**Software Design Document**  
Project: **StudentCourse Registration System**

Submitted by:

Khyyati Vegiraju

**1. Introduction**

**1.1 Purpose**

This document provides a detailed design of the StudentCourse Registration System, a web-based platform that facilitates student course registration, login/signup authentication, and audit tracking.

**1.2 Scope**

StudentCourse enables students to:

* Sign up and log in securely
* Register with personal details
* Select available courses
* Confirm course selection with a timestamped audit

Administrators or system maintainers can manage database records and monitor course audits.

**1.3 Definitions**

* **Frontend**: User interface built using Angular
* **Backend**: RESTful APIs built using Java (Quarkus)
* **Database**: MySQL for persistent data storage
* **Audit**: Log of course registration confirmation

**1.4 Overview**

This document includes architecture, system and user requirements, UI and data design, API details, sequence diagrams, testing, and future enhancements.

**2. System Overview**

The StudentCourse system provides an intuitive interface for students to sign up, register, and track course selection. It consists of:

* Frontend in Angular with standalone components
* REST APIs in Quarkus
* MySQL database with foreign key constraints

**3. System Architecture**

**3.1 Overall Architecture**

* **Frontend**: Angular 16 (modular, form-driven components)
* **Backend**: Quarkus REST API with JPA
* **Database**: MySQL with relational structure
* Communication via HTTP (JSON format)

| **Layer** | **Technology** |
| --- | --- |
| Frontend | Angular 16 |
| Backend | Quarkus (Java) |
| Database | MySQL |
| Styling | CSS (Responsive) |
| Tools | Postman, VS Code |

**4. Functional Requirements**

**4.1 User Requirements**

* Students can sign up and log in
* Registration form with validations
* View and select available courses
* View confirmation and sign out

**4.2 System Requirements**

* Prevent duplicate emails during signup
* Store student and course data with integrity
* Provide consistent session-based navigation

**5. UI Design**

* Responsive and centered layout using CSS Flexbox
* Color palette: light green/blue with white backgrounds
* Components:
  + LoginComponent
  + SignupComponent
  + StudentRegistrationComponent
  + CourseSelectionComponent
  + ConfirmationComponent

**6. Data Design**

**6.1 Tables**

* **users**: id, email, password
* **student**: student\_id, first\_name, last\_name, email, phone, college\_name
* **course**: course\_id, course\_name, description, duration
* **registration**: registration\_id, student\_id (FK), course\_id (FK), registration\_date
* **course\_audit**: audit\_id, registration\_id (FK), student\_id, course\_id, course\_start\_date, course\_end\_date

**6.2 Data Flow**

A diagram of a student table

AI-generated content may be incorrect.

**7. API Design**

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Purpose |
| POST | /auth/signup | Register new user |
| POST | /auth/login | Authenticate user |
| POST | /students | Save student details |
| GET | /courses | Fetch available courses |
| POST | /registrations | Register student to course |
| POST | /course-audits | Confirm course registration |

**8. Non-Functional Requirements**

* **Performance**: Response time under 2s for major actions
* **Security**: Email uniqueness, password handling
* **Maintainability**: Modular code, reusable services
* **Scalability**: DB structure supports multiple students & courses

**9. Testing Strategy**

* **Manual Testing** using Postman for all APIs
* **Form Validation Testing** in Angular
* **Integration Testing** for session-based flow
* **Error Handling**: Alert messages, backend error mapping

**10. Future Enhancements**

* Password hashing and JWT security
* Admin panel for managing students and courses
* Email confirmation on registration
* Progress tracking and course history

**11. Screenshots**

**A screenshot of a computer

AI-generated content may be incorrect.**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer screen

AI-generated content may be incorrect.