**SOFTWARE PROJECT PLAN**

**Student Course Enrollment System**

**Project Manager : Uday Shankar Boddupalli (2868122)**

**Design Lead : Yogeshwar Suresh**

**Technical Lead : Yogeshwar Suresh**

1. **Introduction**

The **Student Course Enrollment System** is a web-based application designed to automate and streamline the process of course enrollment for both students and administrators in educational institutions. The system provides a centralized platform for managing course offerings, student enrollments, and class progress. This Software Design Document (SDD) outlines the architectural and design decisions for the Student Course Enrollment System, ensuring the software’s functional and non-functional requirements, as stated in the Software Requirements Specification (SRS), are met.

The SDD includes an overview of the system architecture, detailed design descriptions for each module, and the interactions between different components. The document also covers the data design, application logic, and interface design, providing a comprehensive guide for developers and stakeholders.

**1.1 Goals and Objectives**

The primary goals and objectives of the Student Course Enrollment System are:

**1.1 Goals and Objectives**

1. **Simplify Course Enrollment:** Provide an intuitive platform for students to view and enroll in available courses within predefined limits.
2. **Streamline Course Management:** Enable administrators to create, update, and manage course details and monitor student enrollments.
3. **Ensure Compliance and Data Integrity:** Maintain compliance with institutional policies and ensure secure handling of sensitive data.
4. **Support Scalability and Performance:** Design a system capable of handling concurrent users during peak registration periods.

**1.2 Statement of Scope**

The **Student Course Enrollment System** is a web-based platform that allows educational institutions to manage student enrollment in courses efficiently. The system provides two main user roles: **Admins** and **Students**. Admins can create and manage courses, set enrollment limits, and track student progress. Students can browse available courses, enroll in a maximum of three active classes, and monitor their progress.

The scope includes functionalities for:

1. **Students**: Logging in, viewing available classes, enrolling in classes, and tracking their progress.
2. **Admins**: Creating, updating, and deleting classes, managing student enrollments, and marking student progress.

**1.3 Software Context**

The **Student Course Enrollment System** is a web-based application designed for internal use within an educational institution. It will be hosted on a secure server and accessed by authorized users only—students and admins. The system will be deployed on Azure for scalability and availability, and will provide role-based access control to ensure data security and integrity. All interactions will be secured using HTTPS to protect sensitive information during data transmission.

**1.4 Major Constraints**

The Student Course Enrollment System is subject to the following constraints:

1. **Time Constraint**: The system must be developed and deployed within 12 weeks.
2. **Resource Constraint**: The project is limited to a team of three developers.
3. **Technical Constraint**: The system must support up to 30 concurrent users without performance issues.
4. **Compliance Constraint**: The system must comply with data protection regulations and university policies.
5. **Scope Constraint**: Initial release will include basic enrollment and management features, excluding advanced integrations and automation.

**2.0 Data Design**

The data design for the Student Course Enrollment System outlines the structure and relationships of core entities within the system.

**2.1 Data Description:**

The Student Course Enrollment System manages data for five primary entities:

1. **Admin**: Stores admin credentials and contact details.
2. **Student**: Contains student profiles and enrollment dates.
3. **Course**: Holds course details such as name, description, and seat availability.
4. **Enrollment**: Tracks student enrollments and completion statuses for courses.
5. **Progress**: Records the progress of each student in enrolled courses.

Each entity has a unique primary key and relationships to ensure data consistency and integrity.

**2.2 Entity Schema Design**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1. Admin Table** | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **Attribute** | | | **Type** | | | **Description** | |
| AdminID | | | INT (PK) | | | Unique identifier for each admin | |
| Username | | | VARCHAR | | | Admin’s username | |
| Password | | | VARCHAR | | | Encrypted password | |
| Email | | | VARCHAR | | | Admin’s email | |
|  | | |  | | |  | | | | |
| **Relationships**: | | |  | | |  | | | | |
| **Admin** has a **one-to-many** relationship with the **Course** table (each admin manages multiple courses). | | | | | | | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **2. Student Table** | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **Attribute** | | | **Type** | | | **Description** | |
| StudentID | | | INT (PK) | | | Unique identifier for each student | |
| FirstName | | | VARCHAR | | | Student’s first name | |
| LastName | | | VARCHAR | | | Student’s last name | |
| Email | | | VARCHAR | | | Student’s email | |
| Password | | | VARCHAR | | | Encrypted password | |
| EnrollmentDate | | | DATE | | | Date student was enrolled | |
|  | | |  | | |  | | | | |
| **Relationships**: | | |  | | |  | | | | |
| 1. **Student** has a **one-to-many** relationship with the **Enrollment** table (a student can have multiple enrollments). | | | | | | | | | | |
| 1. **Student** has a **one-to-many** relationship with the **Progress** table (a student can have progress records for multiple courses). | | | | | | | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **3. Course Table** | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **Attribute** | | | **Type** | | | **Description** |
| CourseID | | | INT (PK) | | | Unique identifier for each course |
| CourseName | | | VARCHAR | | | Name of the course |
| Description | | | TEXT | | | Detailed description of the course |
| MaxSeats | | | INT | | | Maximum number of students allowed |
| CurrentSeats | | | INT | | | Number of students currently enrolled |
| StartDate | | | DATE | | | Course start date |
| EndDate | | | DATE | | | Course end date |
| AdminID | | | INT (FK) | | | Foreign key from Admin table |
|  | | |  | | |  | | | | |
| **Relationships**: | | |  | | |  | | | | |
| 1. **Course** has a **many-to-one** relationship with the **Admin** table (each course is managed by one admin). | | | | | | | | | | |
| 1. **Course** has a **one-to-many** relationship with the **Enrollment** table (many students can enroll in one course). | | | | | | | | | | |
| 1. **Course** has a **one-to-many** relationship with the **Progress** table (multiple students can have progress in one course). | | | | | | | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **4. Enrollment Table** | | | | | |  | | | | |
|  | | |  | | |  | | | | |
| **Attribute** | | | **Type** | | | **Description** |
| EnrollmentID | | | INT (PK) | | | Unique identifier for each enrollment |
| StudentID | | | INT (FK) | | | Foreign key from Student table |
| CourseID | | | INT (FK) | | | Foreign key from Course table |
| EnrollmentDate | | | DATE | | | Date the student enrolled in the course |
| CompletionStatus | | | VARCHAR | | | Status of the enrollment (e.g., Active, Completed) |
|  | | |  | | |  | | | | |
| **Relationships**: | | |  | | |  | | | | |
| 1. **Enrollment** has a **many-to-one** relationship with the **Student** table (many enrollments for one student). | | | | | | | | | | |
| 1. **Enrollment** has a **many-to-one** relationship with the **Course** table (many enrollments for one course). | | | | | | | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **5. Progress Table** | | |  | | |  | | | | |
|  | | |  | | |  | | | | |
| **Attribute** | | | **Type** | | | **Description** | |
| ProgressID | | | INT (PK) | | | Unique identifier for each progress record | |
| StudentID | | | INT (FK) | | | Foreign key from Student table | |
| CourseID | | | INT (FK) | | | Foreign key from Course table | |
| Progress | | | DECIMAL | | | Percentage of course progress | |
|  | | |  | | |  | | | | |
| **Relationships**: | | |  | | |  | | | | |
| 1. **Progress** has a **many-to-one** relationship with the **Student** table(one student can have multiple progress | | | | | | | | | | |
| 1. **Progress** has a **many-to-one** relationship with the **Course** table (multiple students can have progress records in one course). | | | | | | | | | | |

**2.2 ER Diagram**

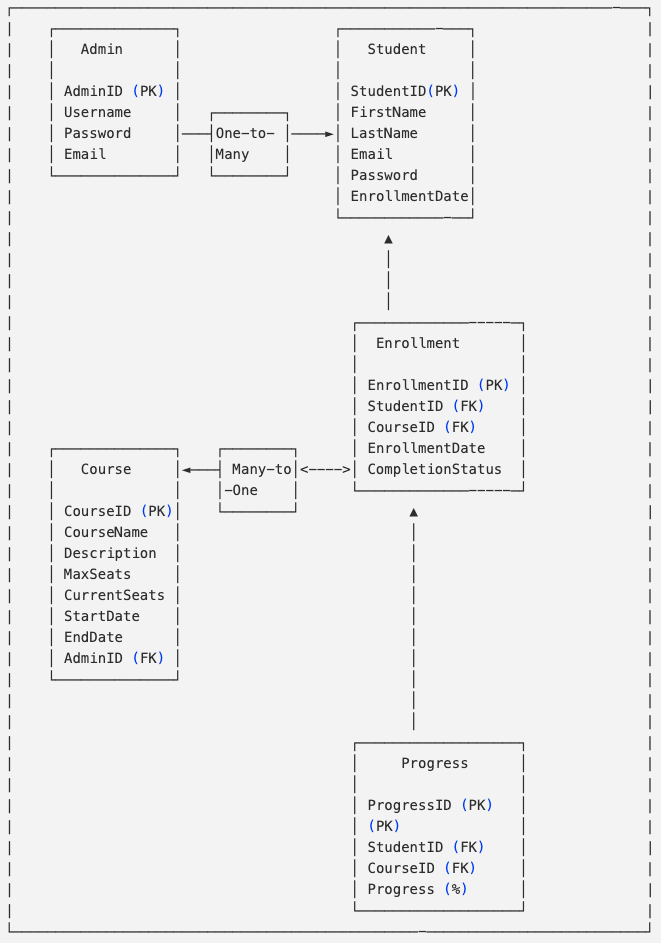


Fig: ER Diagram

**3.0 Architectural and Component-Level Design**

This section outlines the architecture and component-level design of the Student Course Enrollment System, providing a clear view of how different components interact within the system. The architecture ensures that the system is modular, scalable, and easy to maintain.

**3.1 Architecture Diagrams**

**3.1.2 Project Architecture Diagram**

A diagram of a web server

Description automatically generated

Fig: Project Architecture

**3.1. System Design**

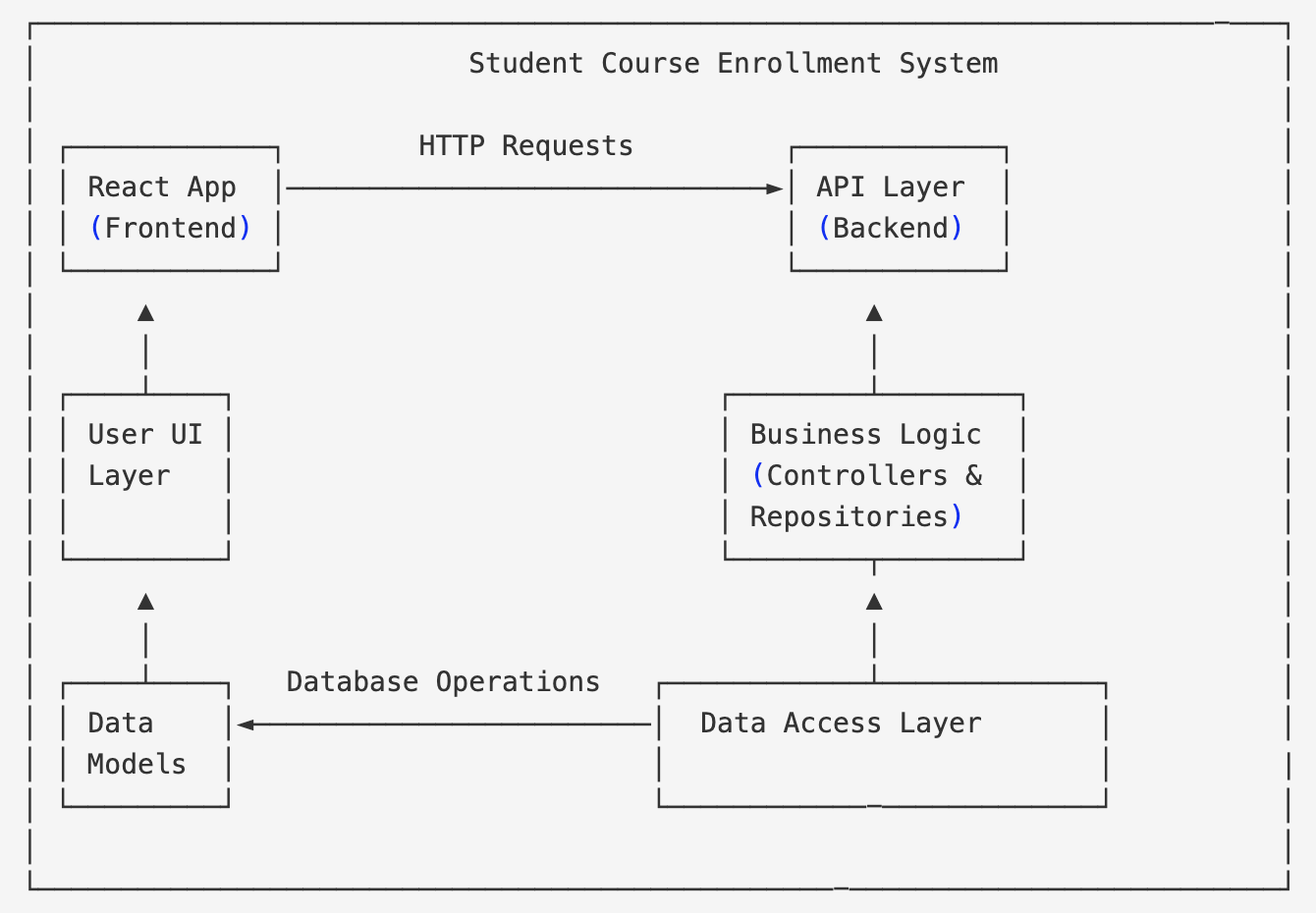


Fig : System Design

**3.1.3 Sequence Diagram**

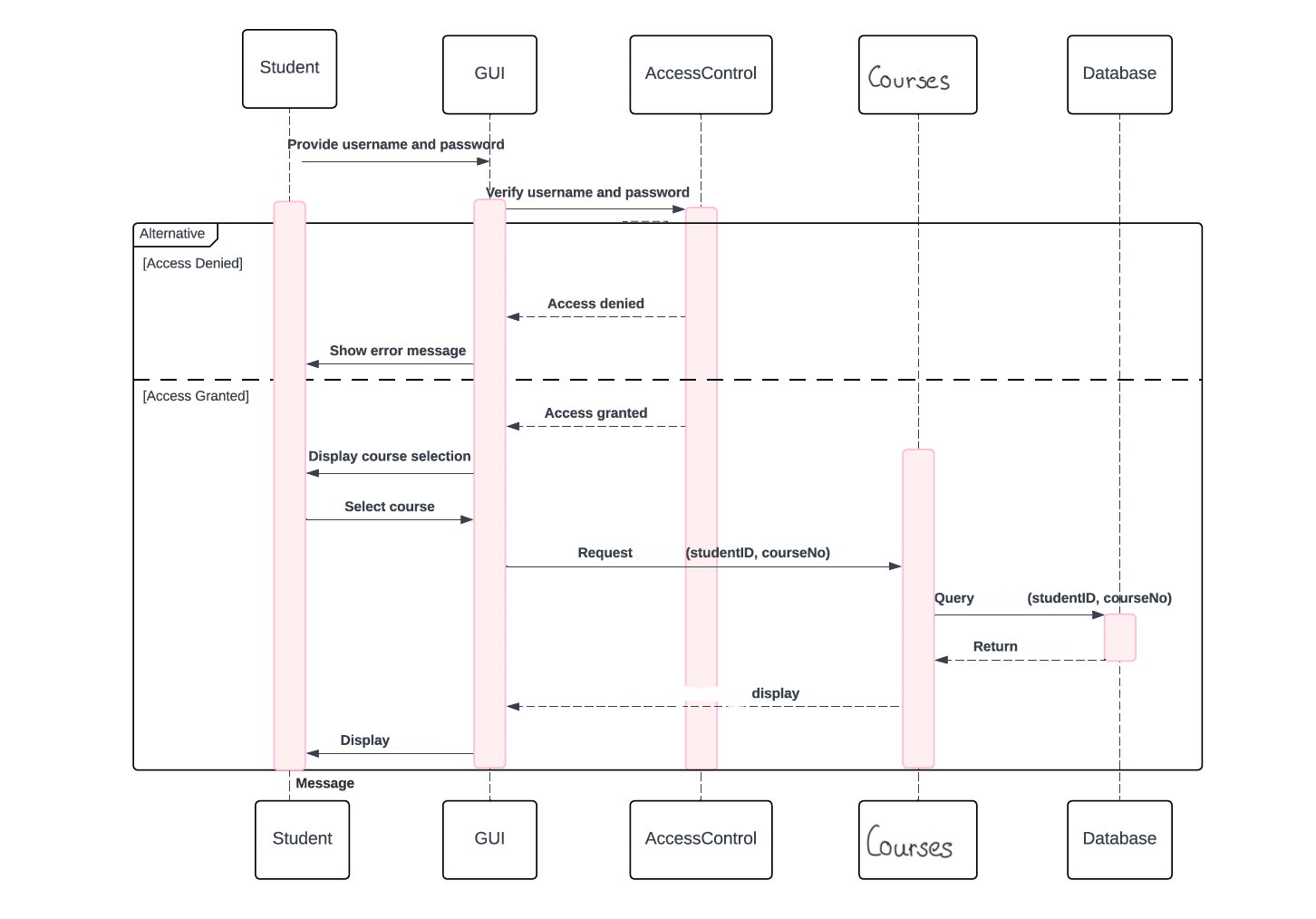


Fig: Sequence Diagram

**3.2 Description for Components**

1. **Presentation Layer**:
   * This layer consists of the **User Interface** (UI), where students and admins interact with the system.
   * It includes forms and views for user login, course enrollment, and course management.
2. **Business Logic Layer**:
   * Contains the core business logic that governs how data is processed and managed.
   * Ensures that business rules are enforced, such as a student being limited to three active enrollments.
   * Handles course creation, updating, and student progress tracking.
3. **Data Access Layer**:
   * Manages database interactions and ensures that data is stored and retrieved in a consistent manner.
   * Uses a relational database schema to manage entities such as students, courses, enrollments, and progress.

**4.0 User Interface Design**

The User Interface (UI) design for the Student Course Enrollment System is focused on simplicity and usability, ensuring that both students and admins can easily navigate and manage their activities. The design follows a dashboard layout, providing a centralized view of key functionalities.

**4.1 Student Dashboard**

The Student Dashboard provides an overview of the courses a student is enrolled in and allows for quick access to enrollment management features. The main elements include:

* **Enrolled Courses**: A table or card view displaying all the active courses the student is enrolled in, along with status information (e.g., Active, Completed).
* **Available Courses**: A section showing courses that the student can enroll in. Each course displays basic information such as course name, schedule, and seat availability.
* **Actions**:
  + **Enroll in Course**: A button next to each available course, allowing the student to enroll if eligibility criteria are met.
  + **Drop Course**: A button allowing the student to drop an active course if necessary.

**4.2 Admin Dashboard**

The Admin Dashboard provides a comprehensive view of course management and student enrollments. The main elements include:

* **Course Management**: A list of all existing courses with options to create, update, or delete courses.
* **Enrollment Overview**: A table or card view showing the number of students enrolled in each course and available seats.
* **Student Progress**: A detailed view of each student’s progress within a course, with the ability to mark courses as complete.
* **Actions**:
  + **Create Course**: A button that allows the admin to add a new course by providing course details such as name, description, and schedule.
  + **Update Course**: Options to modify course information or adjust seat availability.
  + **Manage Enrollments**: A feature to view and manage student enrollments, including the ability to add or remove students from a course.

**4.2 Interface Design Rules**

* + 1. **Consistency:** Both dashboards follow a consistent layout and design pattern, making it easy for users to switch between roles if necessary.
    2. **Usability:** Elements are designed to be self-explanatory and user-friendly, with clear labels and intuitive controls.
    3. **Minimalist Design:** The interface avoids clutter and focuses on displaying the most relevant information for each role.