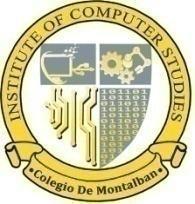
**COLEGIO DE MONTALBAN**

*Kasiglahan Village, San Jose, Montalban, Rizal*

**INSTITUTE OF COMPUTER STUDIES**

Department of Computer Engineering

**Learning Management**

**System for Colegio de Montalban portal**

**BSCPE 2D**

**AGCAOILI, PRINCESS MAE**

**CONSTANTINO, VAN JOSEPH**

**DITA, ERICA MAE**

**ESCA**[**Ñ**](https://filipiknow.net/enye/)**ON, ALEXANDER SHEEN**

## [TABLE OF CONTENTS]

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## PLANNING

**Introduction**

**Brief Overview of the Project**

This project involves developing a comprehensive Learning Management System (LMS) for Colegio de Montalban using Python Django and PostgreSQL. An LMS is an integral component of modern educational environments and corporate training programs. It is a sophisticated software application or web-based technology designed to plan, implement, and assess a specific learning process. An LMS provides a centralized, cohesive platform where educators and administrators can deliver content, track participation, and assess performance. It supports a variety of learning methodologies, including both online and blended learning, and serves as a powerful tool to enhance the educational experience.

**Background and Context**

Traditional training programs and educational environments frequently struggle with issues including rigidity, accessibility issues, and tracking and evaluating student progress. The emergence of digital technology presents a chance to get over these challenges. Institutions may provide a more effective and engaging learning environment that is not limited by time zones or location by putting an LMS into place. Additionally, this technology offers insightful data analytics to improve the educational process.

**Objectives**

● Enhance communication and collaboration between educators and students.

● Streamline assignment creation, submission, and grading processes.

● Provide real-time feedback and performance tracking for students.

● Promote personalized learning experiences tailored to individual student needs.

● Ensure data privacy and security for all users.

● Support seamless integration with existing educational tools and platforms.

**Scope**

* **User Management:** 
  + **Administrators:** Full control over the system, managing user roles, permissions, and configurations.
  + **Educators/Trainers:** Tools to create, manage, and deliver courses, materials, and assessments.
  + **Students:** Access to course materials, assignment submissions, discussions, and progress tracking.
  + **Parents/Guardians:** Limited access to monitor student progress and communicate with educators.
* **Course Management:**
  + **Course Creation:** Features for designing and setting up courses with various multimedia content.
  + **Course Enrollment:** Automated and manual enrollment options, including prerequisite management.
  + **Content Delivery:** Support for videos, documents, quizzes, and interactive learning modules.
* **Assessment and Evaluation:**
  + **Assignment Management:** Create, manage, and grade assignments with flexible submission options.
  + **Gradebook:** Comprehensive tracking and management of student grades and performance analytics.
* **Security and Compliance:**
  + **Data Security:** Robust measures to protect user data and ensure privacy.
  + **Access Control:** Role-based access control to manage permissions.

**Limitations**

* **Communication Tools:** Forums, messaging, and announcements.
* **Analytics and Reporting:** Tracking student progress and performance metrics.
* **Chat Features:** Real-time messaging between users.
* **Live Meetings:** Integrated video conferencing for virtual classes and meetings.
* **User Profile Search:** Ability to search and view profiles of other users.
* **Archiving Past Classes:** Storage and access to past classes that users have been enrolled in.
* **Offline Access:** The LMS is designed primarily for online access. Limited offline functionalities (e.g., downloadable content) may be provided,

**Development Team Roles**

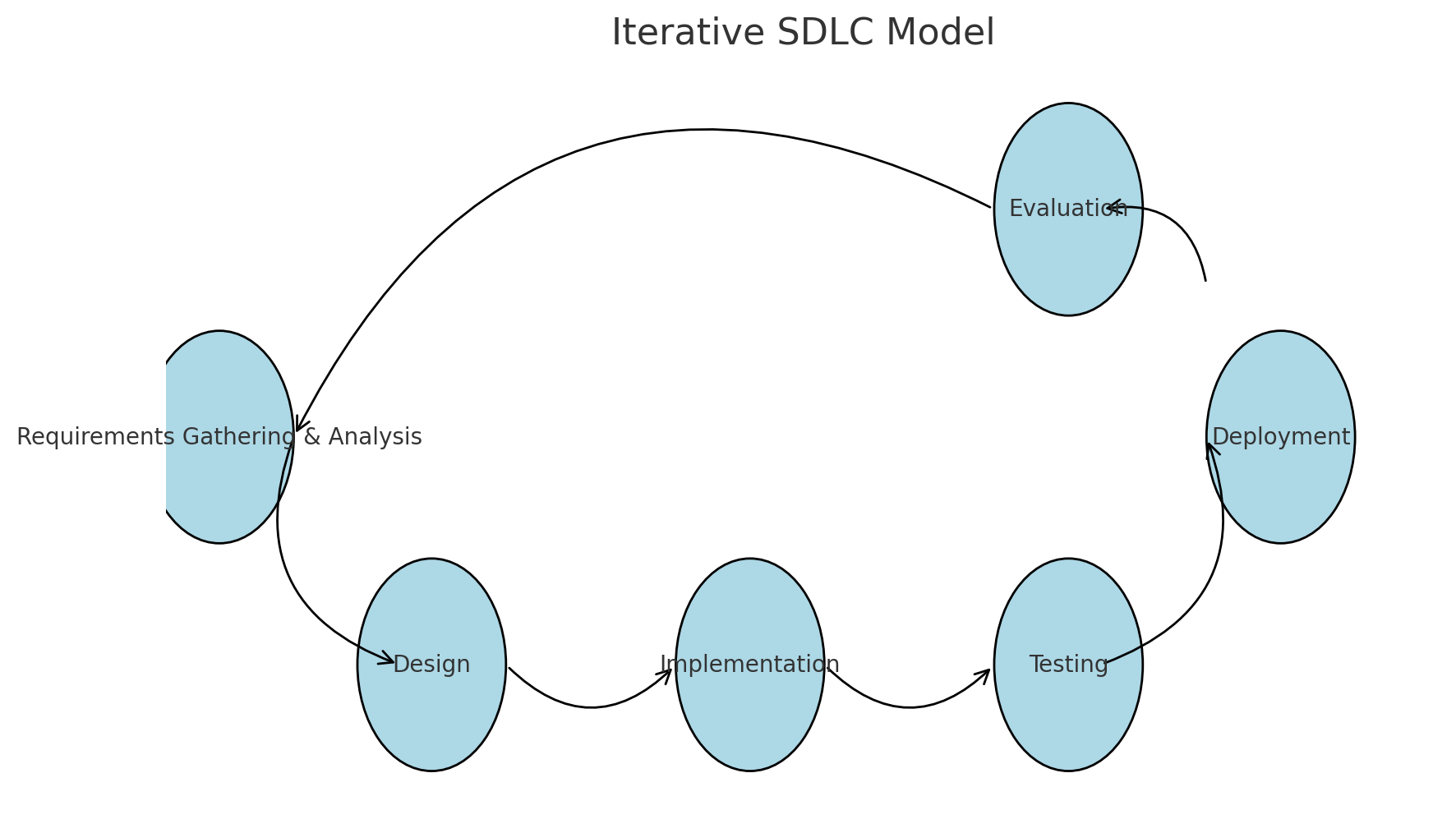
*Note: you can use photos or diagrams here.*

* Escañon, Alexander Sheen C. - **Project Manager:** Oversees the project, ensuring it stays on schedule and meets requirements.
* Constantino, Van Joseph - **Lead Developer:** Responsible for the overall system architecture and coding standards.
* Escañon, Alexander Sheen C. - **Frontend Developer:** Designs and implements the user interface.
* Ungay, Jaeny Anne M. - **User Interface Designer:** Design the UI of login and dashboard
* Constantino, Van Joseph - **Backend Developer:** Manages the server-side logic and database integration.
* Agcaoili, Princess Mae N. - **QA Engineer:** Conducts testing to ensure the system functions correctly.
* Dita, Erica Mae P. - **Technical Writer:** Prepares and maintains project documentation.

**Software Development Cycle**

*Note: include a photo or diagram of your SDLS here*

Chosen Development Methodology: Iterative



Stages, Milestones, and Timeline:

* Planning Phase: 2 weeks
  + Initial project setup and requirements gathering.
* Design Phase: 3 weeks
  + Creating design documents, including wireframes, flowcharts, and diagrams.
* Implementation Phase: 4 weeks
  + Developing the system components.
* Testing Phase: 1 week
  + Conducting system testing and bug fixing.
* Deployment Phase: 1 week
  + Deploying the system to the production environment.

**Gantt Chart**

It requires more communication and collaboration among

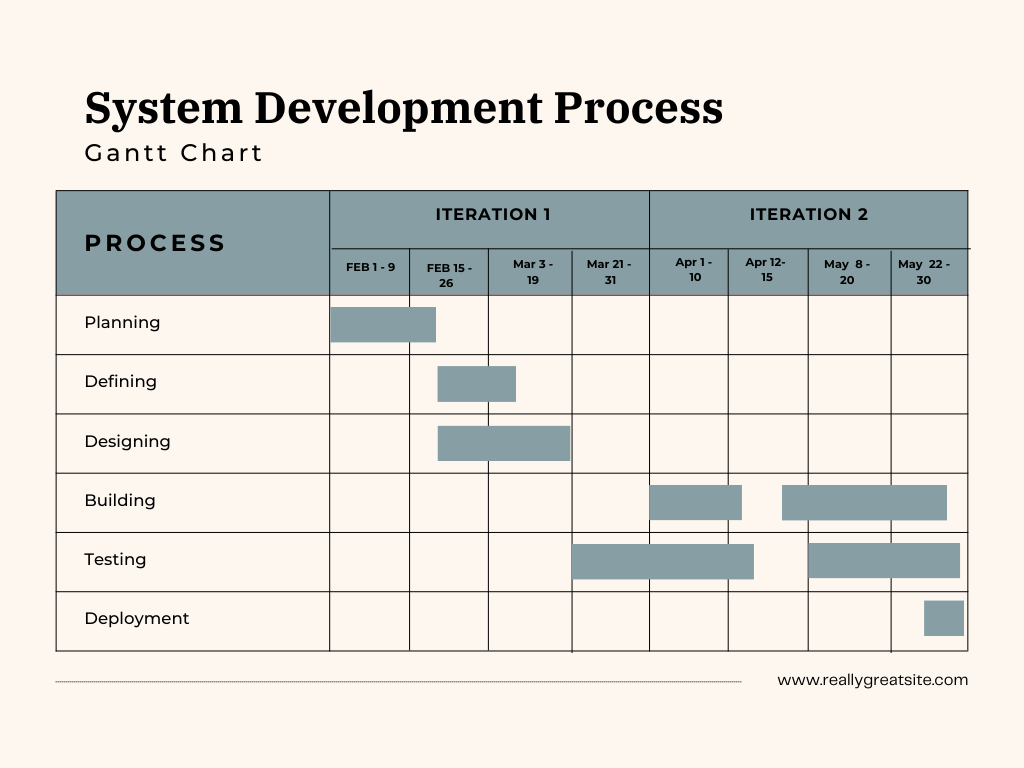
the team and flexible that can easily accommodate changes in

requirements and if we have unexpected ideas that we can add to

our project. It's also Improved the quality of the project because we

can frequent testing and feedback allow developers to identify and

fix issues more quickly, leading to a higher-quality final product.



(Figure 1)

Figure 1 presents the Gantt Chart for the Learning Management

System for Colegio de Montalban portal project timeline at Colegio de Montalban Learning. The Gantt Chart visually represents the project schedule, highlighting key phases, milestones, and the duration of each task involved in the development and deployment of the LMS.

## REQUIREMENT GATHERING

**Technique**

**Brainstorming Sessions**

Brainstorming sessions were conducted with key stakeholders, including Instructors and students. The goal was to generate ideas and concepts that address the most pressing needs of the Learning system. Key ideas and concepts generated during these sessions include:

* Efficient user authentication to ensure secure access to the system
  + Inputs: User credentials
  + Processes: Validate user credentials, Grant access
  + Outputs: Access granted or denied.
* Course Management:.
  + Inputs: Course materials, User requests (enroll, drop)
  + Processes: Enroll/drop users from courses, Upload course materials
  + Outputs: Updated course enrollment, Uploaded materials
* Course Interaction:
  + Inputs: User interactions (view lecture, submit assignment), Course materials
  + Processes: Display course content, Evaluate submissions
  + Outputs: Viewed lectures, Graded assignments

**Interviews**

Interviews were conducted with professors and students to gather detailed insights into the current system's pain points and potential improvements. The interview process involved:

* One-on-one sessions with professors and students discuss their daily workflows and challenges.
* Identifying the need for features and additional functions with the system management.
* Understanding the specific requirements for user management, including registration and authentication.

**Surveys**

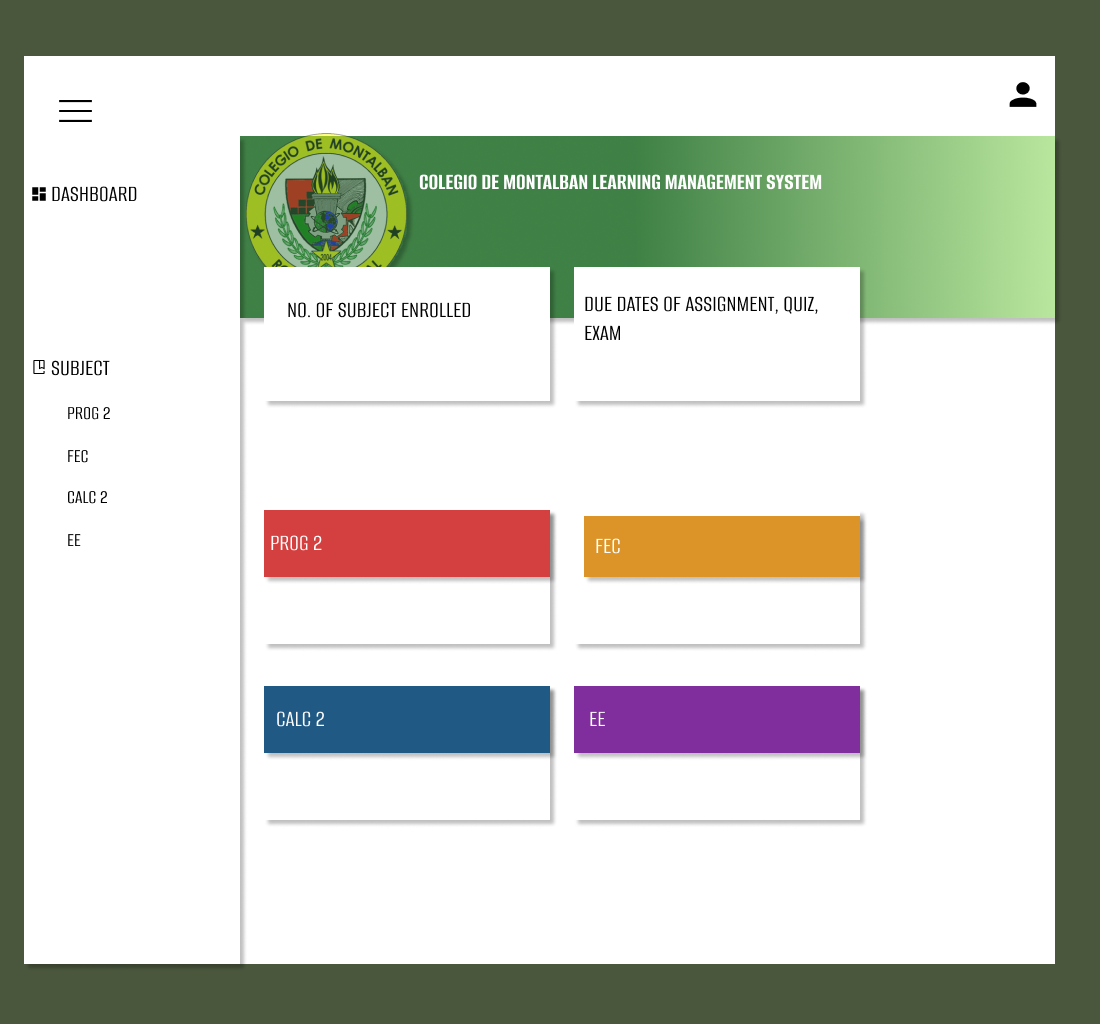
Surveys were distributed to professors using the Learning to gather feedback on desired features and user experience. The survey design included questions about:

* Can be created by group assigning activities and also can comment from it.

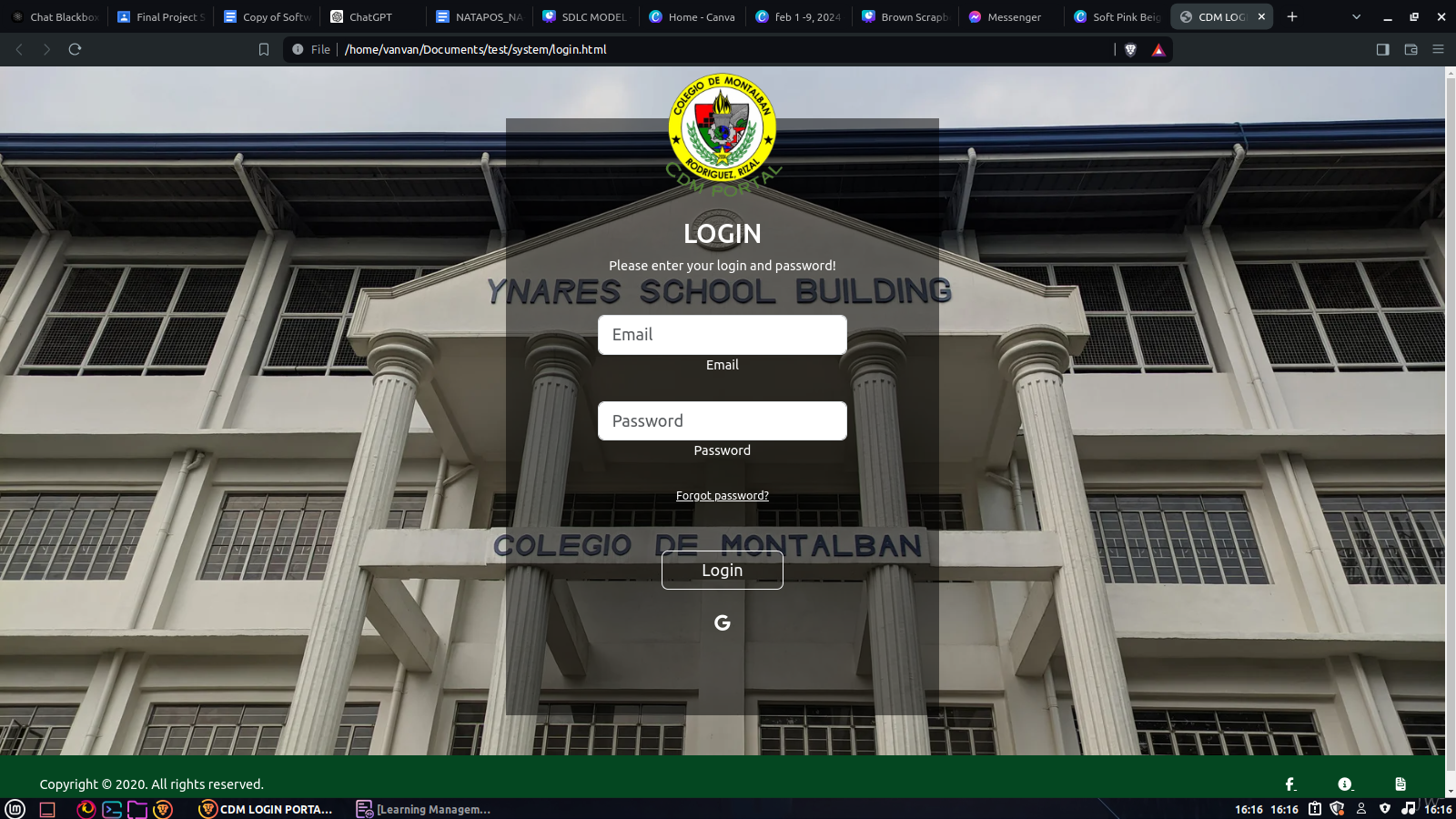
**Prototyping**

Prototyping involved creating initial versions of the user interface to gather feedback from users and stakeholders. The process included:

UI Dashboard:



UI Login:



**Document Analysis**

Document analysis was performed on existing Learning management documentation to identify essential features and workflow inefficiencies. This included:

* Reviewing current processes for courses and assignment management.
* Analyzing existing reports and user records to understand data management requirements.
* Identifying gaps and areas for improvement in the current system.

**Functional Requirements**

* User authentication and role-based access control to ensure secure access to resources.
* Course creation and enrollment management to facilitate organization and administration of classes.
* Assignment creation, submission, and grading functionalities with support for various file formats.
* Calendar integration for assignments, and deadlines.
* Gradebook and progress tracking tools to monitor student performance and learning outcomes.
* Mobile compatibility for access on smartphones and tablets, ensuring flexibility and accessibility.

**Non-functional Requirements**

* **Performance:** Fast response times, minimal page load times, and ability to handle large numbers of concurrent users and courses.
* **Security:** Robust security measures, secure authentication and authorization, data encryption, and access controls.
* **Scalability:** Designed to scale horizontally and vertically to accommodate growth.

## 

## DESIGN

[Wireframe Placeholder]

(Figure 2)

**Figure 2**: illustrates the wireframe for the main user interface of the Learning Management System (LMS) for Colegio de Montalban Learning. The wireframe serves as a blueprint for the design and layout of the LMS, highlighting key components and their placement within the interface.

Key Components:

* Header Section:
  + Located at the top of the interface.
  + Includes the Learning logo, navigation menu (Home and Account), and a search bar for quick access to book searches.
* Navigation Menu:
  + Positioned below the header.
  + Contains links to the main sections of the LMS: Logout..
* Sidebar:
  + Displayed on the left side of the interface.
  + Provides additional navigation options such as Courses and Profile
  + Includes quick links for subjects.
* Main Content Area:
  + Occupies the central portion of the interface.
  + Initially displays a welcome message and overview of the Learning's services.
  + Dynamically updates to show detailed content based on user interactions, such as subject details.
* Featured Subject Section:
  + Positioned prominently in the main content area.
  + Display the counts of enrolled subject of students and instructor
* Footer Section:
  + Located at the bottom of the interface.
  + Contains links to Learning policies, contact information, and social media channels.
  + Provides quick access to user support and feedback forms.

Interactive Elements:

* Subject Card:
  + Display all contents of enrolled subject
* Selected Subject :
  + Clickable icon that opens the contents of subjects such as: assignments.
  + Create, Read, Update, Submit, Attach file, and Delete Assignment.
* Navigation Links:
  + Enable easy access to different sections of the LMS.
  + Redirect to Selected subjects

User Experience Considerations:

* Intuitive Layout:
  + Designed to be user-friendly with a clean and organized structure.
  + Ensures that essential features are easily accessible and prominently displayed.
* Responsive Design:
  + Wireframe accounts for various screen sizes and devices, ensuring a seamless experience across desktops, tablets, and smartphones.
* Accessibility:
  + Incorporates accessibility features such as keyboard navigation and screen reader compatibility to ensure usability for all users, including those with disabilities.

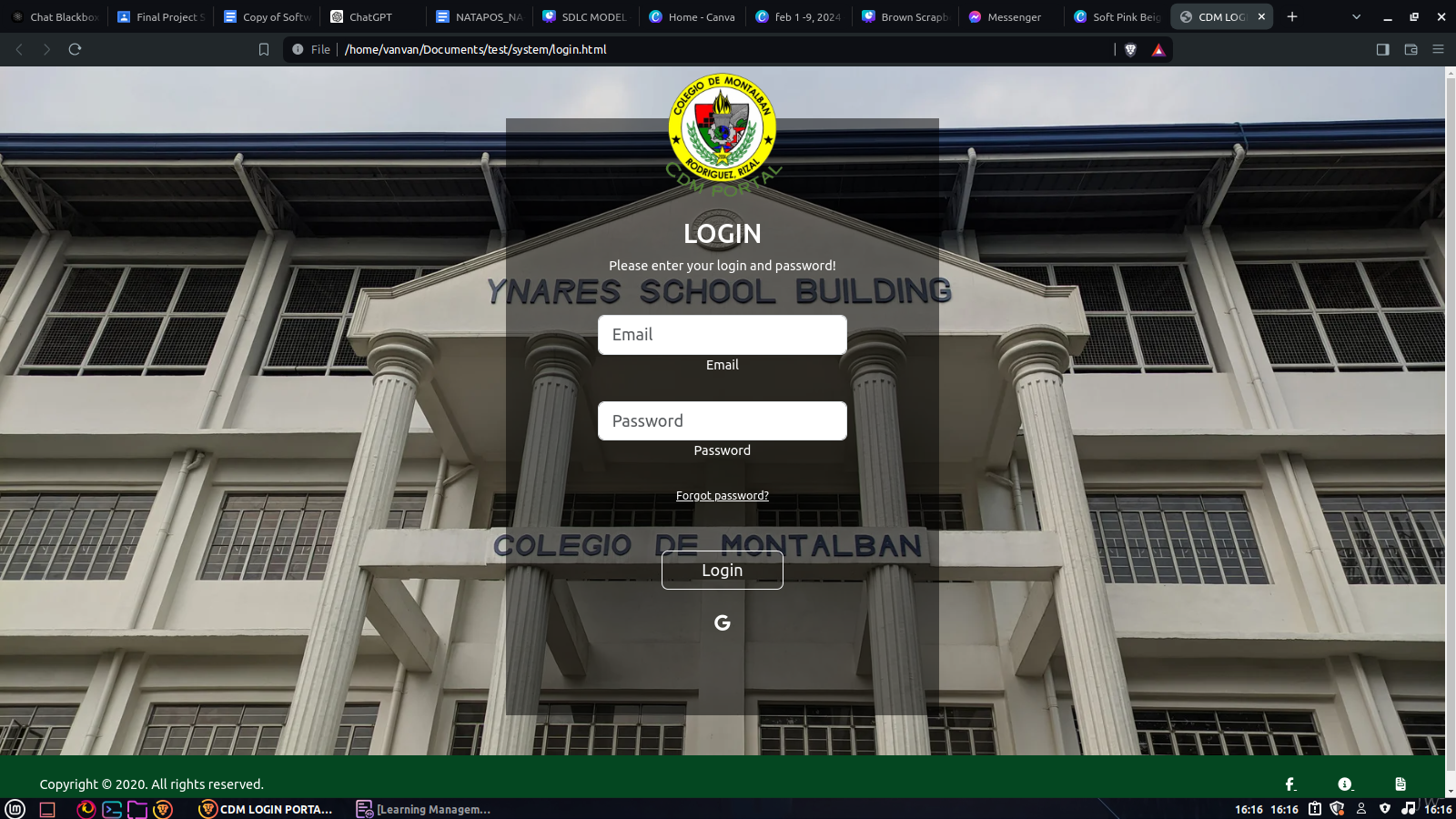
[Flowchart Placeholder - System Workflow]

(Figure 3)

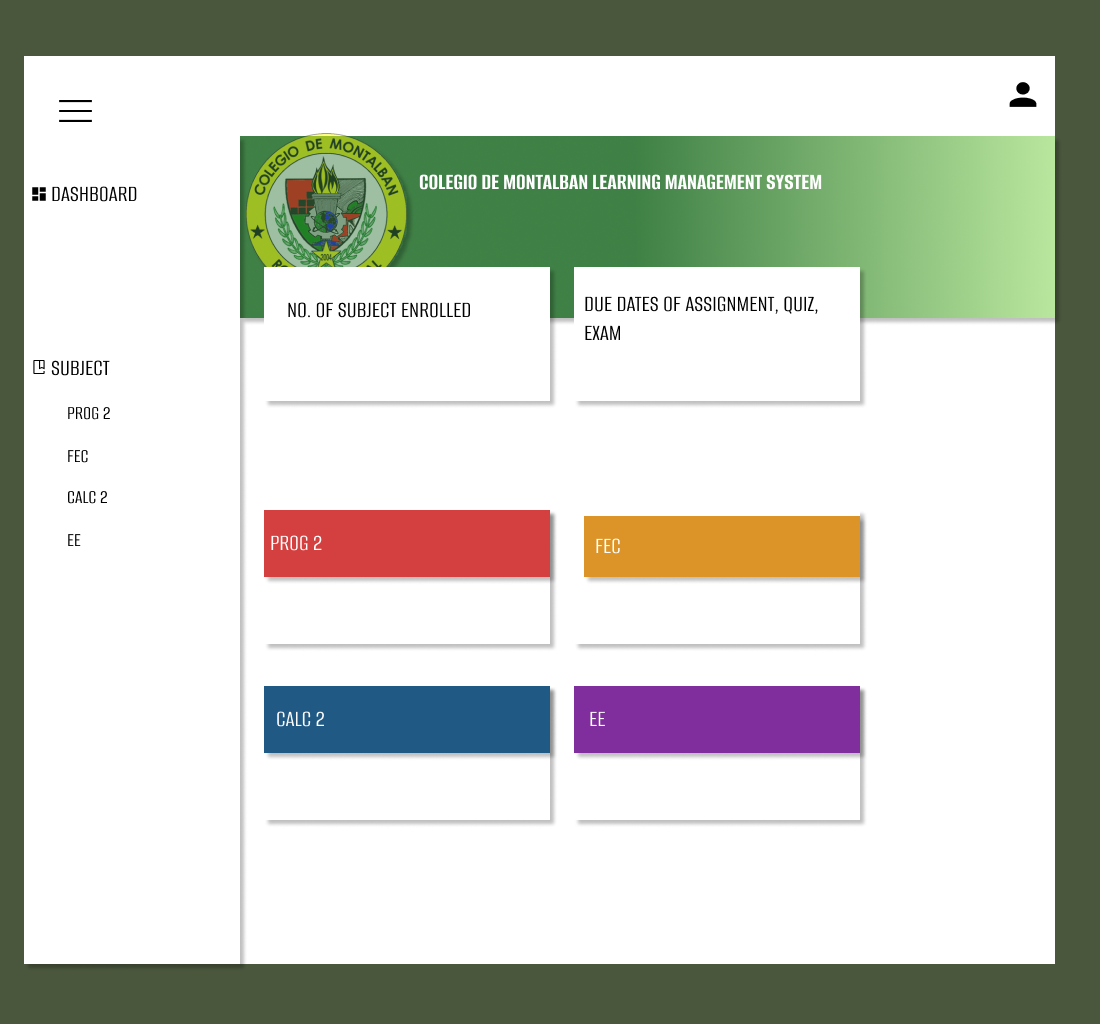
**Figure 3**: illustrates the flowchart for the Learning Management System (LMS) workflow at Colegio de Montalban Learning. The flowchart provides a visual representation of the key processes and interactions within the LMS, outlining the sequence of operations from the user's perspective and the system's internal processing.

Key Components and Flow Steps:

* User Login:
  + Start: The user initiates the process by logging into the LMS.
  + Authentication Check: The system verifies user credentials.
  + Successful Login: If credentials are valid, the user is granted access to the system.



* Dashboard Screen:
  + Navigation Menu: User selects an option from the navigation menu (Create, Delete, Submit subjects).
  + Display Options: The system displays the chosen section's interface.



* Subject card
  + Display the total of enrolled/taught courses of students and instructors.
  + Display the contents of each subject.
* Create Assignment:
  + Select Title and Description of assignment.
  + Attach file (any type of file)
* View Assignment:
  + Able to download the file
  + Able to delete the assignment if user is instructor
* User Account Management:
  + View Account: User accesses their account details.
  + View Details of User: Fullname, age, address, and phone number.

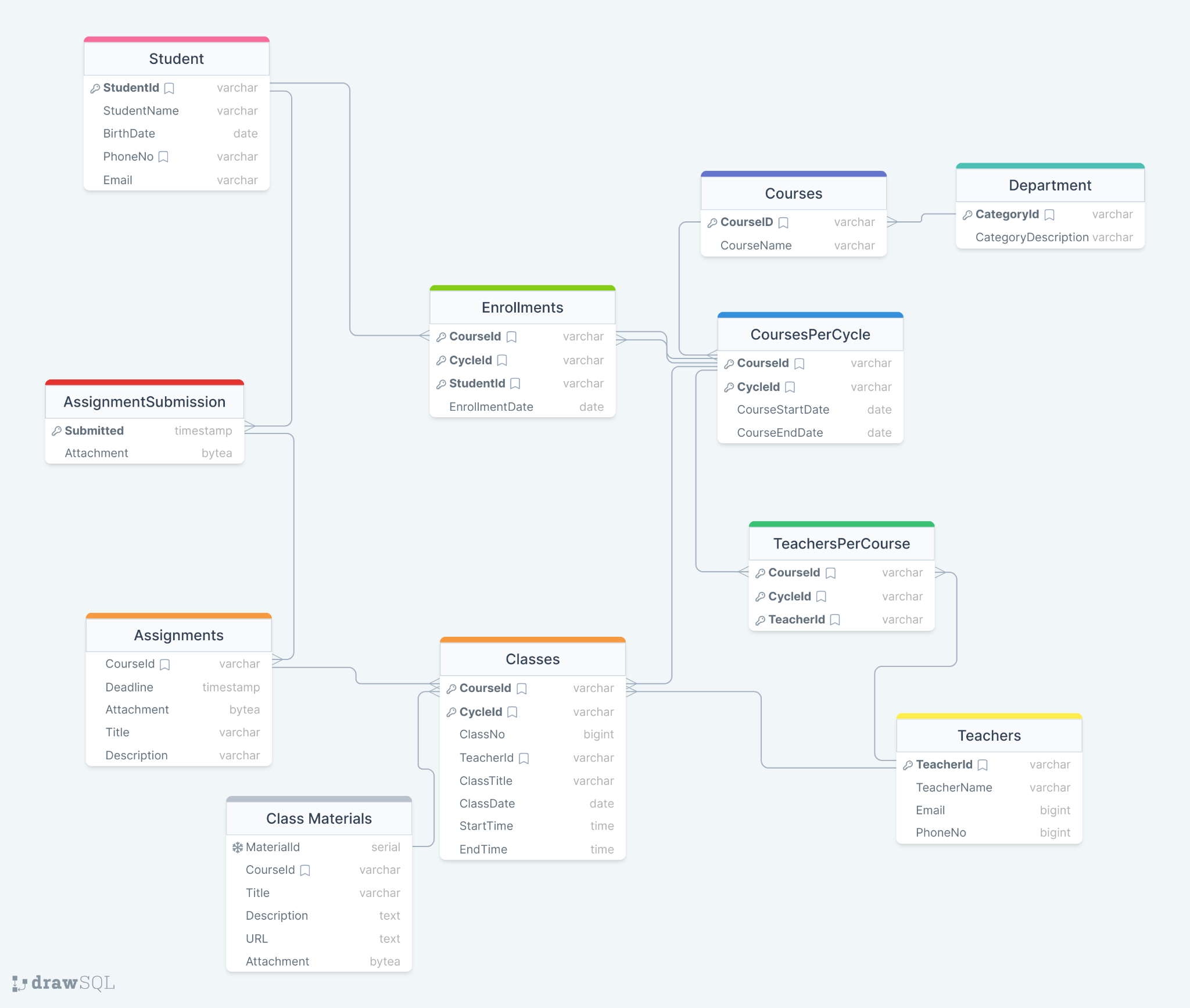
User Actions and System Responses:

* User Actions: Logging in, view assignments, creating and submitting assignments.
* System Responses: Authenticating users, displaying contents of subjects and assignments.

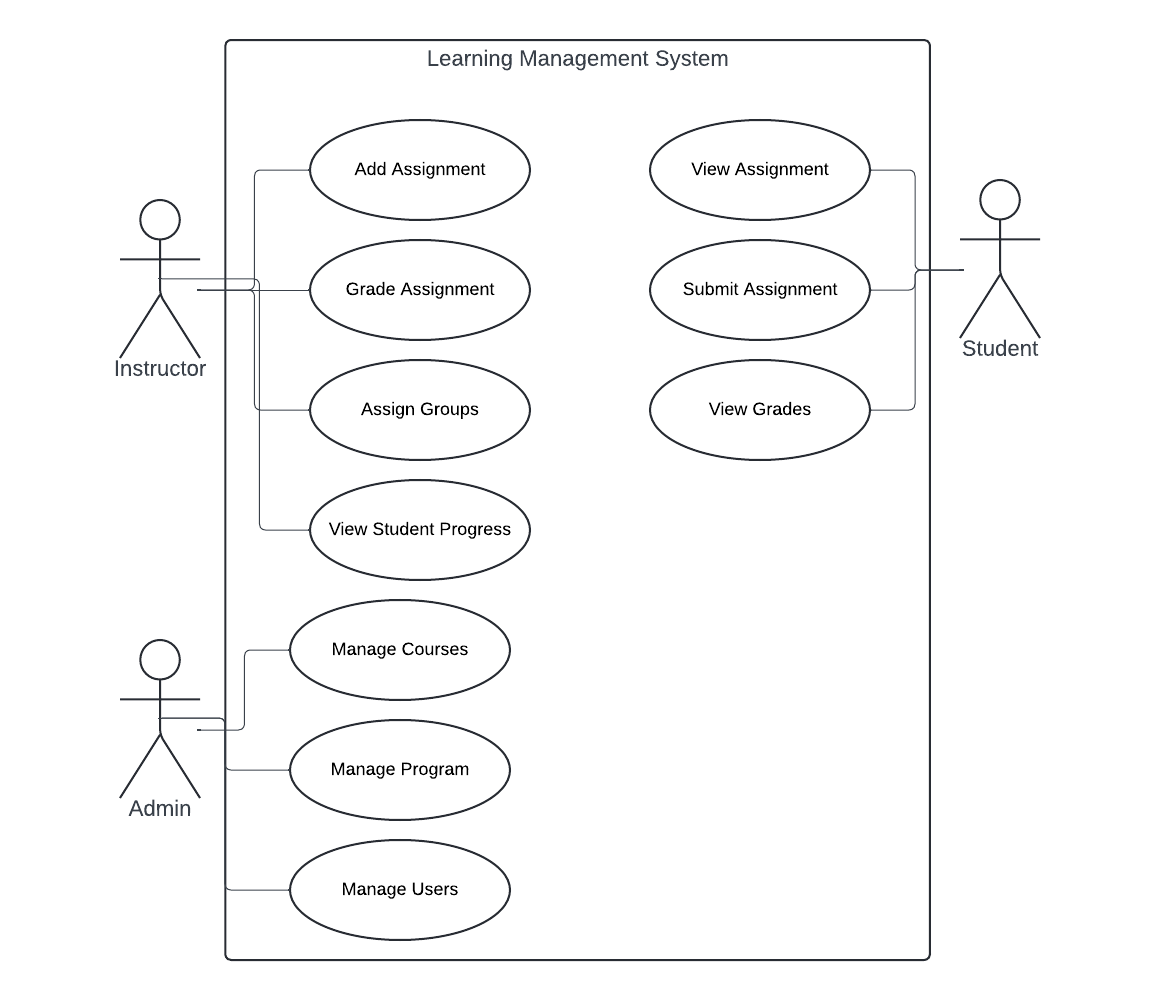
[DFD Placeholder - Data Movement]

(Figure 4)

**Figure 4**: Data Flow Diagrams (DFDs) illustrate the flow of data within a system, delineating how information moves between various components. These diagrams provide a visual representation of processes, data stores, and external entities, aiding in the understanding of system functionalities and data transformations.



**Use Case Diagram:**

****

| **Level** | **Process/Functionality** | **Inputs** | **Process** | **Output** |
| --- | --- | --- | --- | --- |
| 0 | Context | User requests (login, course enrollment, etc.) | User authentication, Course management | Access to courses, Graded assignments, Progress reports |
| 1 | User Authentication | User credentials | Validate user credentials, Grant access | Access granted or denied |
| 1 | Course Management | Course materials, User requests (enroll, drop) | Enroll/drop users from courses, Upload course materials | Updated course enrollment, Uploaded materials |
| 2 | Upload Course Materials | Course materials, Course material database | Validate file format, Upload materials to database | Updated course material database |
| 1 | Course Interaction | User interactions (view lecture, submit assignment), Course materials | Display course content, Evaluate submissions | Viewed lectures, Graded assignments |
| 2 | View Lecture | Lecture selection, Course material database | Retrieve selected lecture, Display to user | Displayed lecture |
| 2 | Submit Assignment | User-submitted assignment, Assignment database | Validate assignment format, Save submission | Saved assignment submission |
| 2 | Evaluate Submissions | User-submitted assignments, Grading rubric | Evaluate submissions based on rubric, Assign grades | Graded assignments |