Software Design Specifications for

Registration of elective courses or Project

Prepared by:

Shiva Siddhartha se22uari063

Shiva Ranjani Se22uari158

Darshan se22uari041

Dharani se22uari149

**Document Information**

**Title:** Registration of elective courses or projects Document version No: 1.0

Document version Date: 9th April 2025

Prepared By- shiva siddhartha, shiva ranjani , dharani, darshan Preparation date: 9th April 2025

**Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ver.No** | **Ver.Date** | **Revised by** | **Description** | **File name** |
| **1.0** | **9th April 2025** | **Shiva siddhartha** | **Initial version** | **SDD\_REGISTRATION\_OF\_ELECTIVE\_COURSES\_OR\_PROJECTS.pdf** |

**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 | D ATA 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 Software Design Specification is to define the design architecture of the Registration of Elective Courses or Project system within the overall context of the academic course management solution. It outlines how each component of the system is structured and how these components interact. The document is primarily intended for software developers, testers, and project maintainers who are responsible for building and maintaining the system. It also serves as a reference for stakeholders who need to understand the technical aspects of the system.

# 1.2 Scope

# This document describes the design for a web-based application that allows students to register for elective courses or final year projects. The system includes functionality for user authentication, elective/course listing, registration workflows, administrative control, and reporting features. The SDS focuses on the system’s architecture, module descriptions, data flow, and interface design. It does not cover hardware specifications or operational manuals.

# 1.3 Definitions, Acronyms, Abbreviations

# SRS – Software Requirements Specification

# UI – User Interface

# DB – Database

# API – Application Programming Interface

# 1.4 References

• IEEE Software Design Specification Template

• UML Diagrams: Use Case, Class, Sequence

# 2. Use Case View

A diagram of a diagram

AI-generated content may be incorrect.

### Use Case

**2.1.1 User Authentication**

**1.** **Actors**: Student, Admin, Faculty

**2**. **Description**: Ensures secure access to the system by validating user credentials.

**3**. **Steps:**

* User enters login ID and password.
* System checks credentials**.**
* Access is granted based on user role**.**
* Invalid login shows error: “Invalid username or password.”

**2.1.2 Register for Elective Course**

**1. Actors:** Student

**2. Description:** Allows students to view and register for available elective

courses

**3. Steps:**

* Student logs in.
* Checks available courses.
* System verifies prerequisites and availability.
* Student selects and registers for a course.
* Confirmation is displayed.

**2.1.3 Choose Available Projects**

1. **Actors:** Student
2. **Description:** Students can browse and choose from available final year projects.
3. **Steps:**

* Student logs in and navigates to project selection.
* Views available projects based on department.
* Selects a project and submits choice.
* Await faculty approval**.**

**2.1.4 Faculty Approval**

1. **Actors:** Faculty
2. **Description:** Faculty reviews and approves/rejects student project registrations.
3. **Steps:**

* Faculty logs in.
* Views list of students requesting project approval.
* Reviews project details and approves/rejects.
* System notifies student.

**2.1.5 Assign Electives**

1. **Actors:** Admin
2. **Description:** Admin assigns students to electives after verifying requirements.
3. **Steps:**

* System verifies eligibility.
* Assigns electives to students.
* Updates course registration status.

**2.1.6 Manage Elective Courses**

1. **Actors:** Admin
2. **Description:** Admin can add, edit, or remove elective courses.
3. **Steps:**

* Admin logs in.
* Navigates to elective course management panel.
* Modifies course information.
* Saves and publishes updates.

**2.1.7 Generate Reports**

1. **Actors:** Admin, Faculty
2. **Description:** Generate reports on registrations by course, student, or project.
3. **Steps:**

* User selects report type and filters.
* System fetches and displays results.
* Option to export to PDF/CSV**.**

**2.1.8 View Assigned Students**

1. **Actors:** Faculty
2. **Description:** Faculty can view students assigned to their courses or projects.
3. **Steps:**

* Faculty logs in.
* Navigates to assigned students list.
* System displays registered student details**.**

# Design Overview

### Design Goals and Constraints

### Goals

* Ensure intuitive and secure user interfaces for different roles (Student, Faculty, Admin).
* Enable dynamic registration of elective courses and final year projects.
* Provide real-time status updates and approval workflows.
* Allow scalable architecture to support additional departments or course types.

### Constraints

* System must be web-based and accessible via standard browsers.
* Must integrate with the institution’s existing student information system.
* All transactions (registrations, approvals) must be completed within academic deadlines.
* Use of open-source frameworks and databases preferred to minimize cost.

### Design Assumptions

* Database and authentication infrastructure exists
* Faculty and admins are familiar with the system
* Roles: Student, Faculty, Admin

### Significant Design Packages

* **Authentication Package:** Manages login, logout, and user role verification.
* **Elective Management Package:** Handles CRUD operations for courses and student elective registrations.
* **Project Management Package:** Supports project listing, student selection, and faculty approval.
* **Reporting Package:** Generates analytics and reports for admin and faculty use.
* **Notification Package:** Sends confirmation emails or dashboard notifications upon status changes.
* **Database Access Layer:** Interfaces with the DB to store/retrieve information.

### Dependent External Interfaces

The table below lists the public interfaces this design requires from other modules or applications

|  |  |  |
| --- | --- | --- |
| **External Application And Interface Name** | **Module Using The Interface** | **Functionality/ Description** |
| Email Server API(SendGrid) | Notification Module | Sends the email confirmation about the course and project registration |
| University Database | Authentication Module | Verifies the User |

### 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** |
| POST /register | Authentication Module | Registers a new user (student or faculty) with validation and stores hashed credentials. |
| POST /login | Authentication Module | Authenticates users and returns a JWT token for session management. |
| GET /courses | Course Module | Retrieves the list of available elective courses for selection. |
| POST /select-course | Selection Module | Allows a student to select an elective course (within eligibility and availability constraints). |
| GET /my-courses | Selection Module | Displays courses currently selected by the logged-in student. |
| POST /allocate-project | Project Allocation Module | Allows faculty to allocate a project topic to a student. |
| GET /project-allocations | Project Allocation Module | Displays all project allocations for faculty or students. |
| POST /submit-feedback | Feedback Module | Allows students to submit feedback for a course or project. |
| GET /feedback/:courseId | Feedback Module | Allows faculty/admin to view feedback submitted for a particular course. |
| GET /admin/registrations | Admin Module | Displays all student registrations for electives and projects (admin only). |

# 

# 4. Logical View

# Design Model

# Student

# Responsibilities: Register for elective courses or project, view personal profile

# Attributes: studentid, name, email

# Operations: registerCourse(), viewProfile()

# Relationships: Linked to Course and Project classes

# Faculty

# Responsibilities: Approve student projects, manage course-related activities

# Attributes: facultyid, name, email

# Operations: approveProject(), viewStudents()

# Relationships: Associated with Project and Course classes

# Admin

# Responsibilities: Manage system settings, user accounts, and overall control

# Attributes: adminid, name, email

# Operations: manageUsers(), updateSettings()

# Relationships: Oversees interactions with all user types (Student, Faculty)

# Course Managing

# Responsibilities: Handle creation, assignment, and updates of courses

# Attributes: courseList, facultyAssignments

# Operations: addCourse(), assignFaculty()

# Relationships: Connects Faculty to Student via course management

# Notification Service

# Responsibilities: Send system notifications and alerts

# Attributes: senderEmail, smtpSettings

# Operations: sendNotification()

# Relationships: Supports all modules for communication purposes

# Use Case Realization

# Use Case 1 – User Authentication

# User submits login credentials via the interface.

# System sends credentials to Authentication Service.

# Authentication Service verifies with the database.

# System grants access or returns an error message.

# Sequence diagram:

# 

# Use Case2 – Register for Elective Courses

# Student selects a course through the interface.

# Request is sent to Registration Manager (admin) for validation.

# Registration Manager confirms eligibility and availability.

# System registers the student and sends a confirmation.

# Sequence diagram:

# 

# Use Case 3 – Project Approval by Faculty

# Student submits a project request.

# Project Approval Manager stores and displays the request to faculty.

# Faculty reviews and makes a decision.

# System records the response and notifies the student.

# Sequence diagram:

# 

# Use Case4 – Admin Managing Courses and Reports

# Admin logs in and accesses course or report management.

# Admin adds/edits data or selects report filters.

# Course Manager or Report Generator processes the action.

# System saves changes or displays the report.

# Sequence diagram:

# 

# Use Case 5 – Notification on Status Updates

# A key event occurs (e.g., course registration or project approval).

# The responsible module triggers the Notification Service.

# Notification Service prepares and sends a message (email/SMS).

# The system stores a log of the notification activity.

# Sequence diagram:

# 

# Activity Diagram:

# 

1. **Data View**
   1. **Domain Model:**
      1. **Student**
         1. **Attributes:**
      * StudentID, Name, Email, Department, Year, Password
        1. **Relationships:**
        + Registers for one or more Courses
        + May submit one Project
        1. **Description:**
      * Represents a student user who can log into the system to register for elective courses or submit a project.
      1. **Faculty**
         1. **Attributes:**
      * FacultyID, Name, Email, Department, Designation, Password
        1. **Relationships:**
      * Can approve or review Projects
      * Can monitor students registered for their courses
        1. **Description:**
      * A user with authority to manage academic components like approving projects and overseeing courses.
      1. **Admin**
         1. **Attributes:**
      * AdminID, Name, Email, Password
        1. **Relationships:**
      * Manages the system including Courses, Students, Faculty, and Projects
        1. **Description:**
      * Handles administrative controls such as adding/removing users, managing course/project catalogues.
      1. **Course**
         1. **Attributes:**
      * CourseID, Title, Department, Credits, Semester, FacultyID
        1. **Relationships:**
      * Can be registered by multiple Students
      * Is handled by one Faculty
        1. **Description:**
      * Elective courses offered by departments, which students can register for.
      1. **Project**
         1. **Attributes:**
            * ProjectID, Title, Description, StudentID, FacultyID
         2. **Relationships:**
            * Submitted by one Student
            * Reviewed by one Faculty
         3. **Description:** 
            * Final year or elective project work submitted by a student and assessed by faculty.
   2. **. Data Model (persistent data view)**
      1. **Data Dictionary:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Entity** | |  | | --- | |  |  |  | | --- | | **Attributes** | | **Data Type** | **Description** |
| Student | |  | | --- | |  |  |  | | --- | | StudentID  Name  Email  Department  Year  Password | | Varchar2()  Varchar2()  Varchar2()  Varchar2()  Int  Varchar2() | Primary key  Name of the student  Email id of student  Student department  Student academic year  Student password |
| Faculty | FacultyID  Name  Email  Department  Designation  Password | Varchar2()  Varchar2()  Varchar2()  Varchar2()  Varchar2()  Varchar2() | Primary key  Faculty name  Faculty mail id  Faculty department  Designation of faculty  Faculty password |
| Admin | |  | | --- | |  |  |  | | --- | | AdminID  Name  Email  Password | | Varchar2()  Varchar2()  Varchar2()  Varchar2() | Primay key  Admin name  Admin mail id  Admin password |
| Course | CourseID  Title  Department  Credits  Semester  FacultyID | Varchar2()  Varchar2()  Varchar2()  Int  Int  Varchar2() | Primary key  Course title  Department of course  No of credits  Semester  Id of the faculty |
| Project | ProjectID  Title  FacultyID | Varchar2()  Varchar2()  Varchar2() | Primary key  Title of the project  Id of the faculty |

# 6. Exception Handling

# Invalid Login Exception

# Circumstances: When a user enters incorrect credentials.

# Handling Method: Display an error message.

# Logging & Follow-up Action: Log failed attempts with a timestamp and alert the admin if repeated

# Course Limit Exceeded Exception

# Circumstances: When a student tries to register for more electives than allowed

# Handling Method: Show a notification and prevent registration

# Logging & Follow-up Action: Log the student ID and course attempted

# Database Connection Exception

# Circumstances: When backend database connection fails

# Handling Method: Use a retry mechanism

# Logging & Follow-up Action: Log system error with stack trace and alert technical support

# Invalid Input Exception

# Circumstances: When users submit forms with missing or incorrect data

# Handling Method: Show validation messages and stop form submission

# Logging & Follow-up Action: Log the form source and user ID; guide the user to correct the input

# 7. Configurable Parameters

|  |  |  |
| --- | --- | --- |
| Configuration Parameter Name | Definition and Usage | Dynamic? |
| MAX\_LOGIN\_ATTEMPTS | Defines the maximum number of consecutive failed login attempts before lockout. | No |
| SESSION\_TIMEOUT\_MINUTES | Specifies the time (in minutes) after which inactive sessions will be terminated. | No |
| MAX\_ELECTIVE\_COURSES | Maximum number of elective courses a student can register for in a semester. | Yes |
| DB\_CONNECTION\_RETRY\_LIMIT | Number of times the system retries connecting to the database before throwing error. | No |
| ENABLE\_EMAIL\_NOTIFICATIONS | Enables or disables email notifications for system events. | Yes |

**8. Quality of Service**

**8.1 Availability**

This application is designed to ensure high availability through:

* System supports high availability with database replication
* **Impact Considerations:** Operations such as bulk data imports or system-wide updates are restricted during peak usage hours to prevent service disruptions.

**8.2 Security and Authorization**

The system incorporates robust security and role-based authorization to protect data access:

* **User Roles:** Admin, Faculty, and Student roles each have defined permissions and access scopes.
* **Authentication:** Users log in with secure credentials, and session handling
* **Data Access Control:** Sensitive operations are restricted to authorized users only.

**8.3 Load and Performance Implications**

The application is optimized for efficiency under expected usage conditions:

* **Expected Transactions:** Supports up to 500 concurrent users during peak registration periods.
* **Database Load:** Efficient indexing and query optimization used to handle large volumes of student and course data.
* **Response Time Goals:** Critical operations such as course registration or login complete within 2 seconds on average.

**8.4 Monitoring and Control**

To ensure the system remains healthy and responsive:

* **Logs:** System maintains structured logs for all operations, accessible through the admin panel.
* Admin dashboard shows registration statistics