
Software Design Specifications

for

**REGISTRATION OF COURSE ELECTIVES OR
PROJECTS**

Prepared by:

Teja - SE22UCSE189-frontend developer

mohith-se22ucse174-frontend developer

srishanth rishi -se22ucse007-backend developer

aryan-se23lcse002-backend developer

chandu-se22ucse070-backend developer

dinesh-se22ucse267-designer.

Document Information

Title : Registration of electives or projects Project Manager: Nartkanni mam	Document Version No:01
Prepared By:Team 2	Document Version Date:09/04/25
	Preparation Date:08/04/25

Version History

Ver. No.	Ver. Date	Revised By	Description	Filename

Table of Contents

1 INTRODUCTION	4
1.1 PURPOSE	4
1.2 SCOPE	4
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	4
1.4 REFERENCES	4
2 USE CASE VIEW	4
2.1 USE CASE	4
3 DESIGN OVERVIEW	4
3.1 DESIGN GOALS AND CONSTRAINTS	5
3.2 DESIGN ASSUMPTIONS	5
3.3 SIGNIFICANT DESIGN PACKAGES	5
3.4 DEPENDENT EXTERNAL INTERFACES	5
3.5 IMPLEMENTED APPLICATION EXTERNAL INTERFACES	5
4 LOGICAL VIEW	5
4.1 DESIGN MODEL	6
4.2 USE CASE REALIZATION	6
5 DATA VIEW	6
5.1 DOMAIN MODEL	6
5.2 DATA MODEL (PERSISTENT DATA VIEW).....	6
5.2.1 Data Dictionary.....	6
6 EXCEPTION HANDLING	6
7 CONFIGURABLE PARAMETERS	6
8 QUALITY OF SERVICE	7
8.1 AVAILABILITY.....	7
8.2 SECURITY AND AUTHORIZATION	7
8.3 LOAD AND PERFORMANCE IMPLICATIONS	7
8.4 MONITORING AND CONTROL	7

1 Introduction

1.1 Purpose

The purpose of this document is to define the software design specifications for the “Registration of Elective Courses or Projects” system. It is intended for software developers, testers, project managers, and stakeholders to ensure a shared understanding of the system architecture and design decisions.

1.2 Scope

This system allows students to register for elective courses or final-year projects based on eligibility, availability, and preferences. Admins and instructors manage offerings, approve registrations, and oversee allocation.

1.3 Definitions, Acronyms, and Abbreviations

- | | | |
|----|--------------|--------------------------------------|
| 1. | SDS | Software Design
Specification |
| | User | |
| 2. | UI | Interfa
ce |
| 3. | Databa
DB | se |
| 4. | API | Application Programming
Interface |

1.4 References

1. JUNO
2. Institution Academic Policy Documents
3. SRS Document for Registration System

2 Use Case View

2.1 Use Case

Use Case: Course Registration

- Actors: Student, Admin, Instructor
- Steps:
 - Student logs in

- Views eligible electives/projects
- Selects preferred choices
- Admin reviews and approves
- System finalizes allocation

Other use cases:

- View available electives
- Manage elective offerings (Admin)
- Approve project topics (Instructor)

3 Design Overview

3.1 Design Goals and Constraints

- Support scalability for multiple departments
- Real-time conflict detection
- Secure login and data privacy
- Availability across devices

Constraints:

- Compliance with university regulations and policies.
- Project completion within the academic semester.

3.2 Design Assumptions

- Users have unique institutional credentials
- All electives have pre-defined seat limits
- Admins are responsible for data integrity

3.3 Significant Design Packages

- Authentication Module
- Course Management Module
- Registration Module
- Notification System
- Reporting Module

3.4 Dependent External Interfaces

The "Registration of Elective Courses or Projects" system relies on several external systems and services for core functionality. These interfaces ensure seamless integration with institutional infrastructure and third-party tools.

1. Authentication System

- **Interface:** University Single Sign-On (SSO) / LDAP
- **Purpose:** Validates student and admin credentials during login.
- **Protocol:** HTTPS, REST API or LDAP
- **Dependency:** Must be online and responsive for users to access the system.

2. Student Information System (SIS)

- **Interface:** University SIS API
- **Purpose:** Retrieves student profiles, academic standing, year/semester, and course eligibility.
- **Protocol:** REST API / SOAP

- **Dependency:** Required for accurate elective availability and eligibility checks.

3. Email Notification Service

- **Interface:** SMTP server / Email API (e.g., SendGrid, Mailgun)
- **Purpose:** Sends confirmation emails and registration updates to students and staff.
- **Protocol:** SMTP or HTTPS
- **Dependency:** Used during successful registration, rejections, or allocation confirmations.

4. Course Catalog Database

- **Interface:** Internal or external course catalog system
- **Purpose:** Retrieves elective/project course offerings, details, instructor info, seat limits.
- **Protocol:** REST API or direct database connection
- **Dependency:** Needed to show available electives/projects during registration.

5. Admin Dashboard Framework (Optional)

- **Interface:** Admin Panel Framework (could use Angular/React Admin Template)
- **Purpose:** Allows administrators to manage courses, approve registrations, and view reports.
- **Protocol:** Web interface communicating via RESTful APIs with backend services.

3.5 Implemented Application External Interfaces (and SOA web services)

The table below lists the implementation of public interfaces this design makes available for other applications.

Interface Name	Module Implementing the Interface	Functionality/Description
1.Course API	Course Management	Returns list of electives/projects
2.Registration API	Registration Module	Handles student submissions

4 Logical View

4.1 Design Model

- Classes: Student, Admin, Instructor, Elective, Project, Registration, Notification
- Relationships
- A Student can register for multiple Electives
- An Instructor can supervise multiple Projects

4.2 Use Case Realization

Basic Flow (Main Scenario):

1. Student Logs In

- The student authenticates using institutional credentials.
- The system retrieves user roles and permissions.

2. System Displays Available Electives/Projects

- Based on student's department, semester, and eligibility, the system shows a list of available courses or project topics.
- Shows available slots, instructors, and deadlines.

3. Student Selects Preferences

- The student selects multiple electives/projects in order of preference.
- The system validates inputs and checks for conflicts (e.g., time clashes).

4. Student Submits Choices

- The system temporarily stores the submission and marks it as "Pending Review."

5. Admin Reviews and Approves

- Admin views all student submissions.

- Approves based on capacity, eligibility, or predefined allocation logic (e.g., GPA priority, first-come-first-serve, lottery).
- The system allocates electives/projects accordingly.

6. System Confirms Registration

- Final allocations are stored in the database.
- Students receive notifications (email/SMS) confirming their registered electives or assigned projects.

Alternative Flows:

- **A1: Invalid Input / Preference Conflict**
 - The system alerts the student about conflicts (e.g., two electives at the same time).
 - Student must resolve before submission.
- **A2: Elective/Project Full**
 - If the elective/project limit is reached, student's next preference is considered.
 - If all options are full, the student is placed on a waiting list.

Postconditions:

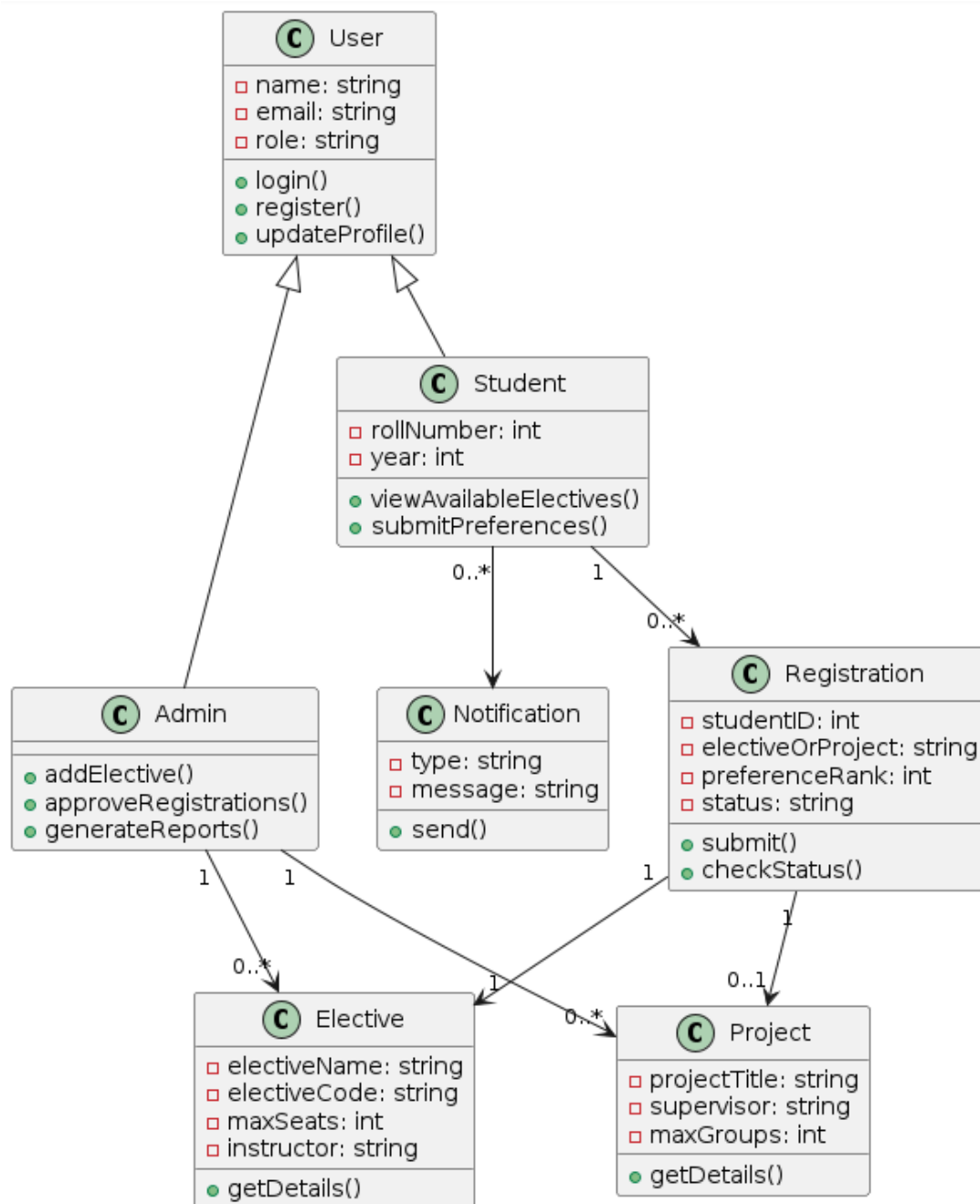
- Student receives confirmation and can view registered electives/projects.
- Admin dashboard updates seat availability.
- Reports can be generated for audit purposes.

5 Data View

5.1 Domain Model

- Entities: User, Elective, Project, Allocation, Preferences
- Relationships

- Allocation links Student and Elective/Project



5.2 Data Model (persistent data view)

The persistent data of the ScoreLens system is stored using **SQLite**, a lightweight relational database. This section outlines the structure of the database schema, representing how domain entities are stored in tables along with their relationships.

Table Name	Attributes	Description
Users	id (PK), name, email, role, courses	Stores user info (both students and professors). The role field distinguishes between them.
Students	user_id (PK, FK), roll_number	Stores student-specific data linked to a user.
Enrollments	user_id (FK), course_id (FK)	Many-to-many mapping between users and courses.
StudentDetails	id (PK), course_id (FK), student_name, student_marks	Stores marks and names for each student in a course.

5.2.1 Data Dictionary

Attribute	Data Type	Description	Constraints
id	INTEGER	Unique ID for the user	Primary Key
name	TEXT	Name of the user	Not Null
email	TEXT	Email address	Unique, Not Null
role	TEXT	Role of user (student/professor)	Not Null
courses	TEXT / JSON	List of associated course IDs (if any)	Optional

Students Table

Attribute	Data Type	Description	Constraints
user_id	INTEGER	Linked ID from Users table	Primary Key, Foreign Key (Users.id)
roll_number	TEXT	Unique roll number of student	Not Null, Unique

Professors Table

ssAttribute	Data Type	Description	Constraints
user_id	INTEGER	Linked ID from Users table	Primary Key, Foreign Key (Users.id)

Courses Table

Attribute	Data Type	Description	Constraints
id	INTEGER	Unique course ID	Primary Key
name	TEXT	Course name	Not Null
course_number	TEXT	Code for the course (e.g., CS101)	Unique
professor_id	INTEGER	Linked to professor creating the course	Foreign Key (Professors.user_id)

Enrollments Table

Attribute	Data Type	Description	Constraints
user_id	INTEGER	User enrolled (student or professor)	Foreign Key (Users.id)
course_id	INTEGER	Course enrolled in	Foreign Key (Courses.id)
			Composite Primary Key (user_id, course_id)

StudentDetails Table

Attribute	Data Type	Description	Constraints
id	INTEGER	Unique record ID	Primary Key
course_id	INTEGER	Associated course	Foreign Key (Courses.id)
student_name	TEXT	Name of the student	Not Null

6 Exception Handling

This section describes the exceptions that may occur within the application, the conditions under which they arise, how they are handled and logged, and any necessary follow-up actions. Proper exception handling ensures the system remains robust, user-friendly, and secure during unexpected events.

6.1 Authentication Failure

Circumstance: User enters invalid credentials during login.

Handling: The system displays an error message indicating incorrect username or password.

Logging: Failed login attempts are recorded with a timestamp and IP address.

Follow-up Action: User is prompted to retry or reset their password after a limited number of failed attempts.

6.2 Data Retrieval Error

Circumstance: Failure to fetch student scores due to database issues or network problems.

Handling: An error message is shown indicating that data could not be loaded.

Logging: Error details, including query failure or timeout messages, are logged.

Follow-up Action: System retries once or suggests the user to refresh the page.

6.3 Visualization Rendering Failure

Circumstance: Chart rendering fails due to invalid or missing data.

Handling: The system notifies the user with a message like "Unable to generate chart. Please check selected filters."

Logging: The exception is logged with the dataset and visualization parameters used.

Follow-up Action: User is advised to modify filters or contact support if the issue persists.

6.4 Access Denied

Circumstance: A student tries to access admin-only features or unauthorized data.

Handling: The system redirects to an error page with a "Permission Denied" message.

Logging: The unauthorized access attempt is logged with user details.

Follow-up Action: Admin may be notified if repeated violations are detected

7 Configurable Parameters

Configuration Parameter Name	Definition and Usage	Dynamic?
maxLoginAttempts	Maximum number of failed login attempts before temporary lockout.	Yes
sessionTimeoutMinutes	Duration (in minutes) before an inactive user is logged out.	No
dataRefreshInterval	Interval (in seconds) for auto-refresh of score data.	Yes
databaseRetryCount	Number of retry attempts for failed database queries.	No

8	Quality of Service		
----------	---------------------------	--	--

8.1 Availability

High availability ensured through cloud deployment and daily backups.

8.2 Security and Authorization

- Role-based access control (RBAC)
- Encrypted login using OAuth2

8.3 Load and Performance Implications

- Expected to handle ~1000 concurrent students during peak registration
- Indexed DB tables for fast lookups

8.4 Monitoring and Control

- Logs student interactions
- Admin dashboard for system monitoring
- Email alerts for system failures