

Software Requirements Specification (SRS) for College Management System

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the **College Management System (CMS)**. It will outline the system's functionality, performance, and design constraints, serving as a blueprint for development and validation.

1.2 Scope

The system is a multi-user, role-based web application for managing the administrative and academic operations of a college. Its core functions include managing student information, course catalogs, faculty records, enrollment, and grade reporting. The system is intended for use by administrators, faculty, and students to streamline daily institutional tasks and academic processes.

1.3 Definitions and Acronyms

- **System (CMS):** The College Management System.
- **User:** Any individual interacting with the system, including **Administrator**, **Faculty**, and **Student**.
- **Course:** A unit of study offered by the college, which students can enroll in.
- **Enrollment:** The process of a student formally registering for a specific Course in a given semester.
- **Grade:** The record of a student's performance in a Course.
- **UI:** User Interface.
- **API:** Application Programming Interface.

2. Overall Description

2.1 Product Perspective

The system will be a standalone, cloud-based web application designed for access via a standard web browser. It will be built using a modern web framework (e.g., React, Angular, or Vue.js) for the frontend and a robust backend technology (e.g., Node.js, Django, or Spring Boot) with a dedicated database.

2.2 Product Functions

The system provides the following major functionalities:

- **Student Management:** Allows administrators to view, add, edit, and delete student records, including personal and enrollment details.
- **Faculty Management:** Enables administrators to manage faculty records, departmental assignments, and teaching loads.
- **Course Management:** Facilitates the creation, modification, and scheduling of courses and their associated details.
- **Enrollment & Registration:** Provides a mechanism for students to register for courses and for administrators to manage class rosters.
- **Grade Management:** Allows faculty to submit and view grades, and students to access their academic transcripts.
- **Dashboard:** Provides role-specific summaries of key metrics, such as total student enrollment, active courses, and faculty-to-student ratio.

2.3 User Characteristics

The system has three primary user roles:

- **Administrator:** College staff responsible for system configuration, data maintenance, and reporting. They require high-level permissions.
- **Faculty:** Educators who need to manage their courses, record attendance, and submit grades. They require computer literacy.
- **Student:** Registered college members who need to view course information, register for classes, and check their grades. They require basic web interaction skills.

2.4 Design and Implementation Constraints

- **Technology:** Must be implemented using a three-tier architecture (presentation, application, and data).
- **Security:** Must use industry-standard authentication (e.g., OAuth 2.0 or JWT) and authorization controls based on user roles.
- **Responsiveness:** The UI must be fully responsive and optimized for various screen sizes, including desktop and mobile, to ensure access for all users.

- **Data Storage:** Data must be stored in a relational database (e.g., PostgreSQL or MySQL) to maintain data integrity across related tables (students, courses, and enrollment).
-

3. Specific Requirements

3.1 Student Management Module

- **3.1.1 View Student Records:** The system shall display a searchable list of all students, including ID, name, program, and current status.
- **3.1.2 Search Students:** The user (Admin/Faculty) shall be able to search for students by ID, name, or program.
- **3.1.3 Add New Student:** The system shall provide a form for an Administrator to add a new student record.
- **3.1.4 View Transcript:** A Student shall be able to view their complete academic transcript, including all courses and grades.

3.2 Course Management Module

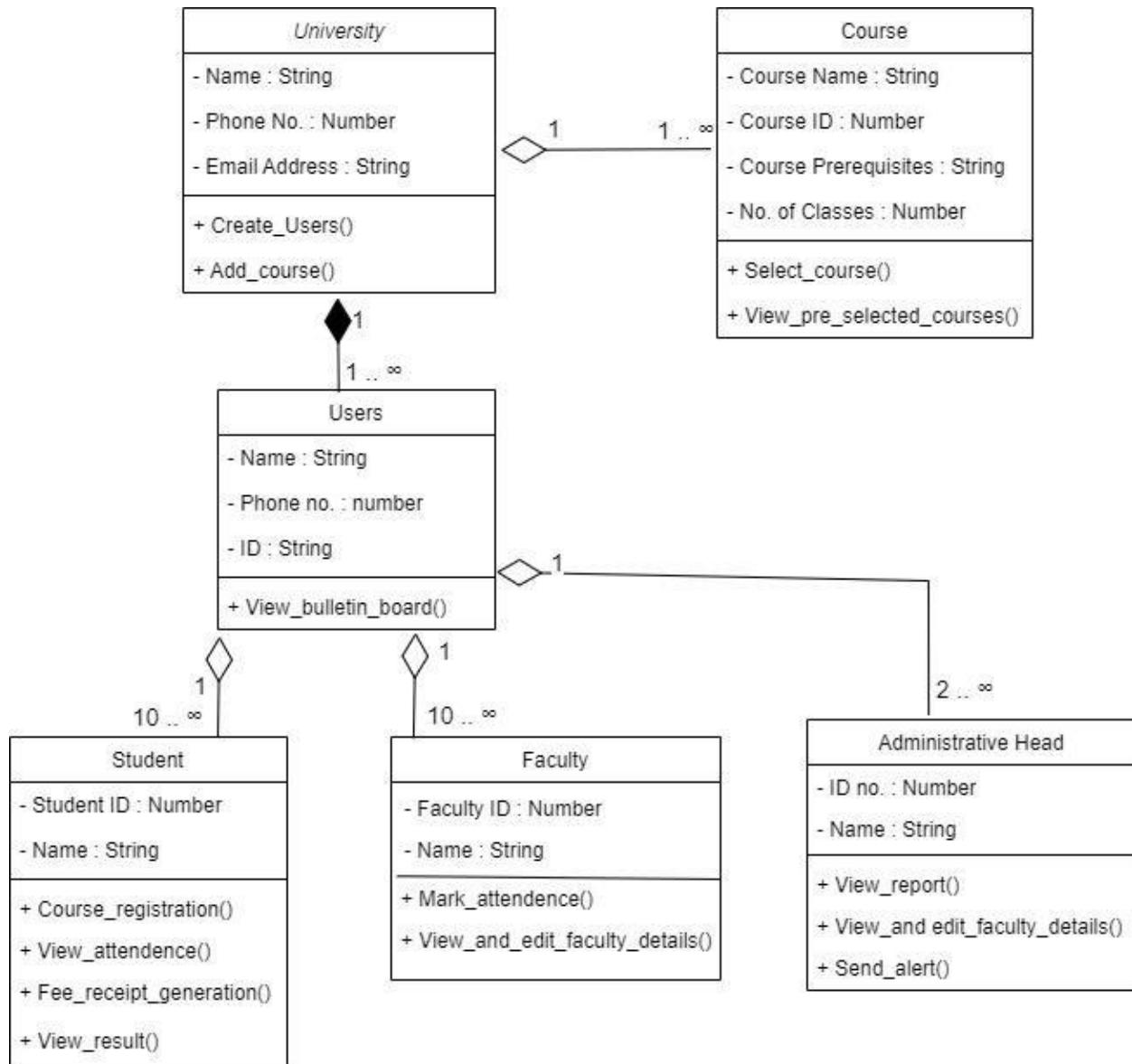
- **3.2.1 View Course Catalog:** The system shall display a list of all active courses, including title, course code, credit hours, and instructor.
- **3.2.2 Add/Edit Course:** An Administrator shall be able to add new courses and modify the details of existing ones.
- **3.2.3 Course Scheduling:** The system shall allow an Administrator to assign a specific Faculty member, time slot, and location to a Course for a given semester.

3.3 Grade Management Module

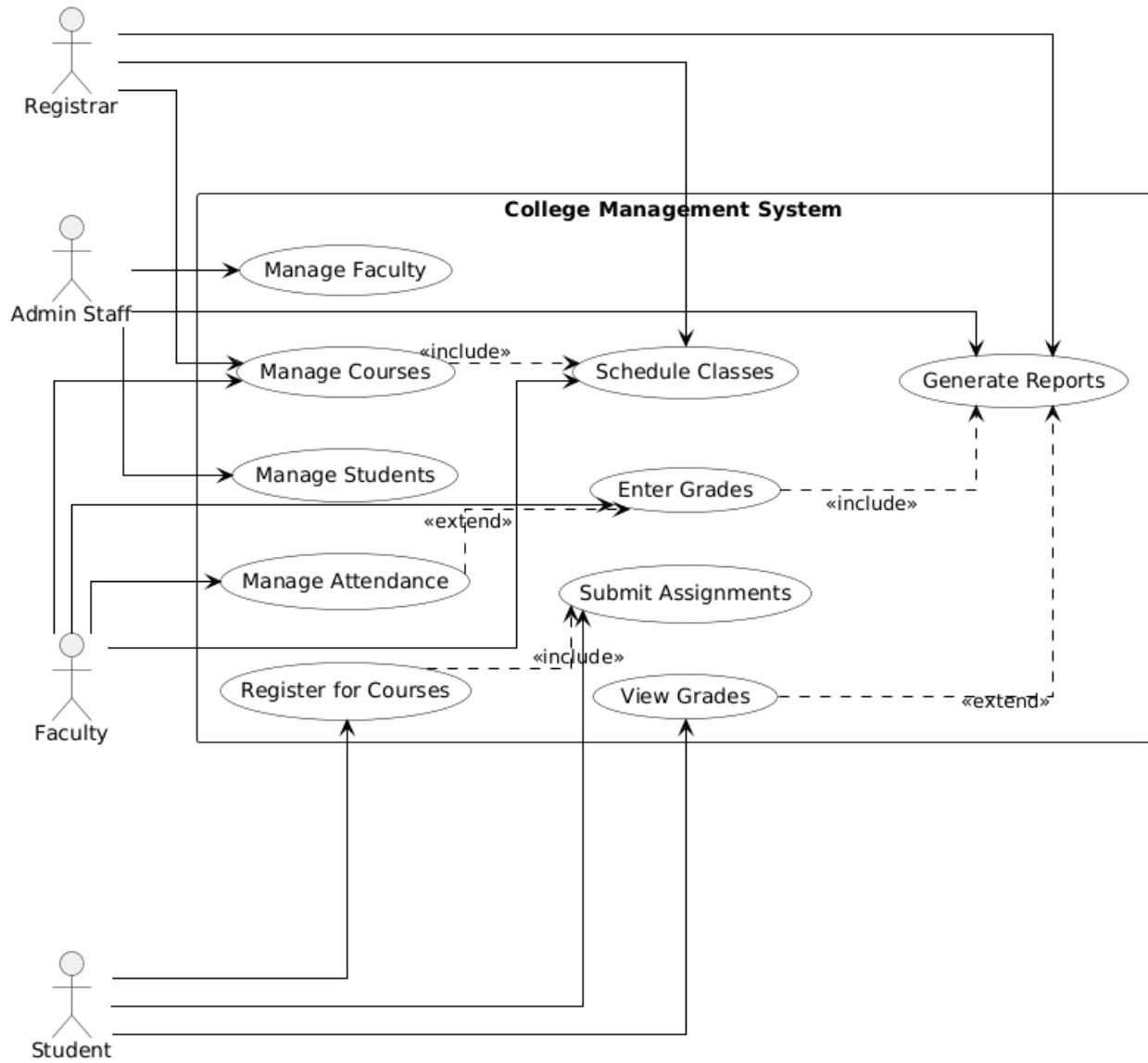
- **3.3.1 Submit Grades:** A Faculty member shall be able to submit final grades for all students in their assigned courses.
 - **3.3.2 Grade Modification:** The system shall require an audit trail for any modifications made to a submitted grade.
 - **3.3.3 Grade Viewing:** A Student shall be able to view only their own grades for enrolled courses.
-

4. System Diagrams

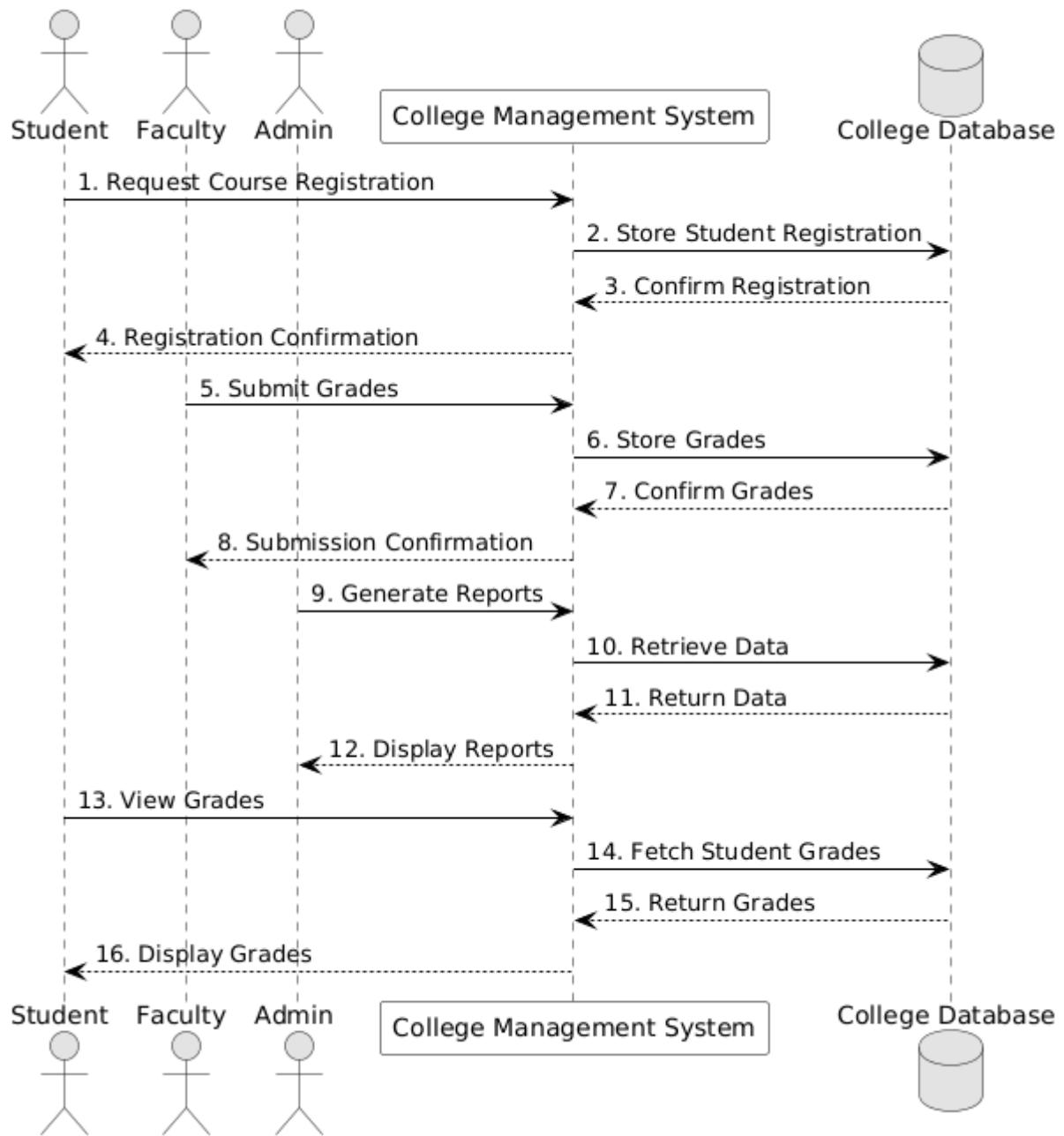
4.1 Class Diagram



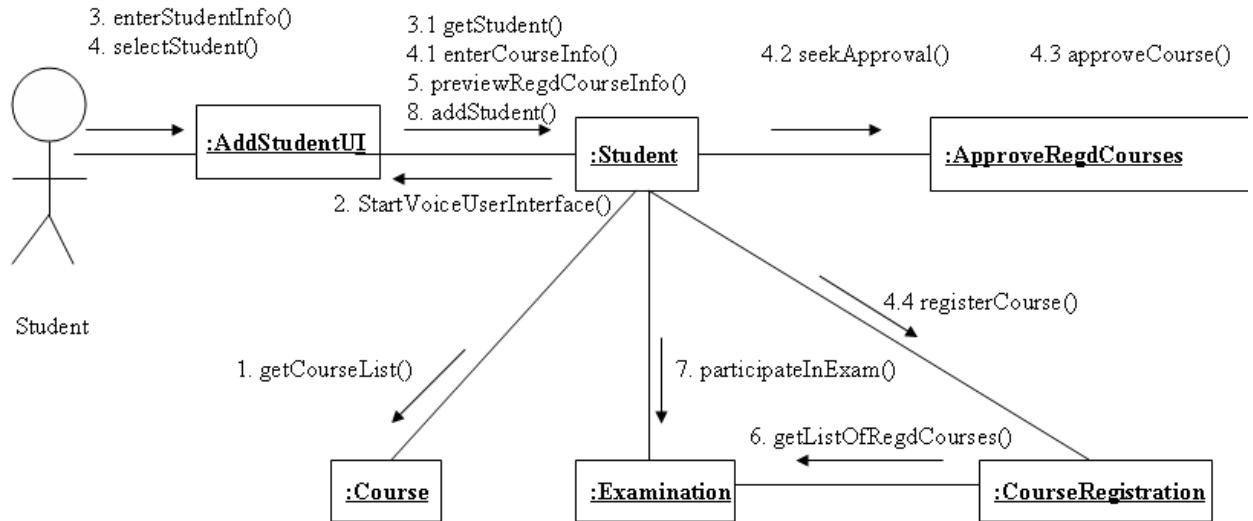
4.2 Usecase Diagram



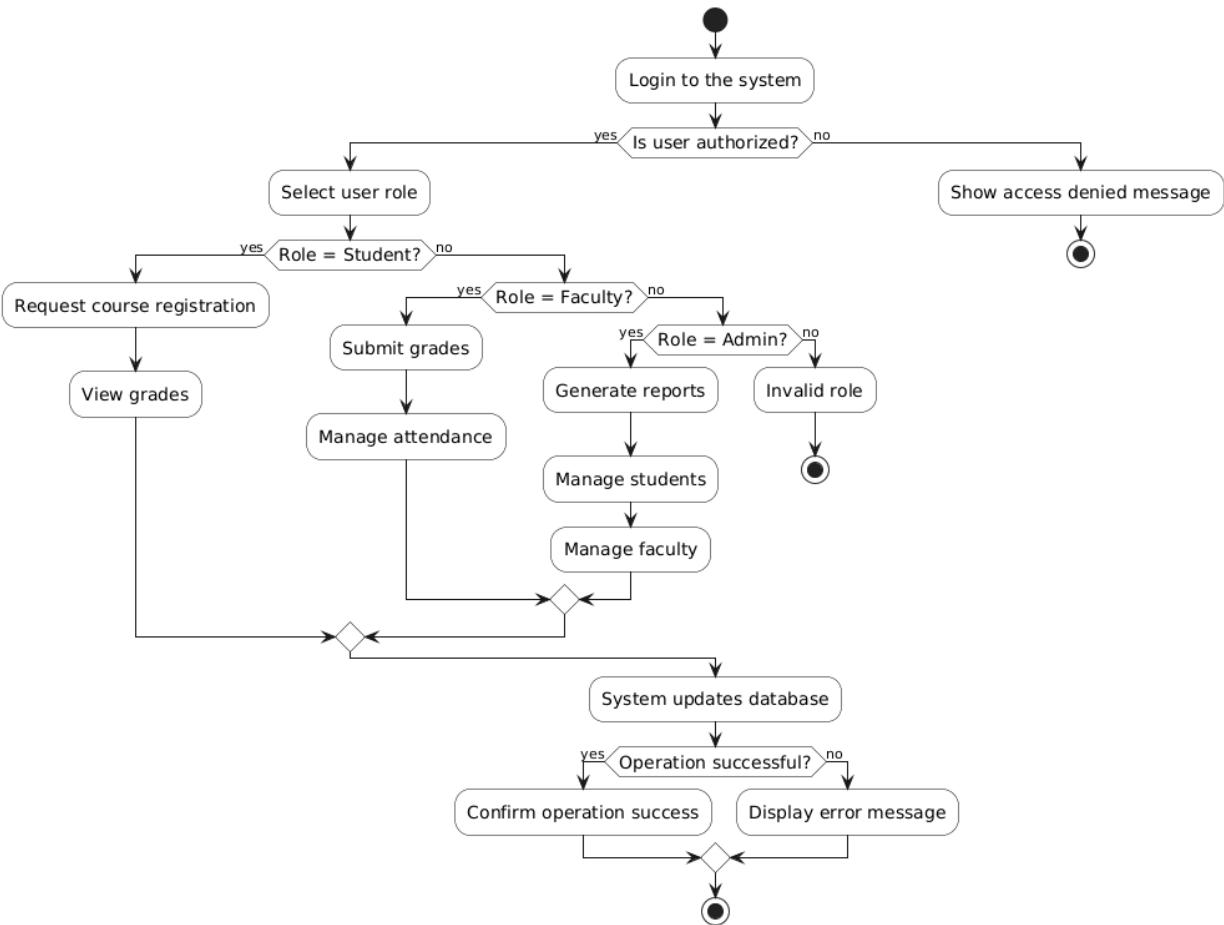
4.3.1 Sequence Diagram



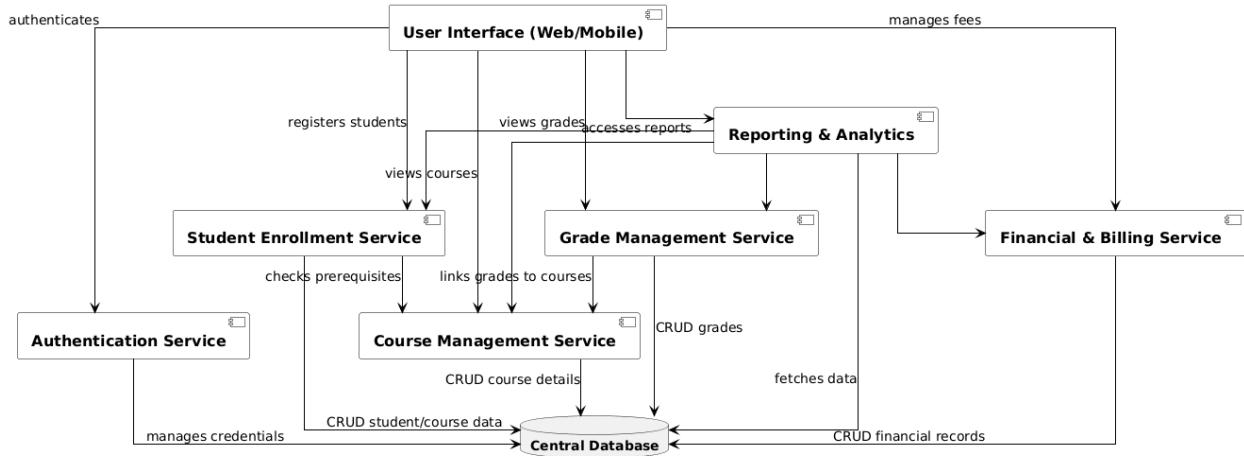
4.3.2 Collaboration Diagram



4.4 Activity Diagram



4.5 Component / Deployment Diagram



Conclusion:

This SRS provides a detailed and structured overview of the requirements for the College Management System. The document and its accompanying diagrams will serve as a foundational guide for the development team. By clearly defining the scope, functionalities, and technical constraints, we ensure that the final product will effectively meet the needs of the college's administrators, faculty, and students.