Campus Content Management System

22-FYP-205

Session 2019-2023

BACHELOR OF SCIENCE

IN

SOFTWARE ENGINEERING

SUBMITTED BY

Mukarram Shahbaz 19-NTU-CS-1117

Wasi Ur Rehman Qamar 19-NTU-CS-1130

SUPERVISED BY

Muhammad Shahid

Humael Hussain

Department of Computer Science

National Textile University, Faisalabad

CERTIFICATION

DECLARATION

We hereby declared that this document is completely written by us, and it is totally our effort and none of anyone from outside of our group have copied it. This Report is purely written in a technical way in accordance with our project.

Group Members:

Mukarram Shahbaz _____ _____

19-NTU-CS-1117 Signature Date

E-Mail: mukarramshahbaz8@gmail.com

Wasi Ur Rehman Qamar _____ _____

19-NTU-CS-1130 Signature Date

E-Mail: waasiqamar@gmail.com

Acknowledgement

Above all else, For the sake of Allah, the Most Advantageous, the Most Benevolent. Every one of the gestures of recognition also, thanks be to Allah. A ton of adoration for our cherished Heavenly Prophet MUHAMMAD (S.A.W) , his direction generally assists us with getting the correct way. From that point onward, we additionally gladly announce that the achievement boundaries of this venture are the vast supplications and backing of our folks. We likewise need to recognize the endeavors of our supervisor, who helped us a lot in the project, particularly in the documentation. Each time we need to meet him, he is continuously there for us with good thoughts, conversation and ideas that help us a great deal in fruition of our project and documentation. We likewise need to thank the individuals who helped us in various phases of the venture. We are exceptionally appreciative to the Program Director, Bachelor of Software Engineering and final year project convener for outrageous strength, mindful, persistence, delicacy, liberality, extraordinary mindfulness and for propelling us to do something remarkable and progressive.

Presently venturing ahead, we wish to pay our glow of appreciation and gratefulness to our noteworthy supervisors who delivered their benevolent administrations to help us in various parts of the trouble spot. We are appreciative to our most regarded educators Mr. Muhammad Shahid and Mr. Muhammad Nouman for being dependably some assistance in all respects of programming standards, this aided us in all advancement levels of the undertaking also, particularly for making us ready to push ahead in the field of computer programming furthermore, plan. We gladly proclaim that the main driver of our outcome in the field of programming designing are our instructors by their very strong disposition for continuously being a making a difference hand for us. We can't overlook the help of the IT administrations division. Particularly the IT administrations staff for giving us a spot to sit and work in the IT division and the related help on request. Their aiding demeanor was amazing. Finally, yet all the same not the least, we additionally need to thank our class colleagues particularly for sacrificial assistance, counsel, and ideas in various stages going from elicitation to organization.

Abstract

We are proud to announce the development of a new campus content management system (CMS) to replace the aging system that has been in use for the past 7 years. The new CMS is built with the latest technology and includes additional functionalities designed to improve the management of digital content across the university. The new CMS is designed to provide an easy-to-use platform for managing and publishing digital content, including website content, documents, images, and videos. It offers an intuitive, user-friendly interface that will help faculty, staff, and students navigate the system with ease.

New functionalities have been added to the system to further enhance its capabilities. For example, the system now includes more powerful tools for managing website content, such as advanced content editing features and streamlined publishing workflows. It also includes new features to assist in managing digital assets, such as a central repository for images and videos that can be easily searched and reused across multiple web pages and publications.

Overall, the new campus content management system represents a significant upgrade to our university's digital infrastructure. It is designed to provide a more streamlined and efficient content management system for the campus, while offering a more modern and intuitive user experience for faculty, staff, and students. We are excited to launch the new system and look forward to the benefits it will bring to our campus community.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CMS	Campus Management System
SDLC	Software Development Life Cycle

CHAPTER 1

1. Introduction

A campus management system is a software solution designed to streamline and automate the various administrative and academic tasks of a university or college. It includes a range of modules that help manage tasks such as student enrollment, course scheduling, faculty management, resource allocation, and more. Campus management systems typically provide a centralized platform for managing all aspects of campus operations, from financial and administrative tasks to academic and research activities. They often include features such as online registration, grade management, and student record tracking, as well as tools for managing campus facilities, equipment, and other resources.

Campus management systems can help universities and colleges improve efficiency, reduce administrative workload, and provide a more streamlined and integrated experience for students, faculty, and staff. They can also provide valuable data insights and reporting to help institutions make data-driven decisions and improve performance over time. With the increasing availability of cloud-based solutions, campus management systems are becoming more accessible and cost-effective for a wider range of institutions, from small colleges to large universities.

1.1. Dynamically created websites

Dynamic website generation or creation refers to the process of creating websites using server-side programming languages and technologies that generate web pages dynamically, based on user requests and other factors. In a dynamic website, content is generated on the fly, allowing for more flexible and customizable user experiences. This contrasts with static websites, which are created using HTML and other static content files that are served to the user without any server-side processing. Dynamic website generation can be accomplished using a variety of technologies, including PHP, Java, Python, Ruby on Rails, and others. These technologies allow for the creation of dynamic web pages that can be customized based on user input, database queries, and other factors.

Overall, dynamic website generation is an important technology for creating modern, user-friendly websites that can be easily customized and maintained. It is an essential tool for many businesses, organizations, and institutions, including universities and colleges that need to manage complex content and provide customized user experiences for students, faculty, and staff.

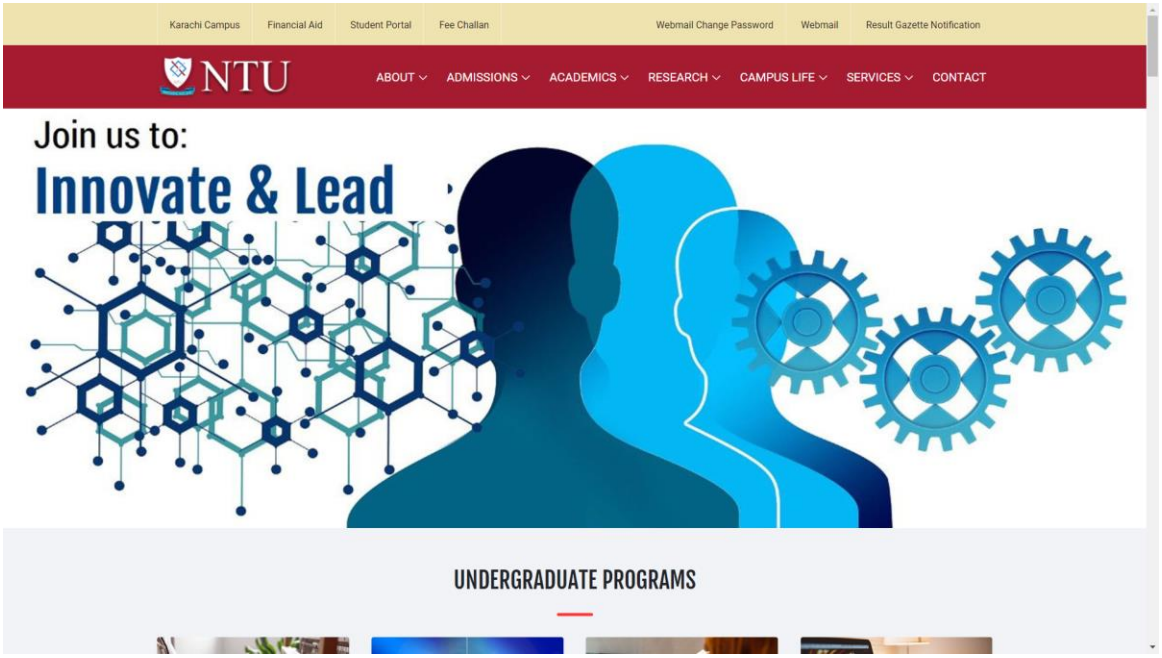

Figure 1.1: Dynamically created website with some limitations and on old technology

1.2. Problem Statement

The current campus content management system at our university is outdated and has been in use for the past 7 years. It has become increasingly difficult to manage and update the system, leading to challenges for students, faculty, and staff in accessing important information and resources.

The lack of these essential functionalities has led to a poor user experience for students, faculty, and staff, and has caused inefficiencies in campus operations. This has resulted in increased administrative workload, longer turnaround times for important tasks, and decreased overall satisfaction with the system.

To address these challenges, we are developing a new campus content management system that leverages modern technology and incorporates key functionalities that were missing in the previous system. The new system will provide a dynamic website that is mobile-responsive, customizable, and accessible on different devices. The new campus

content management system will help address these issues and provide a more streamlined and user-friendly system for the university community.

1.3. Purpose

The purpose of developing a new campus content management system is to address the limitations of the old system, which is built on outdated technology and lacks important features required for modern content management. The new system will provide an updated and more efficient platform for managing and delivering digital content, allowing for improved user experiences, more streamlined workflows, and increased collaboration and communication within the campus community.

1.4. Project Goals

The goal of developing a new campus content management system is to overcome the limitations of the old system and generate a dynamic website using server-side programming languages. By leveraging modern technology and programming languages such as Python (Django), the new system will provide a more flexible, customizable, and efficient platform for managing and delivering digital content to the campus community, allowing for a better user experience, and streamlined workflows.

1.5. Objectives

The objectives of developing a new campus content management system to make it functional on campus, overcome old limitations, and generate a website dynamically through a programming language are:

- To provide a centralized and efficient platform for managing digital content across the campus community.
- To leverage modern technology and programming languages to create a flexible and customizable system that can adapt to the changing needs of the campus.
- To overcome the limitations of the old system and implement new features and functionalities such as social media integration, online forms, and personalized landing pages.

- To provide an improved user experience for students, faculty, and staff by providing easy and quick access to important information, resources, and services through a dynamic and user-friendly website.
- To streamline workflows and improve collaboration and communication within the campus community, leading to a more efficient and productive work environment.

1.6. Project Scope

The scope of developing a campus content management system at the university includes building a system architecture that can handle large volumes of data and traffic, creating a user-friendly interface, and implementing modern programming languages and tools to build a dynamic and customizable website. The system should support content creation, storage, sharing, and distribution across departments and campuses, as well as collaboration and communication among faculty, staff, and students. Security and access control, integration with emerging technologies, and analytics and reporting are also important components of the system.

1.7. Proposed Solution

The purpose of developing a campus content management system using PYTHON (DJANGO) technology is to provide an effective solution to the limitations of the old system that was built using PHP technology. By shifting to modern PYTHON (DJANGO) technology, the new system can overcome the limitations of the old system and provide a dynamic and customizable website that can handle large volumes of data and traffic. The use of the PYTHON (DJANGO) in developing a campus content management system will allow us to generate websites dynamically through code. The code-based approach of the PYTHON (DJANGO) will also enable you to customize the system to the unique needs of your university. The PYTHON development includes using Microsoft SQL Server for database management, Python for server-side scripting, and DJANGO for client-side development. This will result in a more scalable, reliable, and user-friendly system for managing campus content.

The PYTHON (DJANGO) provides a modern and flexible platform for building dynamic web applications that can handle large volumes of data and traffic. Together, the PYTHON (DJANGO) offers a powerful set of tools and technologies to develop a

campus content management system that can meet the complex and diverse needs of a university.

1.8. Project Scheduling

Great, having a complete timeline of the project development is an important step in successful project management. With a complete timeline, you can ensure that all tasks are completed on time, within budget, and according to the project requirements. This can help to ensure that the project is completed on schedule and with the desired quality. The project's overall schedule, as well as the beginning and ending times and length of time for each activity, are displayed using Gantt charts. Therefore, the project's timeline is presented in figures.

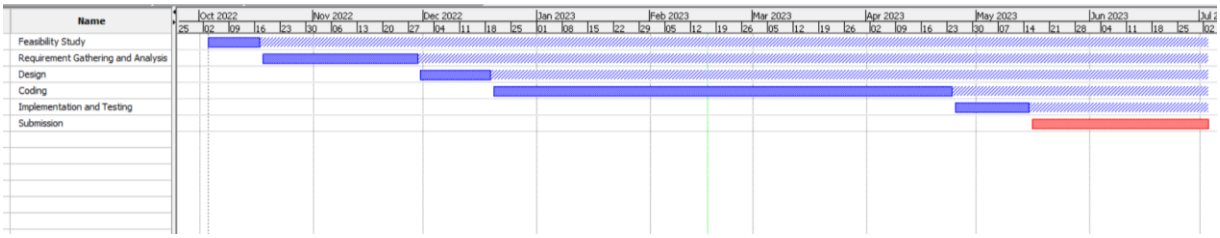

Figure 1.2: Gantt Chart

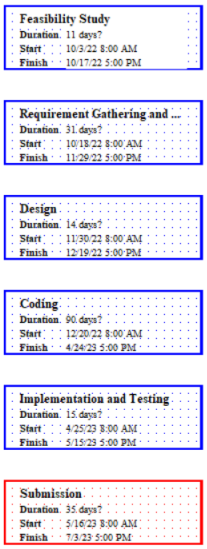

Figure 1.3: Gant Chart Deadlines

1.9. Risks and Risk mitigation

There are several common risks that can occur while developing a campus content management system, including:

1.9.1. Technical Risks

The project may face delays due to unforeseen circumstances, leading to cost overruns.

1.9.2. Technical Risks Time and Cost Overruns

There may be technical limitations or difficulties while integrating new technology and migrating from an old system.

1.9.3. Data Security Risks

A campus management system contains sensitive information of students, faculty, and staff. Thus, ensuring data security is essential.

1.9.4. User Acceptance

User acceptance is critical for the success of the system. If users are not satisfied with the new system, it may lead to low adoption rates and poor results.

1.9.5. System Performance

The system should be compatible with various devices, platforms, and operating systems.

1.9.6. System Compatibility

The system should be capable of handling the volume of users, data, and traffic on the network.

CHAPTER 2

2. Literature Review

A campus content management system (CMS) is a software system that is designed to manage, organize, and distribute digital content across a university or college campus. A review of the literature on campus content management systems reveals that the systems are increasingly being adopted by higher education institutions to improve the management and delivery of digital content. The literature suggests that campus content management systems can improve efficiency and reduce costs associated with the creation and distribution of digital content. Additionally, the systems can enable better collaboration and communication among faculty, staff, and students, leading to improved academic outcomes. Several studies have identified the key features that are important for a successful campus content management system, including ease of use, flexibility, scalability, and customization. In addition, the literature suggests that the system should be secure, with robust user authentication and access controls. Another important consideration in the design and implementation of a campus content management system is user adoption. Studies have highlighted the importance of user training and support to ensure that the system is used effectively and to its full potential. Overall, the literature on campus content management systems suggests that these systems have the potential to transform the way that universities and colleges manage and distribute digital content, leading to improved efficiency and better academic outcomes. However, the success of these systems depends on a range of factors, including the design, implementation, and user adoption.

2.1. Related Work

There are several related works to a campus content management system that create dynamic websites through code. Here are a few examples:

2.1.1. WordPress Multisite: WordPress is a popular content management system that allows for the creation of multiple websites on a single installation. With WordPress Multisite, it's possible to create a network of

sites that share a single codebase and database, making it easy to manage and update all sites simultaneously.

2.1.2. Django CMS: Django is a web framework that includes a content management system called Django CMS. It allows for the creation of dynamic websites through code and offers a user-friendly interface for content management.

2.1.3. Contentful: Contentful is a content management platform that provides a powerful API for creating and managing website content. It enables developers to build dynamic websites through code and allows for easy content collaboration and localization.

2.1.4. Squarespace: Squarespace is a website builder that offers both a user-friendly interface and the ability to create dynamic websites through code. It allows for easy customization and includes built-in tools for e-commerce, blogging, and other site features.

2.2. Reason to Develop

Developing a campus content management system as a final year project has many benefits. It provides an opportunity to apply and integrate knowledge and skills learned throughout the course of study in a practical, real-world context. Additionally, it allows for the exploration and implementation of new and emerging technologies, which is crucial for keeping up with the constantly evolving landscape of web development. Furthermore, a campus content management system is a valuable and practical tool for universities and educational institutions and developing one can provide a meaningful contribution to the academic community.

CHAPTER 3

3. System Requirements

In this chapter, we will be discussing the system requirements for the development of a campus content management system. The system requirements are an essential part of the software development process, as they define what the software is expected to do, and what hardware and software resources are needed to achieve those goals. It is important to have a clear understanding of the system requirements before starting the development process to ensure the system meets the needs of its users.

System requirements for a campus content management system can vary depending on the specific needs of the university. However, some common requirements may include a server infrastructure, a database management system, programming languages and frameworks for web development, and user authentication and authorization mechanisms. The system may also need to be scalable, secure, and able to handle high traffic volumes.

3.1. Functional Requirements

Functional requirements define the expected behavior of a system and specify what the system should do. They describe the specific features and functions that the system needs to have in order to meet the needs of the users and stakeholders. Functional requirements are a critical component of software development and are used to guide the design, development, and testing of the system. They are typically documented in a functional requirements specification and serve as a basis for measuring the success of the system.

3.1.1. Master Panel

The master panel allows administrative users to manage the content and functionality of the system.

3.1.1.1. Signup

In the master panel, users cannot sign up as master details are hard coded in the database. The master panel is used by the administrator to manage and control the system, including adding, editing, and deleting user accounts, managing content, and generating reports. It is a key functional requirement of the campus content management system.

3.1.1.2. Login

The master panel is typically accessible only to staff members who have been granted administrative privileges. These staff members will need to sign in to the system in order to access the master panel and perform various CRUD (Create, Read, Update, Delete) operations on the system's database tables. The master panel typically contains tools and features for managing the system's content, users, and other settings.

3.1.1.3. Assigning groups to user

In the campus content management system, the master panel is designed to be accessed by authorized staff members only, and its main functionality is to assign user groups. These user groups will determine what content and features are accessible to specific user types. The master panel enables staff members to manage user groups and assign permissions to individual users or groups, which can help to ensure the security and integrity of the system.

3.1.1.4. Storing, retrieving, updating deleting data

Performing CRUD operations on tables in a database is an important aspect of managing data in a campus content management system. The tables that will be used in this system include Campus, Department, EventCategory, EventCategoryContent, EventCategoryContentWebsite, MasterUser, NewsUpdates, NewsUpdatesWebsite, Society, Website. The Master Panel will allow staff members to perform CRUD operations on these tables, enabling them to efficiently manage and organize important data related to campus operations.

3.1.2. User Panel

The user panel provides access to features and information for regular users such as faculty, and other staff members. Both panels must be designed and implemented in a user-friendly and efficient manner to ensure the smooth operation of the system.

3.2. Non-Functional Requirements

Non-functional requirements in a system refer to aspects that cannot be directly linked to a specific functionality but are important for the overall performance of the system. These requirements are usually related to the system's performance, usability, reliability, security, and scalability. These requirements are essential to ensure that the system can perform well and meet the needs of the users effectively. In the case of a campus content management system, non-functional requirements might include the following:-

3.2.1. Fast response time: This refers to the system's ability to quickly respond to user requests, such as loading web pages, processing data, or performing other actions.

3.2.2. User-friendly interface: This refers to the system's ability to present information and functionality in a way that is easy to use and understand for the end-users, resulting in a positive user experience.

3.2.3. High security standards: This refers to the system's ability to maintain a high level of security to protect user data and sensitive information from unauthorized access or attacks.

3.2.4. High availability: This refers to the system's ability to remain accessible and fully functional for an extended period of time, without significant downtime or interruptions.

Ability to handle many users simultaneously: This refers to the system's ability to handle a high volume of traffic and user requests without slowing down or crashing.

3.3. Use Case Diagram

A use case diagram is a visual representation of the interactions between actors (users or systems) and a system under consideration. It helps to identify the functionalities that the system needs to perform, and the actors involved in those functionalities. Use case diagrams are an important tool for software development as they provide a clear understanding of the system's requirements and help to communicate those requirements to the development team.

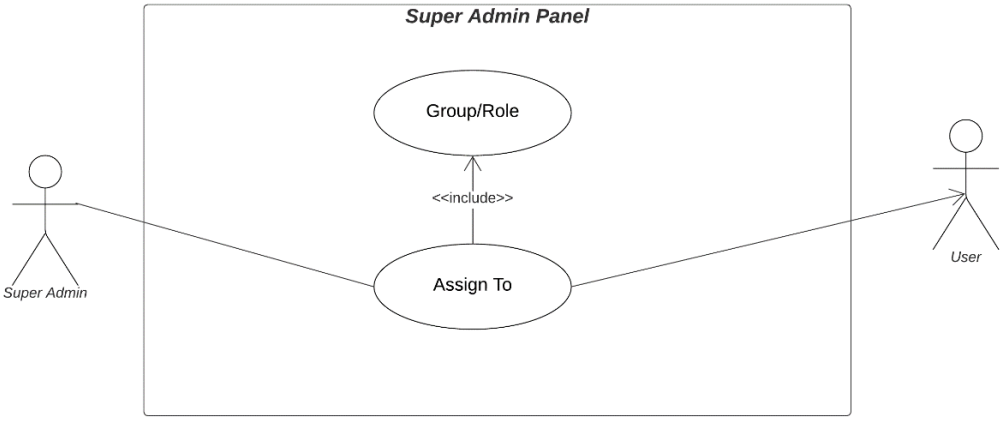

Figure 3.1: Assigning roles to user

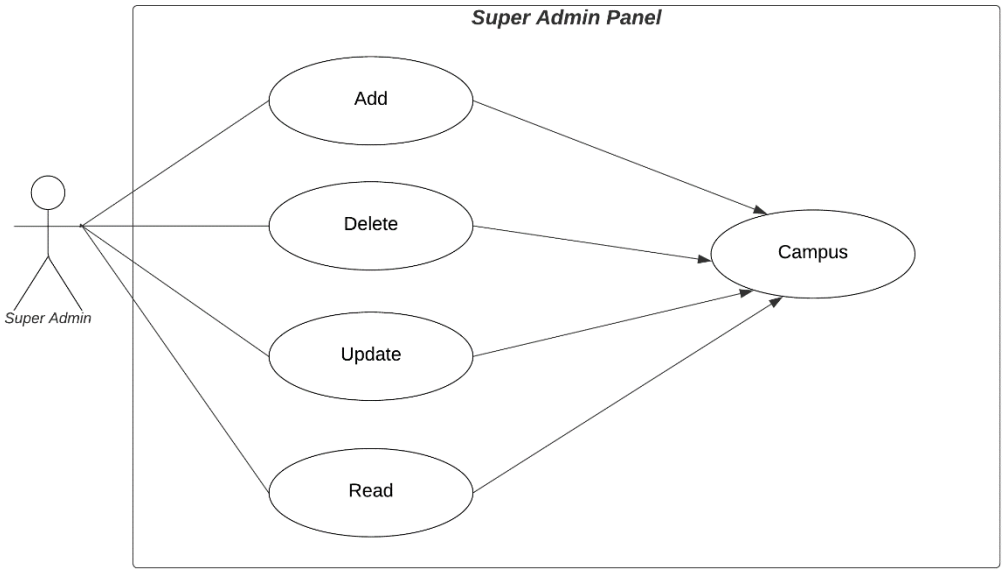

Figure 3.2: Campus Use Case Diagram

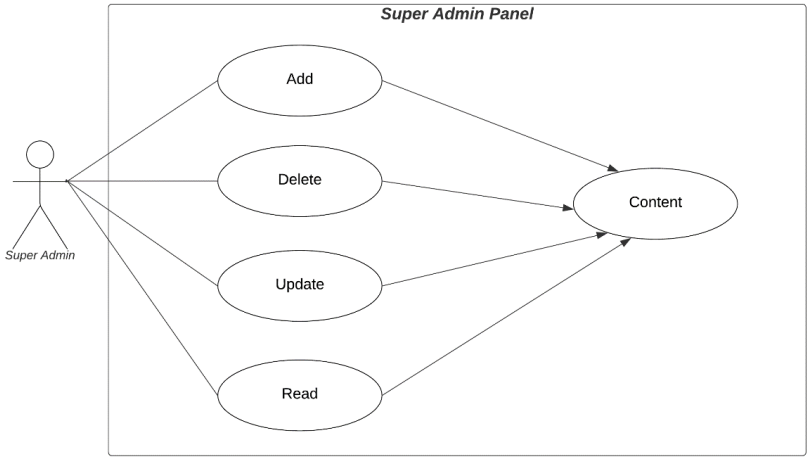

Figure 3.3: Content Use Case Diagram

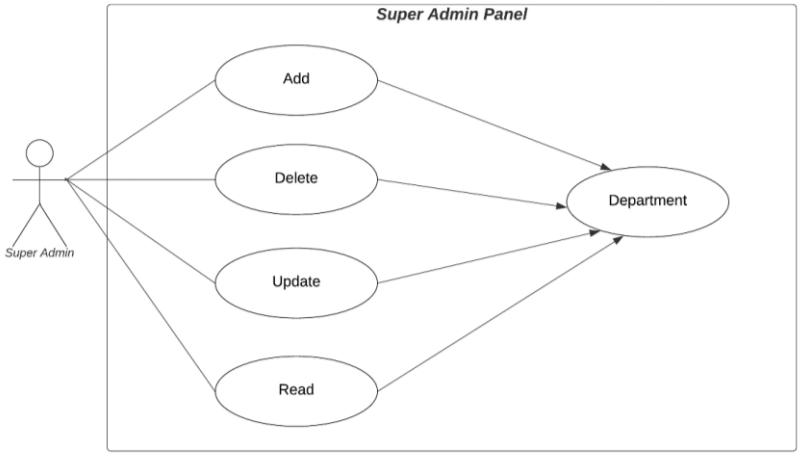

Figure 3.4: Department Use Case Diagram

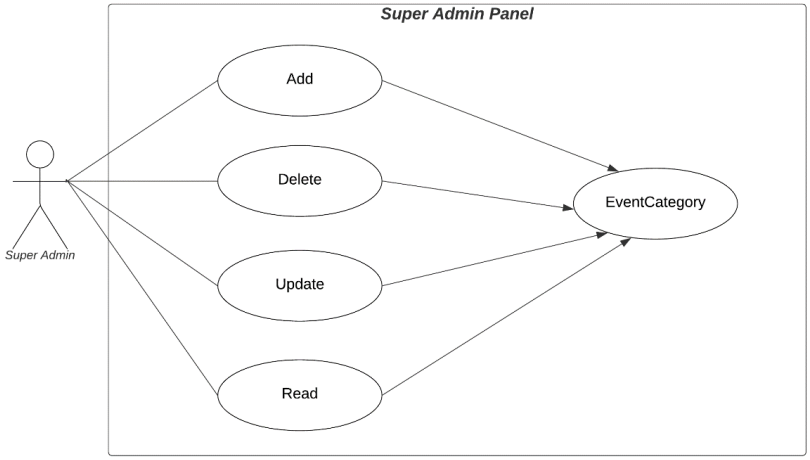

Figure 3.5: Event Category Use Case Diagram

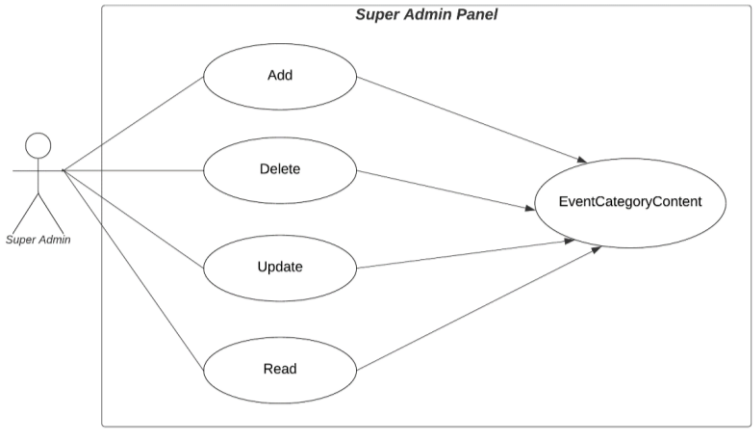

Figure 3.6: Event Category Content Use Case Diagram

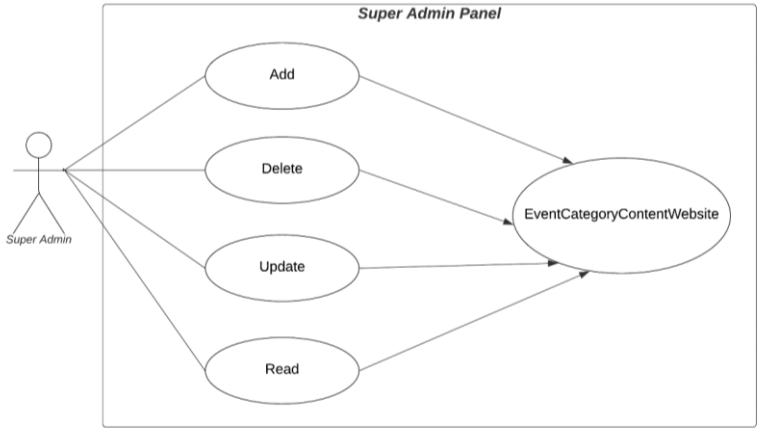

Figure 3.7: Event Category Content Use Case Diagram Website

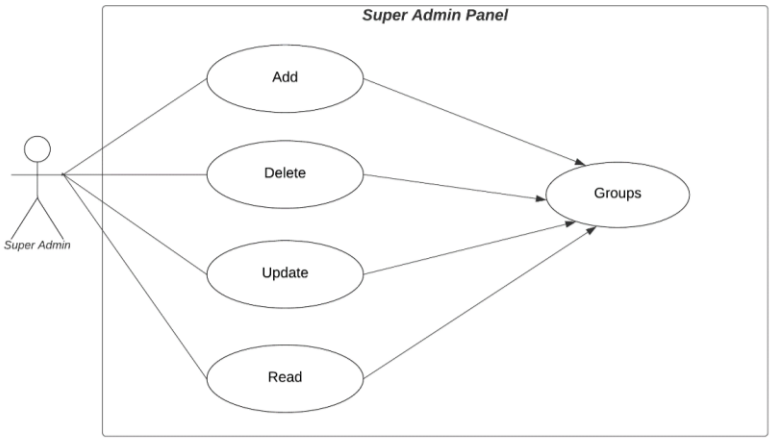

Figure 3.8: Groups Use Case Diagram

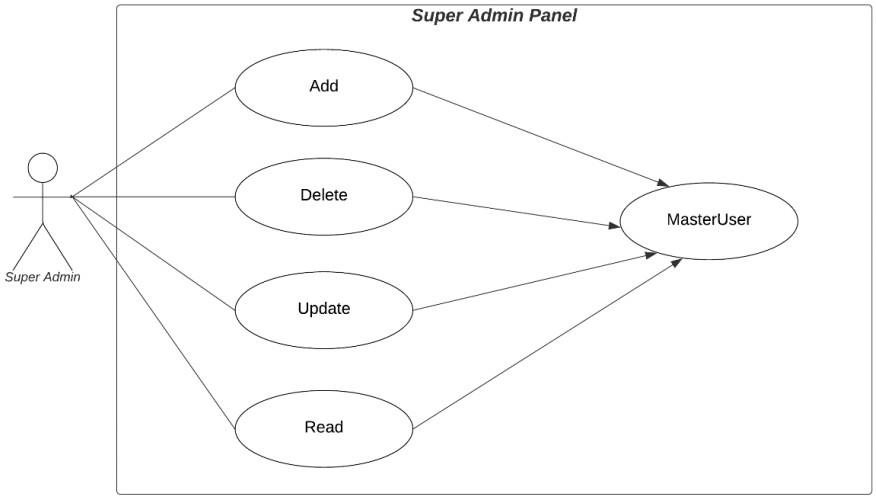

Figure 3.9: Master User Use Case Diagram

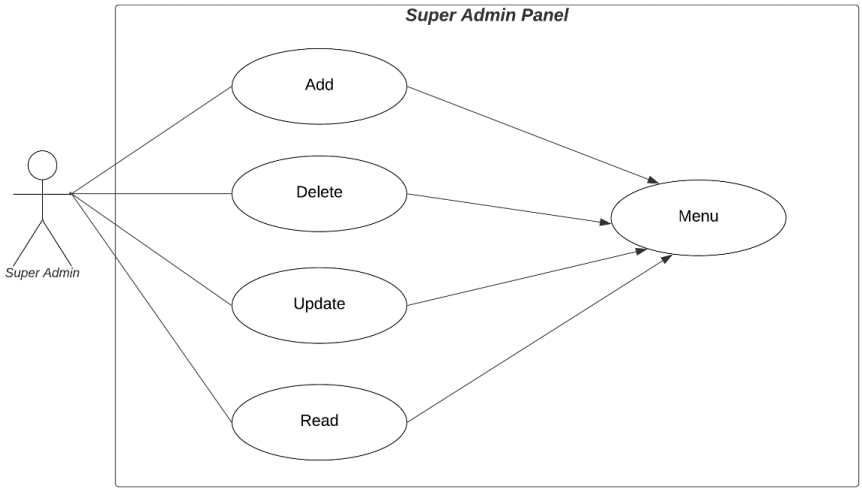

Figure 3.10: Menu Use Case Diagram

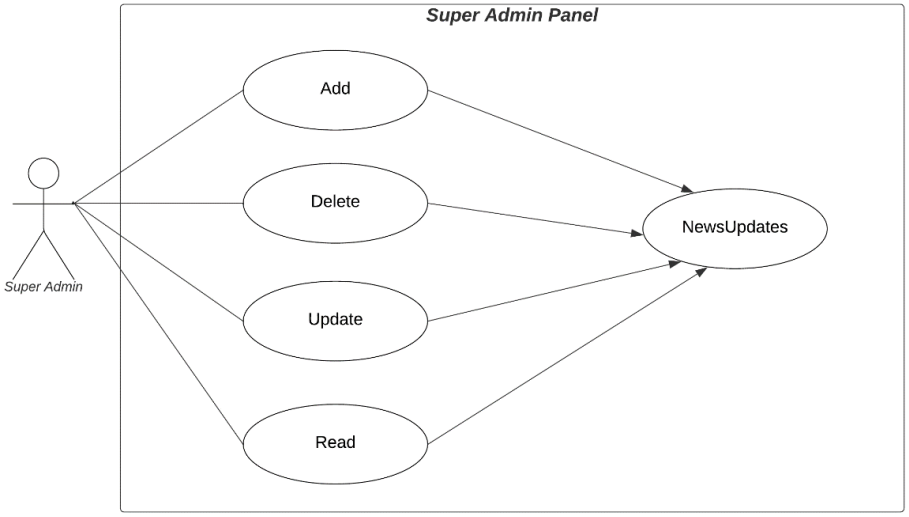

Figure 3.11: News Updates Use Case Diagram

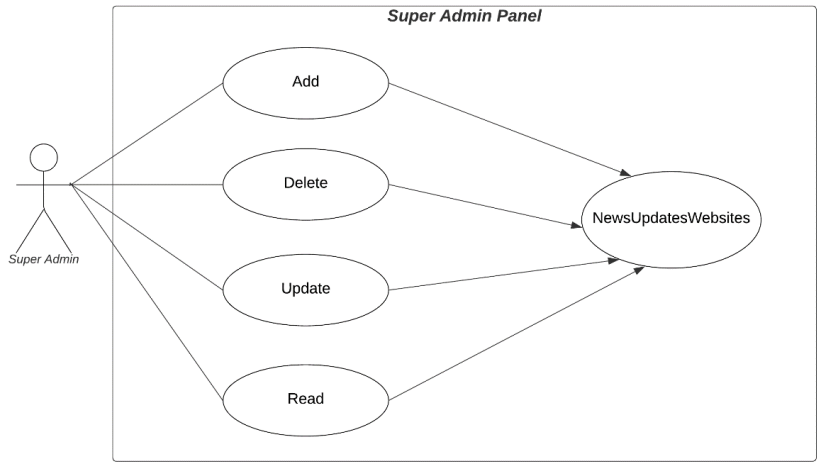

Figure 3.12: News Updates Website Use Case Diagram

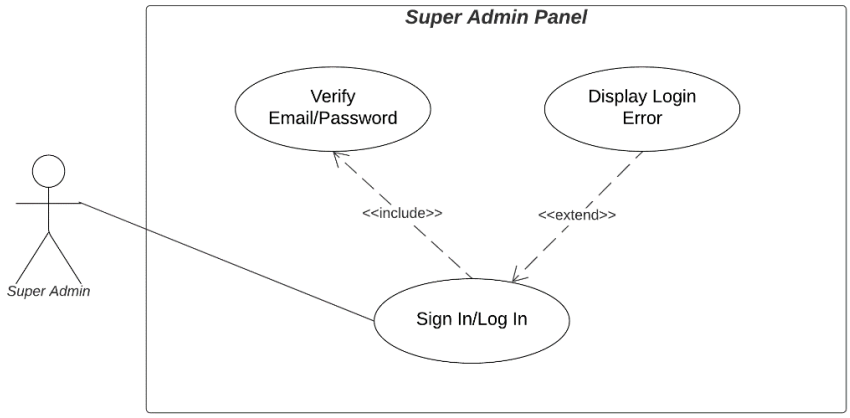

Figure 3.13: Login Use Case Diagram

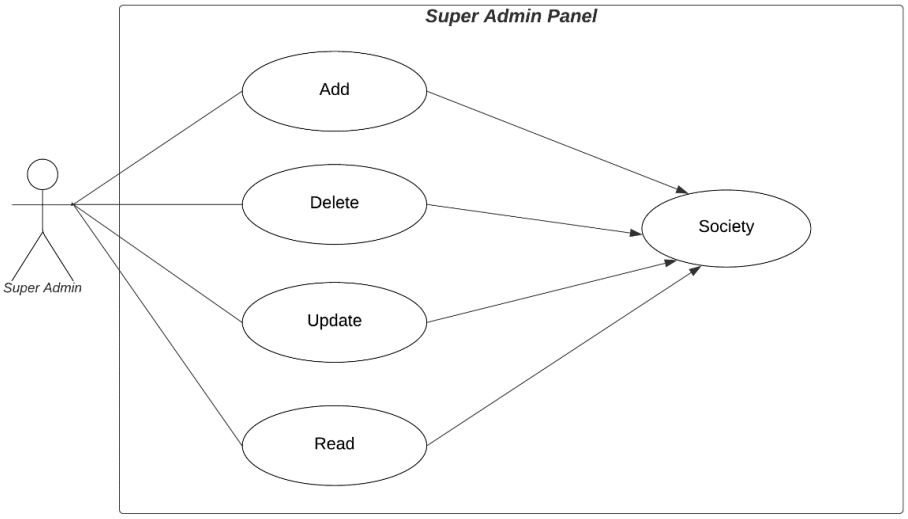

Figure 3.14: Society Use Case Diagram

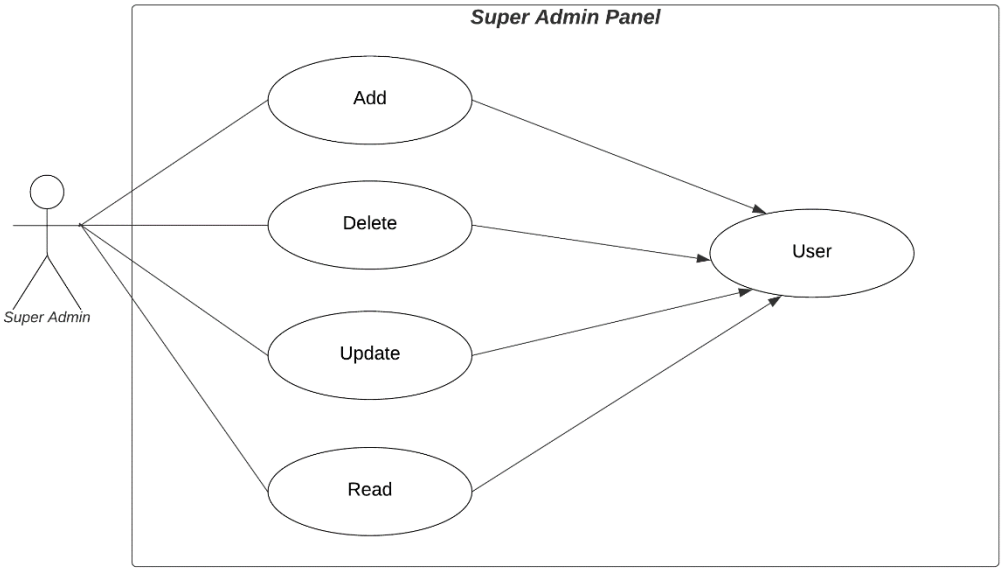

Figure 3.15: User Use Case Diagram

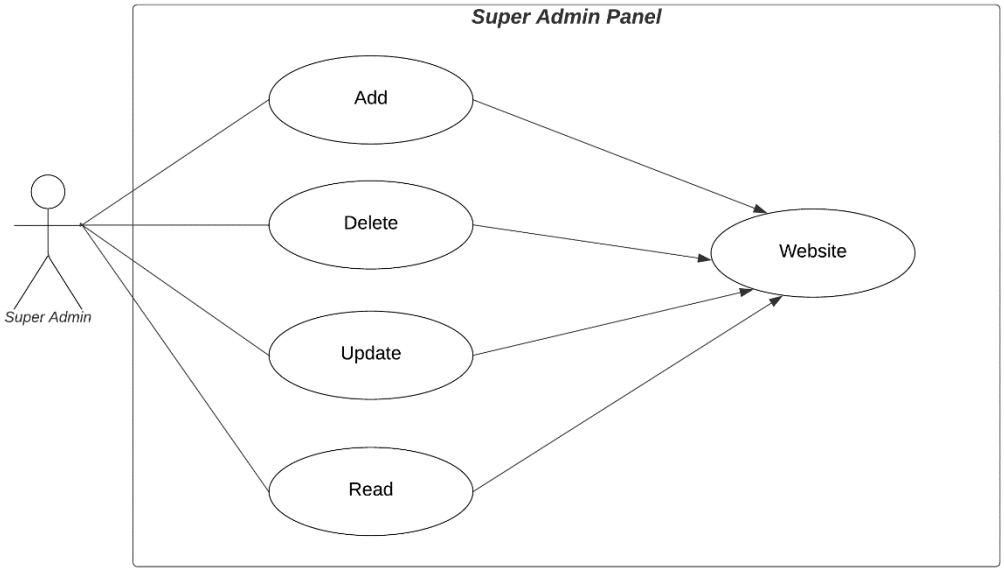

Figure 3.16: Website Use Case Diagram

3.4. Use Case Description

Use case descriptions provide a detailed explanation of the interactions between users and the system. They describe the specific tasks that a user can perform in the system and the responses that the system provides in return. These descriptions help to identify the functional requirements of the system and ensure that all user needs are addressed.

3.4.1. Use Case Campus

Table 3.1: Add new campus

ID	1
Name	Add New Campus
Actor	User
Description	User will be able to create a new campus
Pre-Condition	Campus must not exist already
Post-Condition	Create a campus

3.4.2. Use Case Department

Table 3.2: Add new department

ID	2
Name	Add New Department
Actor	User
Description	User will be able to create a new department
Pre-Condition	Department must not exist already
Post-Condition	Create a department

3.4.3. Use Case Event Category

Table 3.3: Create new event

ID	3
Name	Create New Event
Actor	User
Description	User will be able to create an event
Pre-Condition	Event must not exist already
Post-Condition	Create an event

3.4.4. Use Case Event Category Content

Table 3.4: Add event content

ID	4
Name	Add Event Content
Actor	User
Description	User will be able to add content to existing event
Pre-Condition	Event must exist
Post-Condition	Add content to that selected event

3.4.5. Use Case Master User

Table 3.5: Create user

ID	5
Name	Create User
Actor	Master User
Description	Master User will be able to create a new user
Pre-Condition	User must not exist already
Post-Condition	Create a new user

3.4.6. Use Case News Update

Table 3.6: Create new news/updates

ID	6
Name	Create News/Update
Actor	User
Description	User will be able to create news/update
Pre-Condition	News or update must not exist already
Post-Condition	Create a new news or update

3.4.7. Use Case Society

Table 3.7: Create new society

ID	7
Name	Create Society
Actor	User
Description	User will be able to create a new society
Pre-Condition	Society must not exist already
Post-Condition	Create a new society

3.4.8. Use Case Website

Table 3.8: Create new website

ID	8
Name	Create Website
Actor	User
Description	User will be able to create a new website
Pre-Condition	Website must not exist already
Post-Condition	Create a new website

CHAPTER 4

4. Methodology

Methodology refers to the overall approach or strategy used to plan and execute a project. It encompasses a set of principles, processes, techniques, and tools that guide the project team through each stage of the project lifecycle. The methodology selected for a project can have a significant impact on the success of the project, as it defines how tasks will be organized, how decisions will be made, how resources will be allocated, and how risks will be managed. There are many different methodologies to choose from, each with its own strengths and weaknesses, and the selection of the most appropriate methodology depends on the nature and complexity of the project, as well as the organization's culture and project management capabilities.

4.1. Project Planning

Project planning is the process of creating a roadmap that outlines the tasks, resources, and timeline needed to complete a project successfully. It involves identifying the project's objectives, determining the tasks required to achieve those objectives, assigning tasks to team members, estimating the time and cost required to complete each task, and creating a timeline for the project. Effective project planning is essential to ensure that the project is completed on time, within budget, and meets the project's objectives.

4.2. Methodology for software development

There are various methodologies for software development such as Waterfall, Agile, Scrum, Spiral, etc. Each methodology has its own advantages and disadvantages, and it is important to choose the most suitable methodology for the specific project requirements. The chosen methodology guides the development process, including the planning, design, implementation, testing, and deployment phases of the software.

4.2.1. Waterfall

The Waterfall methodology is a linear and sequential approach to software development. It consists of a series of distinct phases, where each phase must be completed before moving on to the next one. The phases include requirements gathering, design, implementation, testing, and maintenance. The waterfall model is often used in projects where the requirements are well-understood, and the final product is expected to be delivered as a complete package. However, it has some limitations in terms of flexibility and adaptability to changing requirements during the development process.

4.2.2. Agile

Agile is a methodology for software development that emphasizes flexibility and collaboration between cross-functional teams. It involves a iterative and incremental approach to development, with a focus on delivering working software in short sprints. Agile methodology prioritizes customer satisfaction, responding to change, and continuous improvement throughout the development process. It is often used in software projects where requirements and priorities are subject to change or are not fully defined at the outset.

4.2.3. Spiral

The Spiral model is a software development methodology that combines the iterative nature of the Agile model with the systematic, controlled aspects of the Waterfall model. It emphasizes risk management and focuses on identifying and addressing potential problems early on in the development process. The Spiral model consists of several phases, including planning, risk analysis, engineering, and evaluation. The process repeats in a spiral manner, with each iteration building on the previous one, until the software is complete.

4.2.4. Scrum

Scrum is an agile framework for software development that focuses on iterative and incremental delivery of a product. It emphasizes collaboration, flexibility, and continuous improvement throughout the development process. Scrum teams work

in short sprints, typically 1-4 weeks, and have daily meetings to review progress and adjust their plans accordingly. The framework includes roles such as the product owner, development team, and scrum master, and specific artifacts such as the product backlog, sprint backlog, and burn-down charts.

4.3. Selected methodology

I have selected Agile methodology for your project development. This methodology emphasizes on flexibility and continuous improvement, making it ideal for your project needs.

4.4. Reason for selecting agile methodology

Agile methodology is often chosen for software development projects because it is flexible and adaptive to changing requirements. It allows for collaboration and continuous feedback between the development team and stakeholders, resulting in a more efficient and effective development process. Additionally, it encourages an iterative approach to development, allowing for the product to evolve and improve over time. These factors make it a good fit for the development of a campus content management system, where requirements may change over time and stakeholder feedback is important.

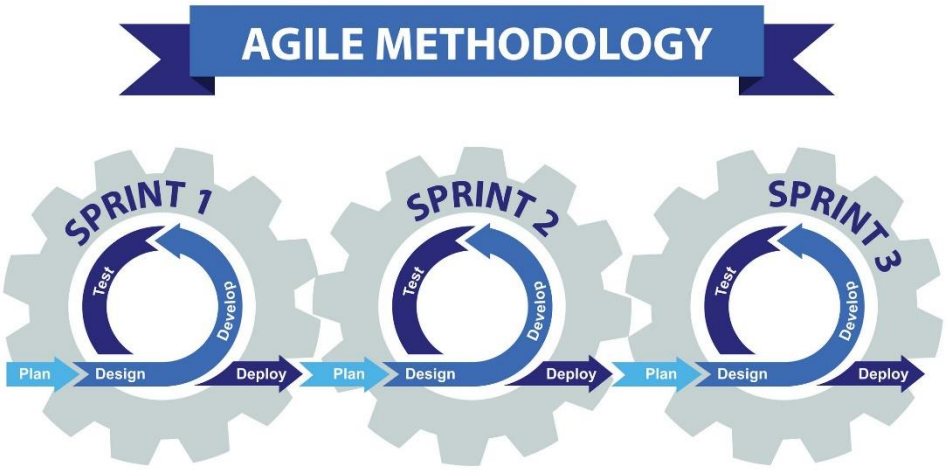

Figure 4.1: Agile Methodology

CHAPTER 5

5. System Architecture

System architecture refers to the structural design of the entire system, including its components, modules, and their relationships. System diagrams, on the other hand, are graphical representations of the system architecture that show how the components and modules interact with each other to perform specific functions.

5.1. Architecture

In general, architecture refers to the fundamental organization of a system and its components. It involves designing and planning the overall structure, behavior, and functionality of a system to ensure that it meets its intended purpose effectively and efficiently. Architecture can refer to a wide range of systems, from physical structures like buildings to software applications and computer systems.

The system architecture of a campus content management system typically consists of four main components: the presentation layer, the application layer, the data management layer, and the database layer.

The presentation layer is responsible for rendering the user interface and managing user interactions. The application layer handles the business logic and processes user requests. The data management layer manages the data and ensures its integrity. The database layer stores the data in a structured format and provides access to it through a database management system.

To illustrate the system architecture, diagrams such as component diagrams, deployment diagrams, and sequence diagrams can be used.

5.2. Activity Diagram

An activity diagram is a visual representation of the flow of actions, activities, and tasks that need to be completed in a particular process or system. It shows the sequence of activities and their relationships and can be used to model complex workflows and business processes.

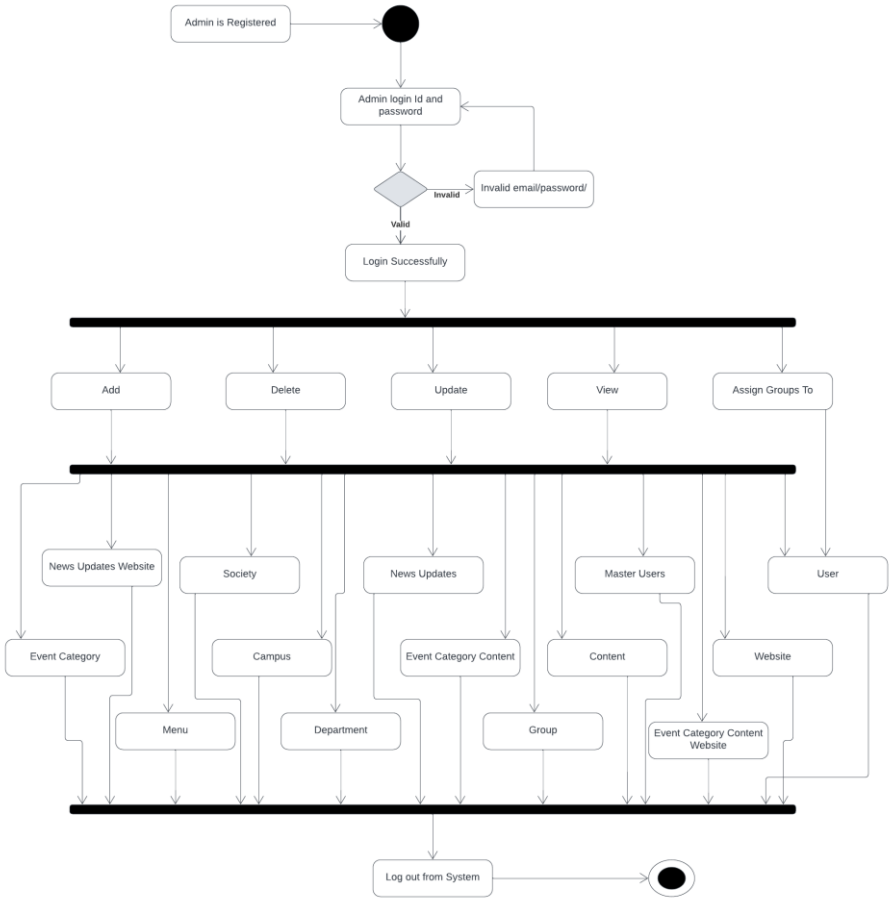

Figure 5.1: Activity Diagram

5.3. Sequence Diagram

A sequence diagram is a type of interaction diagram that shows the interactions and messages exchanged between objects or components of a system in a time-ordered sequence. It depicts the flow of control between the different entities and objects involved in a scenario, showing the order in which messages are exchanged and the lifetimes of the participating objects. Sequence diagrams are often used in software design and development to illustrate complex scenarios and system interactions.

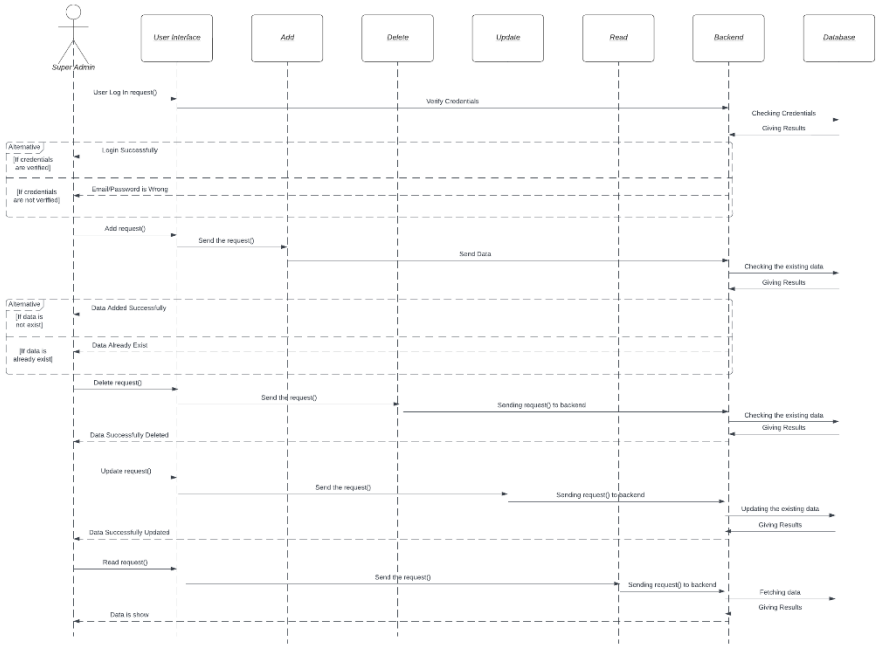

Figure 5.2: Sequence Diagram

5.4. Deployment Diagram

A deployment diagram is a type of diagram in UML that shows the physical deployment of software components to hardware nodes and the relationships between them. It is used to describe the topology of the hardware and software components of a system, and to illustrate how they interact with each other in a distributed environment. The deployment diagram shows the deployment of software artifacts to nodes, the associations between them, and the physical distribution of nodes in the network. It is useful in understanding how the system will be deployed and run in a production environment.

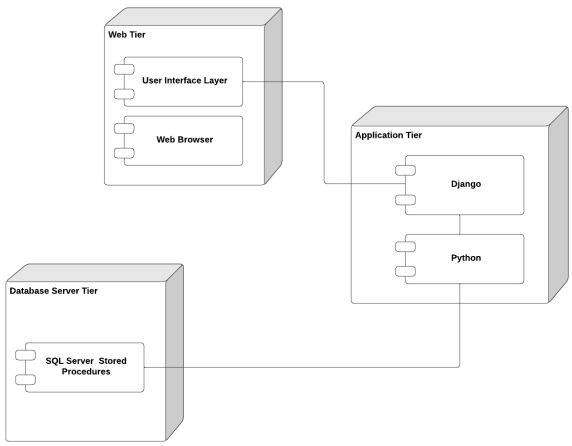

Figure 5.3: Deployment Diagram

CHAPTER 6

6. System Implementation

System implementation refers to the process of putting the software system design into action by writing the actual code and building the software system. It involves various tasks such as coding, testing, debugging, and deployment. This phase requires careful planning and execution to ensure that the system is implemented successfully and meets the needs of the users.

6.1. System Tools and Technology

- Python
- Django
- Microsoft SQL Server

6.2. Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It is widely used in various domains such as web development, data analysis, scientific computing, and artificial intelligence.

6.3. Django

Django is a high-level Python web framework that follows the Model-View-Controller (MVC) architectural pattern. It provides a simple and fast way to build web applications with an emphasis on reusability, modularity, and rapid development. Django's main features include an ORM, URL routing, template engine, middleware, and built-in authentication and authorization systems. It is widely used by developers to build complex web applications, including content management systems, social networks, and e-commerce sites.

6.4. Microsoft SQL Server

Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is used to store and retrieve data as requested by other software applications. It supports SQL (Structured Query Language), which is a language used to manage and manipulate relational databases. SQL Server can be used for a variety of applications, from small desktop applications to large enterprise-class databases.

6.5. Database Diagram/Design

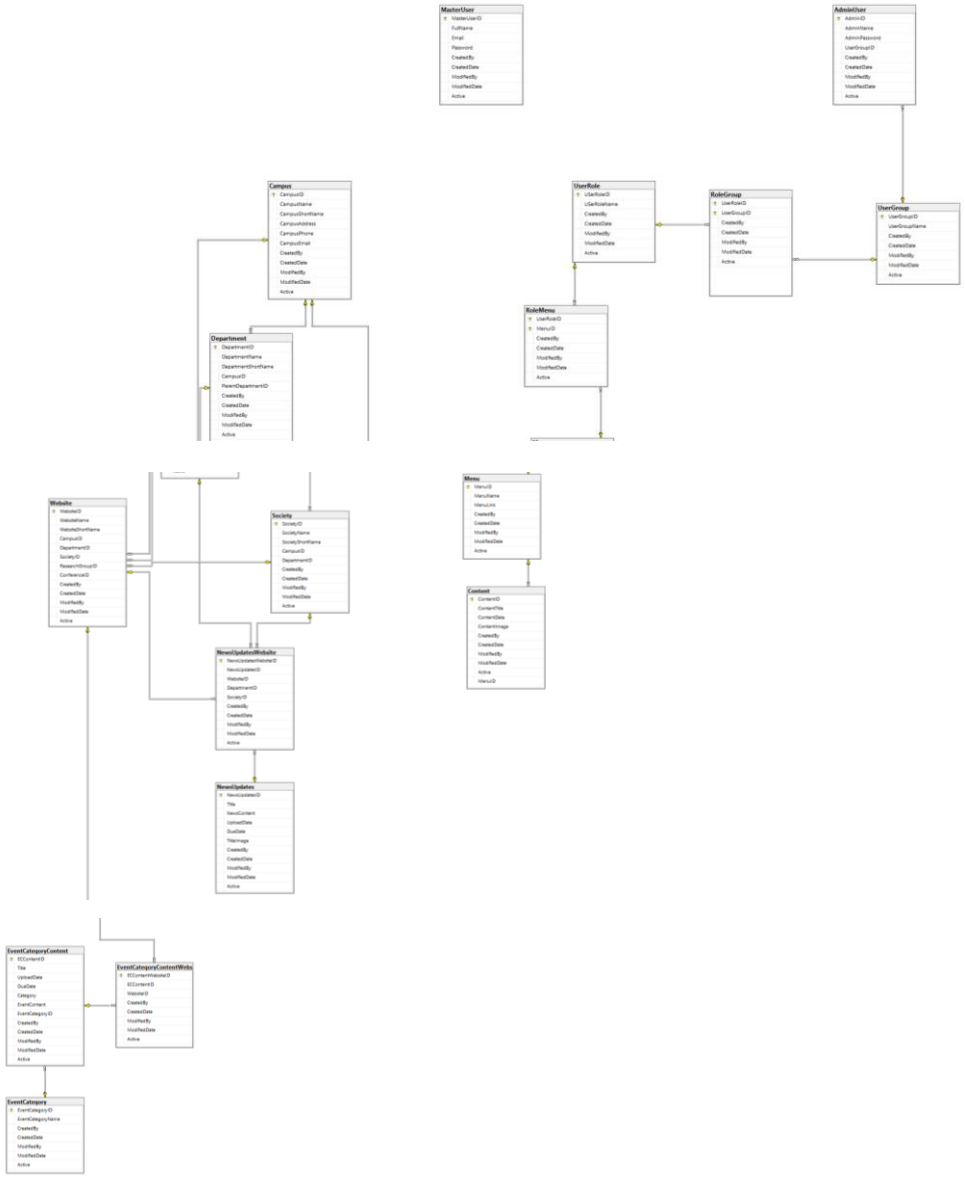

Figure 6.1: Database Design

6.6. Class Diagram

Figure 6.2: Class Diagram