing that isn't a part of the functional process is known as non-functional testing. Non-functional testing can take many different shapes.

-  **Security Testing**
-  **Usability Testing**
-  **Acceptance Testing**
-  **Accessibility Testing**

11.2 Test Case

In software engineering parlance, a system test is an experiment designed to accomplish a particular software testing goal, such confirming that a requirement is fulfilled or assessing a certain program route. A test case consists of an input parameter, programs, techniques, efforts to test a procedure, and expected results. Test cases are the first step in deliberate testing as opposed to random testing. It is possible to assemble a collection of test cases that sufficiently cover the application under test. Regression testing may be carried out more successfully and quickly by applying the same tests to progressively updated versions of the application using test cases that are clearly specified. The following list contains the test cases I used to evaluate this system:

Unit test – test case:

Test Case Name	Unit Test		
Test Class			
Test Description			
Source of Data	Test Steps	Expected Result	Actual Result

Table 29: Unit test – test case

Module test – test case:

Test Case Name	Module Test		
Test Class			
Test Description			
Source of Data	Test Steps	Expected Result	Actual Result

Table 30: Module test – test case

Integration test – test case:

Test Case Name	Integration Test		
Test Class			
Test Description			
Source of Data	Test Steps	Expected Result	Actual Result

Table 31: Integration test – test case

Chapter 12 – Implementation

12.1 Training

Training becomes much more important when software is added since users need to be taught how to operate the program. After software implementation, training is required to ensure optimal productivity, reduce employee turnover, and immediately boost morale and satisfaction levels. There are several methods for providing the training modules to the computer user as well as additional ways to keep the program updated after it is installed. When formulating a plan for successful execution, it is necessary to look at the company's key components. Support staff can be kept available for emergency situations by keeping managed services lines free of minor issue calls. Like port allocations and print capabilities, Training and support are required once the program is constructed because the infrastructure might not be robust enough to support specific software settings. The following table outlines the IUMS ADMISSION SERVICE training procedures:

SN.	Responsible User	Scopes of training	Time Period	Remarks
1	Admission Admin	For starting the training, one should create admission center details, admission requirements with necessary subjects, make available quotas for applicants and the application and admission fees successfully. Moreover, the responsible person needs to be trained on how to use the database in needed. On top of that, admission admin needs to know how to edit or update the information using above functionalities. Along with that, admission admin can change personal information that he	5 hours for training on admission process and 2 hours for application process	All the functions and operations will be understood by the admission admin easily.

		gave during registering into the system. The administrator will be instructed on the needed field for constructing the aforesaid function.		
--	--	---	--	--

Table 32: Training of Admission Admin

12.2 Big Bang

Three categories may be used to group the fundamental methods of implementing new systems at universities: big bang adoption, gradual adoption, and linear adoption. All users are able to become compliant with the current system while continuously improving the previous system since all old and new systems operate in parallel with linear adoption. "Phased adoption" describes how the strategy will be implemented step-by-step, bringing the system closer to full acceptance at each stage. There is a certain day known as the "instant system changeover" when the big bang adoption—the switch from the old system to the new one—occurs. On that day, the new system will become live for everyone, and the old system will be gradually phased away. This preparation will be described methodically using a massive boom acceptance. Big Bang operates by turning off the old systems and turning on the new one right away. Since this method uses the new system right away when the test is over, it is much faster than the others.

The primary advantage of big bang acceptance is that the suggested program doesn't need to interface with or be connected to any systems that already exist. This significantly streamlines the suggested system's overall architecture, especially in a company with several incompatible systems. Even while the big bang has less learning opportunities and needs more consideration than other acquisition kinds, it might nevertheless be more expensive. The systematic technique of the big bang acceptance method will be used to describe this preparation. The way Big Bang operates is by turning on the new system and immediately stopping the old one. This method is much faster than the others since it uses the new system right away when the test is finished. The proposed program may crash during the data transfer from the previous system, resulting in data

loss and instability. It is used on a single location with a unique admissions process for that establishment.

12.3 Scaling

The IUMS team members and team leader of DCL has done the scaling for this admission module. As I work as intern trainee, I received no information on this acceptance of scaling.

12.4 Load balancing

Load balancing is the process of improving a system over the influence of its users. Information on the number of concurrent users and system time is provided by the number of users hit. We call this load balance and load equalization. By dividing up the demand among several servers, it keeps the system operating efficiently. An optimal load balancing strategy should be implemented because there is only one admission admin user and a large volume of daily visits due to the system's size and thousands of customers' information it stores.

Chapter 13 – Critical Appraisal and Evaluation

13.1 Objective Could be Met

The following is the list of goals that were initially made public:

- ❖ Admission center with seat plan configuration and admission requirements setup.
- ❖ Application fees, admission test fees and admission fees setup.
- ❖ Application procedure with education board validation.

Objective/Goal – 1

Rate of achievement and others:

The admission admin of this system will be allowed to create the admission center information and make a plan of admission test's seat planning. Once this system is deployed the admission admin will be able to login to the system and after that he will get all options to create the admission center option. After creating the admission center information, when applicant will apply, he will get applicant list to distribute the applicant to the centers where they will seat for exam. This process will take a few minutes to complete all the action associated with admission center. Furthermore, the admission requirements will set by admission admin. And the everything of this functionality works well. That means that this goal has a 100% success rate.

Objective/Goal – 2

Rate of achievement and others:

Fees management of admission process includes application fees, admission test fees and admission fees. For creating the fees info, admission admin should create fees heads, fees structure type and fees structure respectively. There is a dependency of fees structure on fees head and fees structure type. The admission is also able to update the fees info according to the university policy. Since the job pertaining to fees operates flawlessly, the goal's success rate is 100%.

Objective/Goal – 3

Rate of achievement and others:

In application process, firstly, the applicant's eligibility will be checked whether he is eligible for admission test or not. Furthermore, it will check student's validity by analyzing the information of education board. Then the student will be able to apply and pay the application fees by a payment gateway. After getting all applications, the admission admin will schedule the exam and the applicant will be able to download the admit card and payment option for admission test fees. After the admission test and evaluating the papers the selected application will be able to get admitted through the payment of admission.

The full admission process of an applicant works perfectly so that means that this goal has a 100% success rate.

13.2 How much better could have been done

The IUMS ADMISSION SERVICE is a complete admission module of a university. However, nothing is ever really correct; things may always get better. Additionally, this admission module may be superior to the existing setup. Although all admission module criteria are met by the present system, admission is not yet optimized to its maximum potential. Sometimes the loading time is excessive. Certain applicants have longer than typical wait times while applying for the entrance exam. There are no significant issues that might be resolved in this system if there are no loading issues.

13.3 How better is the features of the solution

- ❖ **Admission requirements and Admission center and seat plan:** With this admission module the admission admin of a university will be able to keep records or create records of admission center, building and rooms for admission test and seat for the applicants. Along with this, the

admission admin will be able to add all the admission requirements which will be checked for applicant during application.

- ❖ **All required fees for admission:** The admission fees, admission test fees or application fees will be collected from applicant during his application process. Before that, admission admin should create all the necessary fees information so that they can collect the actual fees from the applicants.
- ❖ **Application process of applicant:** The applicant's eligibility will be examined in order to determine whether or not he is qualified for the entrance test. Additionally, it will verify the veracity of the student by examining the data from the education board. After that, the student can apply online and use a payment channel to pay the application fees. Following receipt of all applications, the admissions administrator will arrange the exam day and time, and the candidate will have access to download their admit card and choose how to pay for the exam. The chosen application will be eligible to be admitted by paying for admission following the entrance exam and evaluation of the papers.

13.4 Which features could not be touched

Although, this admission module has the all necessary function from apply to admission of an applicant. In addition, the admission admin will do the all essential task to take the admission test. But there is something that is not being touched. One is to distribute the student among departments and second one is, there is no option to assign student to their dormitory.

Chapter 14 – Lesson Learned

14.1 Pre Project – Review – Closing

IUMS ADMISSION SERVICE uses a web-based primary admissions procedure. My business used a predefined framework to build this system, which included obtaining requirements from the customer, creating a strategy for system creation, selecting an architecture, and choosing a project name. After that, I designed this system using Angular and produced project-compliant documentation. This IUMS ADMISSION SERVICE handles admission requirements, application procedures, administration center and fee generating, and other related matters.

14.2 What I have learned

The IUMS ADMISSION SERVICE is a university's admission solution. During my internship, I have worked as a frontend developer and contribute to this admission module. I gained a lot of knowledge while working on this project that will be very helpful to me in the future. Firstly, I had gone through the UI/UX because I had to convert the design by using HTML, SCSS and Bootstrap. During the design, I had to learn how to create tables or form. Then I had to learn that how to make validation for input field which was really one of the toughest task for me. After all of my learning about design and implement within this software. When the design finished, I started to learn about API and how to work with them. Without proper binding the APIs will not work. Then I practiced with dummy project for binding the APIs before starting bind into the real project. After that I started to bind the APIs in IUMS ADMISSION SERVICE to make the functions visual. In these few days, it was really tough to learn these all things and implement is an enterprise application. Along with that I had to write a commented and clean code because whenever a new developer will join he can understand the codes, and it should be more readable for him. Furthermore, I had to make all functions logic easier in order to reduce the complexity. After completing the project, a demonstration session has arrived, for that, I had to learn to build the angular project for deploying on a server. As a result, I can state that the experience was fantastic because this project has taught me much too much.

14.3 The Problems I Have Faced

In this IUMS ADMISSION SERVICE project, I was a frontend developer. I have encountered several challenges and issues during the project. Working with the project architecture that the team leader had built and comprehending the designer's design to convert was the biggest hurdle I faced. because working with architecture that has been designed by others can be quite difficult. Gaining an understanding of the architecture takes time. I have to face and overcome challenges in an iterative way since I employed the Agile DSDM process, which assures iterative development. This shows that issues that arose in the one-time box are fixed in the time box and job that followed. Even though I had started creating the system, I was instructed to use templates. But because I didn't know how to work with the templating, it took me some time to create the design. I also had to pick up new skills in user-friendly design. I ran into a few problems when I first started working with APIs. First, let's learn how to build functions for a variety of tasks, including posting, updating, and deleting data. I found it difficult to work on my first job. I learned how to recover such flaws and used them in my project. Developing the frontend for a project of that scale was particularly challenging for me because it was my first experience working on one. Writing a thousand lines of code from scratch was not much simpler. It took me longer than usual to design the system because there were a lot of new things I had to learn. This project needed to develop with a specific IP address rather than localhost because it needed to be deployed on a server. It was something new for me and wasn't all that difficult.

14.4 What Solution Occurred

Every problem has a solution that may be applied to resolve it. I completed the method and came up with answers for every problem I mentioned above. Firstly, I need to understand the design that was done by the UI/UX designer. Deciphering the designer's vision and navigating the complexities of the pre-established project architecture proved to be significant challenges during my time working as a frontend developer on the IUMS ADMISSION SERVICE project. I carefully documented the architecture, had regular conversations with the team leader, and asked the designers for clarification in order to manage the complexity. I approached problems iteratively, adhering to the Agile DSDM approach, making sure that issues from one time box were resolved

in later rounds. Following instructions to use templates for system design, I committed time to becoming knowledgeable about and putting templating approaches into practice. Online classes and feedback loops were used to refine user-friendly design abilities. There were obstacles in my way when I first started integrating APIs, but I overcome them by gradually creating functions for different jobs and valuing extensive testing. Managing the demands of a major project required breaking it down into smaller, more manageable tasks, keeping regular communication channels open, and seeking assistance as needed. It was a new idea to deploy the project on a server with a designated IP address, but I was able to adjust to this need with the help of DevOps. Every obstacle served as a springboard for my progress as I learned how to handle the complexity of frontend development on a large project.

Chapter 15 – Conclusion

15.1 Summary of the project

The intelligent web application IUMS ADMISSION SERVICE was created using a strong tech stack that included Angular, HTML, CSS, SCSS, Bootstrap, and Angular to produce a dynamic and user-friendly frontend. This all-inclusive system offers administrators and applicants a smooth admissions process by acting as a central center for managing a variety of admissions-related tasks. The frontend development guarantees a responsive and aesthetically pleasing experience by utilizing HTML, CSS, Bootstrap, and SCSS. A consistent and visually appealing design is enhanced by Bootstrap's pre-designed components and responsive layout tools. The use of Angular, a potent JavaScript framework, improves user experience by facilitating the development of dynamic, single-page apps that react rapidly to input from the user. The flexibility with which the IUMS entrance SERVICE manages entrance standards is one of its primary characteristics. The system makes it simple for administrators to add, update, or change admission requirements, guaranteeing that the platform will always be flexible enough to meet the institution's changing demands. Educational institutions are better equipped to keep up with evolving admission norms and requirements because to this functionality. The system extends its capabilities to cover center establishment and fees management in addition to controlling admission criteria. Administrators can effectively configure and arrange several admission centers with the help of center setup functionalities. This function offers a consolidated platform for managing the setup and operation of admission centers, streamlining the administrative processes involved in conducting admissions at several locations. Another essential component of the IUMS ADMISSION SERVICE is fee management. The solution makes it easier to create, adjust, and keep an eye on charge structures, guaranteeing precise and open financial transactions. Setting amounts, managing payment schedules, and configuring various fee categories are all simple tasks for administrators. This all-inclusive fee management strategy gives students and their families transparency while also supporting the financial stability of educational institutions. A user-friendly interface that streamlines the application process is advantageous to applicants. By utilizing Angular's features, the system offers a smooth, single-page application experience, speeding up load times and improving responsiveness all around. Prospective students will find it easy to fill out the

application, submit the essential files, and input the necessary information. The application submission process is made seamless and effective by the system, which improves the applicant experience in general.

In conclusion, IUMS ADMISSION SERVICE is proof of the cooperation of frontend technologies with a carefully considered system architecture. An effective, responsive, and user-friendly platform is created by combining the use of HTML, CSS, SCSS, Bootstrap, and Angular. The system offers a comprehensive solution for managing the intricacies of the admission process, according to the diverse needs of educational institutions. Its features include management of admission requirements, center setup, fees management, and a streamlined applicant interface.

15.2 Goal of the project

During my internship at DCL, the organization's goals dictated the project goal for the IUMS ADMISSION SERVICE system. Within the software industry, a system's objective aligns with the exact requirements and specifications of the customer or client. In a similar vein, the IUMS ADMISSION SERVICE system's objective has been set in compliance with the demands provided by the client. This helps to ensure that the IUMS ADMISSION SERVICE is developed and implemented successfully by ensuring that the system's features and functionalities are customized to match the unique requirements and expectations stated by the customer.

- By carefully configuring and setting up admission centers, seat arrangements, and admission requirements, the main objective is to expedite the admissions process. In order to optimize the admission workflow as a whole, this entails developing an effective and well-structured system that smoothly incorporates a variety of components, including seat arrangements for admission centers.
- The goal is to configure application, admission test, and admission costs in a transparent financial system. In order to do this, a thorough system that supports a variety of fee categories and guarantees transparency and correctness in financial transactions related to the admissions process must be established. Enhancing financial transparency and giving administrators and applicants a clear breakdown of fees are the goals.

- The goal is to create an application process that is both validated and easy to use. This objective includes developing an easy-to-use application process for candidates and integrating validation from the education board to guarantee the veracity and correctness of the data submitted. By streamlining the application process and putting in place validation procedures that adhere to the standards established by education boards, the goal is to improve the user experience and make the admissions process more dependable and effective.

15.3 Success of the project

The IUMS ADMISSION SERVICE has achieved a remarkable degree of accomplishment, having successfully and precisely met its set objectives. The creation of admission centers, in conjunction with the arrangement of seating arrangements and the methodical setting of entrance prerequisites, is indicative of a complete plan aimed at improving the administration and coordination of the admissions process. This achievement guarantees that the admissions process is efficient and well-structured. The careful structuring of application fees, entrance exam fees, and admission fees has simplified financial administration inside the system. Administrators and applicants alike gain from this dedication to accuracy and openness in financial transactions. The administrative procedure is made easier by the system's ability to manage a variety of charge types, which guarantees flexibility in accommodating different admittance scenarios. The system's integration of an easy-to-use application process with education board validation is one of its noteworthy accomplishments. This complies with education board regulations and guarantees a dependable and safe application process. The admissions process is made easier for applicants by a system that not only makes the procedure simpler but also verifies the accuracy of the information submitted, giving the system an additional degree of dependability. The accomplishment of these objectives demonstrates the development team's technical proficiency and their capacity to convert customer needs into a reliable and useful solution. The IUMS ADMISSION SERVICE is proof of how well developers and clients can work together to provide solutions that not only meet but also surpass expectations. This accomplishment enhances the system's reputation as a dependable and effective instrument for handling the difficulties of the admissions process, enhancing administrators' and applicants' overall satisfaction and success.

15.4 What I have done in the documentation

The proposed project's documentation moves through several important phases. The project and system are introduced, and then the first phase's preliminary research findings are examined, covering the project's objectives, problem areas, possible remedies, and history. The chapter on literature review explores the problem domain, current remedies, and suggested changes. The significance and use of different approaches are emphasized in the methodology section. Details on project planning, including project plans, test plans, and risk management, are included in the planning chapter. The feasibility chapter includes a comprehensive report on the feasibility research as well as a cost-benefit analysis. Issue area identification, general need lists, suggested technologies, and their explanations are all covered in the foundation part. The exploration phase includes a prototype, need lists, and UML diagrams; the engineering chapter examines the behavioral and logical models of the proposed system. Development prioritizing techniques and coding samples are covered in the deployment chapter. While tactics and training models are covered in the implementation section, testing includes a variety of test ideas and outcomes. The following step is a critical appraisal and evaluation that looks at how well the first objectives were achieved and the lessons learned. The project's objectives, accomplishments, and most important lessons are outlined in detail in the documentation's conclusion.

15.5 Value of the project

The many functions that IUMS ADMISSION SERVICE offers to simplify and expedite the admissions process for applicants and educational institutions alike form the foundation of its value proposition. One of the system's most notable features is how well it configures admission requirements, giving institutions the ability to design and modify requirements in response to standards that change over time. This keeps schools flexible enough to adapt to shifting admissions environments while simultaneously guaranteeing devotion to academic excellence. An additional layer of value is added by the arrangement of admission centers, which includes seat layouts and makes it easier to handle the admissions process strategically and well. Furthermore, financial transactions are made transparent and clear by the system's skill in fees management setup, which includes application, admission exam, and general admission costs. This function improves the

institution's overall financial integrity while also lessening the administrative strain. The application process is user-centric and incorporates education board validation. It is designed to be a seamless experience for applicants, giving them a dependable and simple application process. Beyond these separate elements, IUMS ADMISSION SERVICE's overall value comes from its capacity to synchronize and integrate various aspects into a seamless, end-to-end solution. The technology guarantees adherence to academic standards, maximizes productivity, decreases manual labor, and improves financial transparency. The ability to provide educational institutions with a comprehensive tool that not only satisfies the complexities of the admissions process but also enhances the overall experience for administrators and applicants is at the core of IUMS ADMISSION SERVICE's value proposition, which ultimately contributes to the success and expansion of educational institutions.

15.6 My Experience

I had the chance to work in a dynamic and demanding environment as a frontend developer for the IUMS ADMISSION SERVICE project, where I was able to improve the system's functionality and user experience by leveraging a flexible tech stack. I created visually beautiful and responsive web pages with HTML and CSS to make sure users had the best possible experience. Effective design improvements were made possible by the greater flexibility and maintainability that the inclusion of SCSS provided to the styling process. Bootstrap, which offered pre-designed components that complemented the project's design language, was essential in accelerating the development process. Angular, being the fundamental JavaScript framework, enabled me to develop dynamic and interactive single-page applications that enhance user experience. I was able to face and conquer challenges in small steps because to the iterative development technique, which was informed by Agile DSDM concepts. This ensured that the frontend will continue to evolve. I became proficient in API integration as part of my duties and learned how to create functions for posting, updating, and deleting data. Through this experience, I was able to improve my coding skills and make a meaningful contribution to the system's overall operation. It was a unique challenge to learn how to design using templates, but after some perseverance, I was able to handle templating effectively, leading to a more uniform and integrated system design. The project's demand that the server be deployed with a specific IP address instead of localhost

broadened my knowledge of the development lifecycle and exposed me to the subtleties of server deployment. To sum up, IUMS ADMISSION SERVICE presented me with a wide range of technical difficulties and educational prospects. My ability to contribute significantly to frontend development was enhanced by the inclusion of HTML, CSS, SCSS, Bootstrap, and Angular. Additionally, the integration of APIs introduced a layer of complexity that broadened my skill set. By conquering obstacles and adopting new technologies, I improved the system's functionality and refined my skills as a frontend developer in a large-scale, real-world project setting.

Appendix

User guide

Admission Circular List

+ New

Name	Application start Date	Application start Time	Exam Date	Exam Time	Application end Date	Application end Time	Status	Action
« Previous 1 Next »								

Admin and admission admin can view the admission circular list

ADMISSION CIRCULAR CREATE

Name

Exam Date

Application Start Date

Application Closing Date

Payment Closing Date

SSC Passing Year

Status

Instruction

Admin and admission can create admission circular

Admission ▲
Admission Circular
Admission Subject
Admission Requirements
Quota
Admit Card Signatory
Applicant Selection
Selected Applicant
Assign Applicant To Center
Admission Center ▼
Fees Management ▼
Reports ▼

Admission Subject

+ New
Search

ID.	Exam Type	Subject Name	Subject Code	Action
-----	-----------	--------------	--------------	--------

Admin and admission admin can see the admission subject list

Admission ▲
Admission Circular
Admission Subject
Admission Requirements
Quota
Admit Card Signatory
Applicant Selection
Selected Applicant
Assign Applicant To Center
Admission Center ▼
Fees Management ▼
Reports ▼

SUBJECT CREATE

Exam Type*
Select exam ▼

Subject Name*
Enter name

Subject Code*
Enter code

CREATE

Admin and admission admin can create the admission subject

Admission ▲
Admission Circular
Admission Subject
Admission Requirements
Quota
Admit Card Signatory
Applicant Selection
Selected Applicant
Assign Applicant To Center
Admission Center ▼
Fees Management ▼
Reports ▼

Admission Requirements

+ New
Search

ID.	Exam	Group	GPA	Action
-----	------	-------	-----	--------

Admin and admission admin can view admission requirements list

Admission ▲
Admission Circular
Admission Subject
Admission Requirements
Quota
Admit Card Signatory
Applicant Selection
Selected Applicant
Assign Applicant To Center
Admission Center ▼
Fees Management ▼
Reports ▼

Exam*
HSC
Group
Select Group
GPA
Enter gpa
Create

HSC Subject*
Select Subject
grade*
Select grade
Add
Add

Subject Name	Code	Grade	Action
--------------	------	-------	--------

Admin and admission admin can create the admission requirements

Admission ▲
Admission Circular
Admission Subject
Admission Requirements
Quota
Admit Card Signatory
Applicant Selection
Selected Applicant
Assign Applicant To Center
Admission Center ▼
Fees Management ▼
Reports ▼

Admit Card Signatory

+ New

ID.	Name	Designation	Signature	Action
« Previous 1 Next »				

Admin and admission admin can see the admit card signatory list

Admission ▲
Admission Circular
Admission Subject
Admission Requirements
Quota
Admit Card Signatory
Applicant Selection
Selected Applicant
Assign Applicant To Center
Admission Center ▼
Fees Management ▼
Reports ▼

SIGNATORY CREATE

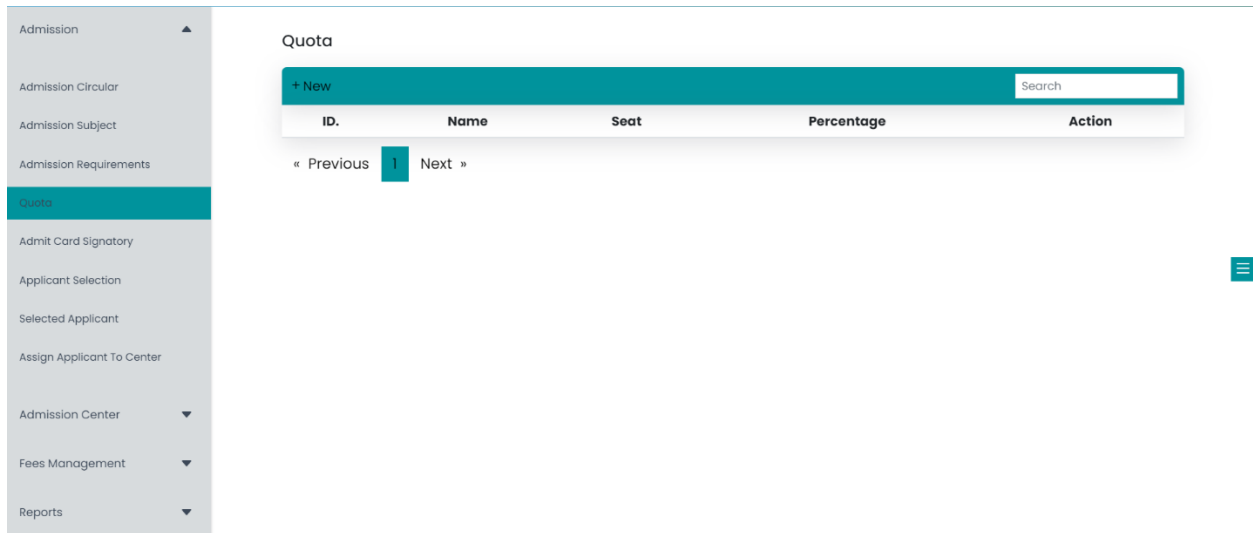
Name

Designation

Signature

CREATE

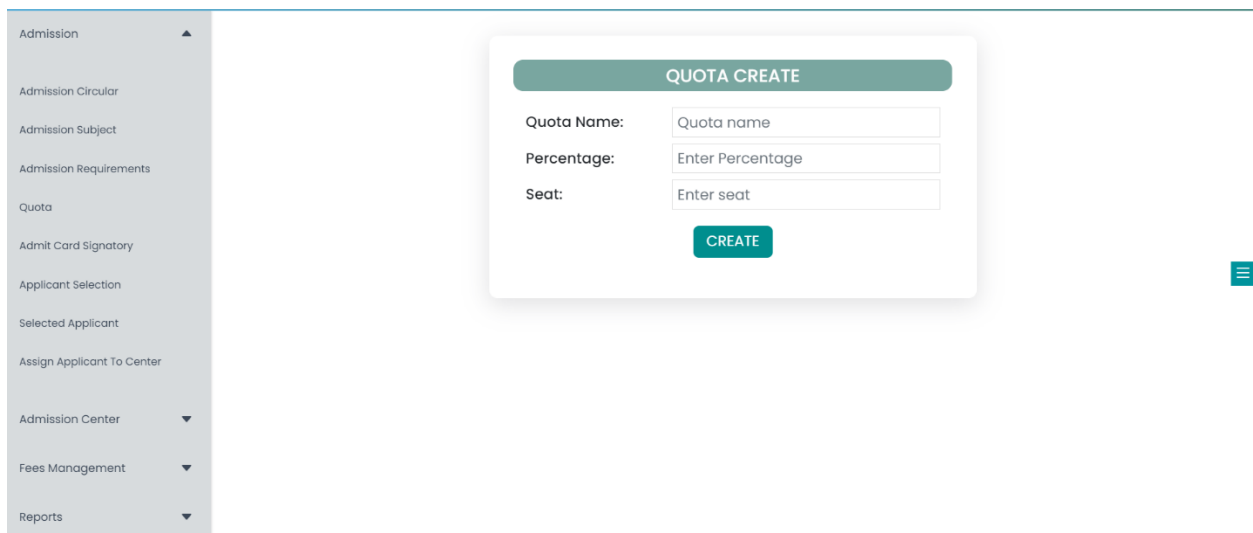
Admin and admission admin can create the admit card signatory



The screenshot displays the 'Quota' management interface. On the left is a sidebar menu with the following items: Admission, Admission Circular, Admission Subject, Admission Requirements, Quota (highlighted), Admit Card Signatory, Applicant Selection, Selected Applicant, Assign Applicant To Center, Admission Center, Fees Management, and Reports. The main content area is titled 'Quota' and features a '+ New' button and a search bar. Below these is a table with the following columns: ID, Name, Seat, Percentage, and Action. The table shows one entry with ID '1'. Navigation links '« Previous' and 'Next »' are present. A teal menu icon is located on the right side of the interface.

ID	Name	Seat	Percentage	Action
1				

Admin and admission admin can see the quota list



The screenshot shows the 'QUOTA CREATE' form. The sidebar menu is identical to the previous screenshot, with 'Quota' highlighted. The main content area contains a form with the following fields: 'Quota Name:' with a text input field containing 'Quota name', 'Percentage:' with a text input field containing 'Enter Percentage', and 'Seat:' with a text input field containing 'Enter seat'. A teal 'CREATE' button is positioned below the form fields. A teal menu icon is located on the right side of the interface.

Admin and admission admin can create the quota

Dashboard

Profile

Change Password

Enterprise ▲

Admission ▼

Admission Center ▲

Center

Building

Room

Fees Management ▼

Reports ▼

Admission Center

+ New

Search

ID.	Name	Location	Capacity	Action
-----	------	----------	----------	--------

Admin and admission admin can see the admission center list

Dashboard
Profile
Change Password
Enterprise
Admission
Admission Center
Center
Building
Room
Fees Management
Reports

ADMISSION CENTER CREATE

Name:
Center name

Location:
Center location

Capacity:
Center capacity

CREATE

Admin and admission can set the admission center

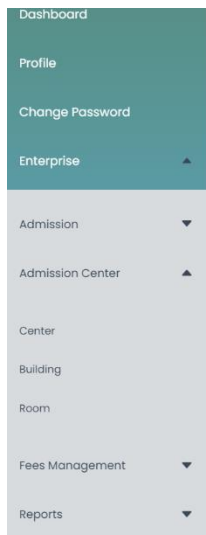
Dashboard
Profile
Change Password
Enterprise
Admission
Admission Center
Center
Building
Room
Fees Management
Reports

Admission building

+ New
Search

ID.	Name	Capacity	Center	Action
« Previous	1	Next »		

Admin and admission admin can see the admission building list



ADMISSION BUILDING CREATE

Admission Center*

Select center

Name*

Building name

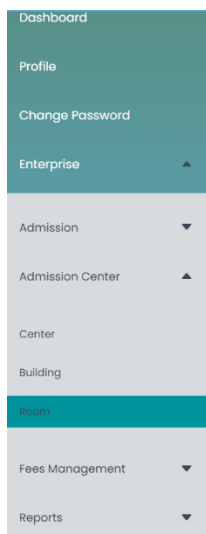
Capacity*

Building capacity

CREATE



Admin and admission admin can create the admission building

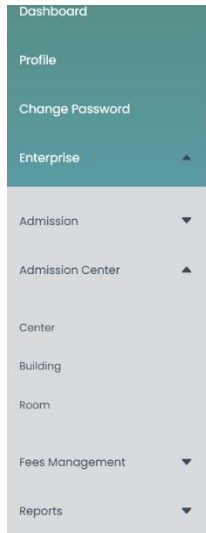


Admission Center Room

+ New						Search
ID.	Center	Building	Room	Capacity	Action	



Admin and admission admin can see the admission center room list



ADMISSION CENTER ROOM CREATE

Center:

Building

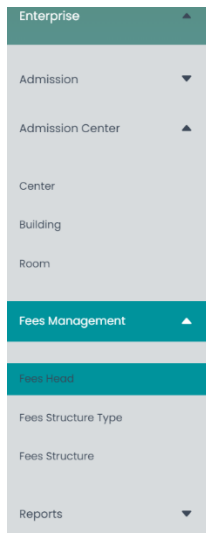
Room No*

Capacity:

CREATE



Admin and admission admin can create admission center room



Fees Head

+ Now					Search
ID.	Name	Amount	Privilege Applicable	Action	



Admin and admission admin can see the fees head list

Enterprise ▲
Admission ▼
Admission Center ▲
Center
Building
Room
Fees Management ▲
Fees Head
Fees Structure Type
Fees Structure
Reports ▼

FEE HEAD CREATE

Name:

Amount:

☐ Scholarship Privilege Applicable

CREATE

Admin and admission admin can create fees head

Enterprise ▲
Admission ▼
Admission Center ▲
Center
Building
Room
Fees Management ▲
Fees Head
Fees Structure Type
Fees Structure
Reports ▼

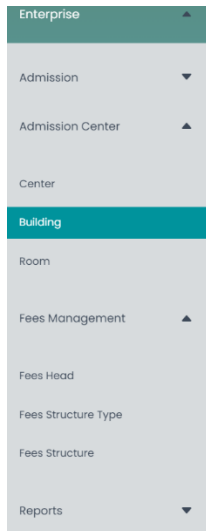
Fees Structure Type

+ New

Search

ID.	Name	Action
« Previous	1	Next »

Admission admin and admin can see the fees structure type



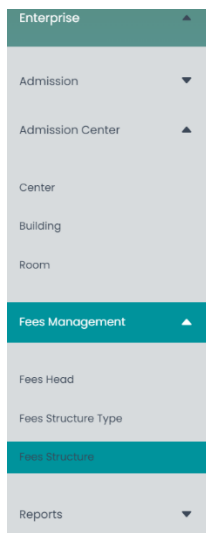
FEES STRUCTURE TYPE CREATE

Name:

CREATE



Admin and admission admin can create the fees structure type



Fees Structure

+ New			
ID.	Name	Type	Action
« Previous 1 Next »			



Admin and admission admin can see the fees structure

The screenshot shows a web application interface for creating a fees structure. On the left is a vertical sidebar menu with the following items: Enterprise (expanded), Admission, Admission Center, Center, Building, Room, Fees Management, Fees Head, Fees Structure Type, Fees Structure, and Reports. The main area contains a modal form titled 'FEES STRUCTURE CREATE'. The form has three input fields: 'Name' with the placeholder text 'Fees structure name', 'Type' with a dropdown menu showing 'Select Type', and 'Fees Head' with a dropdown menu showing 'Select fees head'. Below these fields is a green 'CREATE' button. A hamburger menu icon is located on the right side of the page.

Admin and admission admin can create the fees structure

Works Cited

- ApacheBooster. (2020). *Client server technology*. Retrieved 12 09, 2021, from <https://apachebooster.com/blog/client-server-technology/>
- contributors, W. (2021). *Web application*. Retrieved 12 10, 2021, from https://en.wikipedia.org/w/index.php?title=Web_application&oldid=1069803114
- Dopson, E. (2020). *Resource Allocation*. Retrieved 2 13, 2022, from <https://www.float.com/resources/guide-to-resource-allocation>
- Fink. (2014). *Research Guides*. Retrieved 1 25, 2022, from <https://libguides.usc.edu/writingguide/literaturereview>
- GeeksforGeeks. (2019). *Unit testing*. Retrieved 11 22, 2021
- Hassan, S. (2021). *The rice economy* . Retrieved 11 08, 2021, from <https://www.thedailystar.net/business/economy/news/the-rice-economy-2114957>
- HAYES, A. (2021). *Cost-Benefit Analysis*. Retrieved 2 13, 2022, from <https://www.investopedia.com/terms/c/cost-benefitanalysis.asp>
- Idesis, S. (2020). *Rapid Application Development*. Retrieved 2 13, 2022, from <https://www.outsystems.com/blog/posts/rapid-application-development>
- Ltd}, { . A. (2021). *Dynamic Systems Development Methodology*. Retrieved 11 19, 2021, from <https://www.ukessays.com/essays/information-systems/dynamic-systems-development-methodology.php>
- Martin, M. (2021). *What is a Functional Requirement in Software Engineering? Specification, Types, Examples*. Retrieved 2 13, 2022, from <https://www.guru99.com/functional-requirement-specification-example.html>
- Martin, M. (2021). *What is Non-Functional Requirement in Software Engineering? Types and Examples*. Retrieved 2 13, 2022, from <https://www.guru99.com/non-functional-requirement-type-example.html>
- Musotsi, L. (n.d.). *Problem Statement For University Management System*. Retrieved from <https://www.scribd.com/document/598325429/problem-statement-for-university-management-system>

- Nikolaieva, A. (2018). *Best Software Development Methodologies*. Retrieved 2 13, 2022, from <https://www.uptech.team/blog/software-development-methodologies>
- ogbebor, o. (2011). *Feasibility*. Retrieved 2 13, 2022, from <https://osarome.blogspot.com/2011/10/1-technical-feasibility-2-operational.html>
- Project Manager. (2015). *Project Planning*. Retrieved 2 13, 2022, from <https://www.projectmanager.com/project-planning>
- Radel, F. E. (n.d.). *Problems and limitations of university management*. Retrieved from https://www.researchgate.net/publication/338528891_Problems_and_limitations_of_university_management
- Sherman, R. (2015). *Project Management*. Retrieved 2 13, 2022, from <https://www.sciencedirect.com/topics/computer-science/waterfall-methodology>
- Sixsigma. (2021). *Dynamic Systems Development Method*. Retrieved 2 13, 2022, from <https://www.whatissixsigma.net/dsdm-dynamic-systems-development-method/>
- THE INVESTOPEDIA TEAM. (2021). *Feasibility Study*. Retrieved 2 13, 2022, from <https://www.investopedia.com/terms/f/feasibility-study.asp>
- ValuePLUS Computer Systems Ltd. (n.d.). *Integrated University Management System(IUMS)*. Retrieved from <https://www.vpcsb.com/products/iums/>
- Waterfall methodology*. (2021). Retrieved 11 19, 2021, from <https://www.workfront.com/project-management/methodologies/waterfall>
- Wikipedia. (2021). *Comparison*. Retrieved 2 13, 2022, from <https://en.wikipedia.org/wiki/Comparison>

Plagiarism Report

Habiba 201-16-490

ORIGINALITY REPORT

1 %

SIMILARITY INDEX

1 %

INTERNET SOURCES

0 %

PUBLICATIONS

1 %

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Colorado Technical University

Student Paper

< **1** %

2

dspace.daffodilvarsity.edu.bd:8080

Internet Source

< **1** %

3

Submitted to Softwarica College Of IT & E-Commerce

Student Paper

< **1** %

4

Submitted to University of Maryland, Global Campus

Student Paper

< **1** %

5

Submitted to Oxford & Cherwell Valley College

Student Paper

< **1** %

6

byte-project.eu

Internet Source

< **1** %