SOFTWARE REQUIREMENT SPECIFICATIONS

FOR

REGISTRATION OF ELECTIVE COURSES AND PROJECTS

Prepared by

Group Name: **Team 34**

K. Shiva Siddhartha	**se22uari063**	**se22uari063@mahindrauniversity.edu.in**
N. Shiva Ranjani	**se22uari158**	**se22uari0158@mahindrauniversity.edu.in**
K. Dharshan	**se22uari041**	**se22uari041@mahindrauniversity.edu.in**
S. Dharani	**se22uari149**	**se22uari149@mahindrauniversity.edu.in**

Instructor:	***Avinash Arun Chauhan***
Course:	**Software Engineering**
Lab Section:	***IT2***
Teaching Assistant:	***Swapna***
Date:	**10- March-2025**

Contents

CONTENTS ..... **II**

REVISIONS ..... ERROR! BOOKMARK NOT DEFINED.

1 INTRODUCTION **1**

1.1 DOCUMENT PURPOSE 1
1.2 PRODUCT SCOPE 1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW 1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS 1
1.5 DOCUMENT CONVENTIONS 2
1.6 REFERENCES AND ACKNOWLEDGMENTS 2

2 OVERALL DESCRIPTION **2**

2.1 PRODUCT OVERVIEW 2
2.2 PRODUCT FUNCTIONALITY 3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS ..... **ERROR! BOOKMARK NOT DEFINED.**
2.4 ASSUMPTIONS AND DEPENDENCIES 3

3 SPECIFIC REQUIREMENTS **4**

3.1 EXTERNAL INTERFACE REQUIREMENTS 4
3.2 FUNCTIONAL REQUIREMENTS 4
3.3 USE CASE MODEL 5

4 OTHER NON-FUNCTIONAL REQUIREMENTS **11**

4.1 PERFORMANCE REQUIREMENTS 11
4.2 SAFETY AND SECURITY REQUIREMENTS 12
4.3 SOFTWARE QUALITY ATTRIBUTES 12

5 OTHER REQUIREMENTS **12**

APPENDIX A – DATA DICTIONARY **13**

APPENDIX B - GROUP LOG **13**

1 Introduction

1.1 Document Purpose

This document outlines the software requirements for the Registration of Elective Courses or Projects system. The purpose of this system is to provide students with an efficient and user-friendly platform to register for elective courses or projects offered by their institution. This SRS covers the functional and non-functional requirements, design constraints, and other relevant details necessary for the development of the system.

1.2 Product Scope

The Registration of Elective Courses or Projects system is designed to streamline the process of course and project registration for students. It will allow students to view available electives, check prerequisites, register for courses, and receive confirmation of their registration. The system will also provide administrative tools for faculty to manage course offerings, monitor enrollment, and generate reports. The primary benefit of this system is to reduce manual effort, minimize errors, and improve the overall efficiency of the registration process.

1.3 Intended Audience and Document Overview

This document is intended for the following audiences:

Developers: To understand the system requirements and design the software accordingly.

Project Managers: To oversee the development process and ensure that the project meets its objectives.

Students and Faculty: To understand the functionality and features of the system.

Testers: To verify that the system meets the specified requirements.

The document is organized into sections that cover the overall description, specific requirements, and non-functional requirements. Readers are encouraged to start with the introduction and product overview before diving into the detailed requirements.

1.4 Definitions, Acronyms and Abbreviations

SRS: Software Requirements Specification

UI: User Interface

API: Application Programming Interface

DBMS: Database Management System

Elective Course: A course that students can choose to take as part of their degree program.

Prerequisite: A course or requirement that must be completed before enrolling in another course.

1.5 Document Conventions

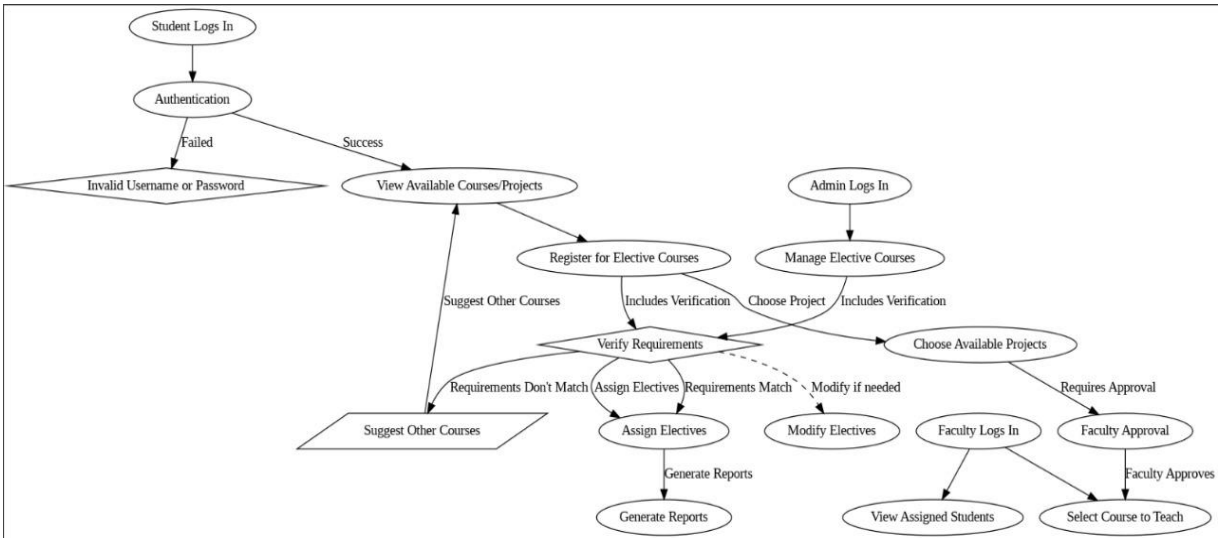
This document follows the IEEE formatting standards. The font used is Arial, size 11 or 12. Section titles are bolded, and subsections are italicized. Comments are written in italics.

1.6 References and Acknowledgments

- *IEEE Template for Software Requirements Specification.*
- *Drawi.io for use case Diagram.*
- *SmartDraw for Flow chart*

2 Overall Description

2.1 Product Overview



The Registration of Elective Courses or Projects system is a web-based application that allows students to register for elective courses or projects offered by their institution. The system will integrate with the existing student information system to retrieve student data and course information. The system will provide a user-friendly interface for students to browse available courses, check prerequisites, and register for courses. Faculty members will have access to an administrative interface to manage course offerings and monitor enrollment.

2.2 Product Functionality

The system will provide the following major functions:

- **Student Registration**: Students can register for elective courses or projects.
- **Course Browsing**: Students can view available courses, including course descriptions, prerequisites, and schedules.
- **Prerequisite Checking**: The system will automatically check if students meet the prerequisites for a course.
- **Administrative Tools**: Faculty can add, modify, or remove courses, monitor enrollment, and generate reports.
- **Confirmation and Notifications**: Students will receive confirmation of their registration and notifications about any changes to their registered courses.

2.3 Design and Implementation Constraints

- The system must be developed using the **COMET** method for software design and **UML** for modeling.
- The system must integrate with the existing student information system.
- The system must be accessible via web browsers on both desktop and mobile devices.
- The system must comply with institutional data security and privacy policies.

2.4 Assumptions and Dependencies

- *The system assumes that student data and course information will be provided by the existing student information system.*
- *The system depends on the availability of the student information system's API for data retrieval.*
- *The system assumes that students will have access to the internet and a compatible web browser.*

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Student Interface**: A web-based interface where students can browse courses, check prerequisites, and register for courses.
- **Faculty Interface**: A web-based interface where faculty can manage course offerings, monitor enrollment, and generate reports.

3.1.2 Hardware Interfaces

- The system will be accessible via standard web browsers on desktop and mobile devices.
- The system will interact with the institution's student information system via API.

3.1.3 Software Interfaces

- The system will integrate with the student information system to retrieve student data and course information.
- The system will use a database to store registration data and course information.

3.2 Functional Requirements

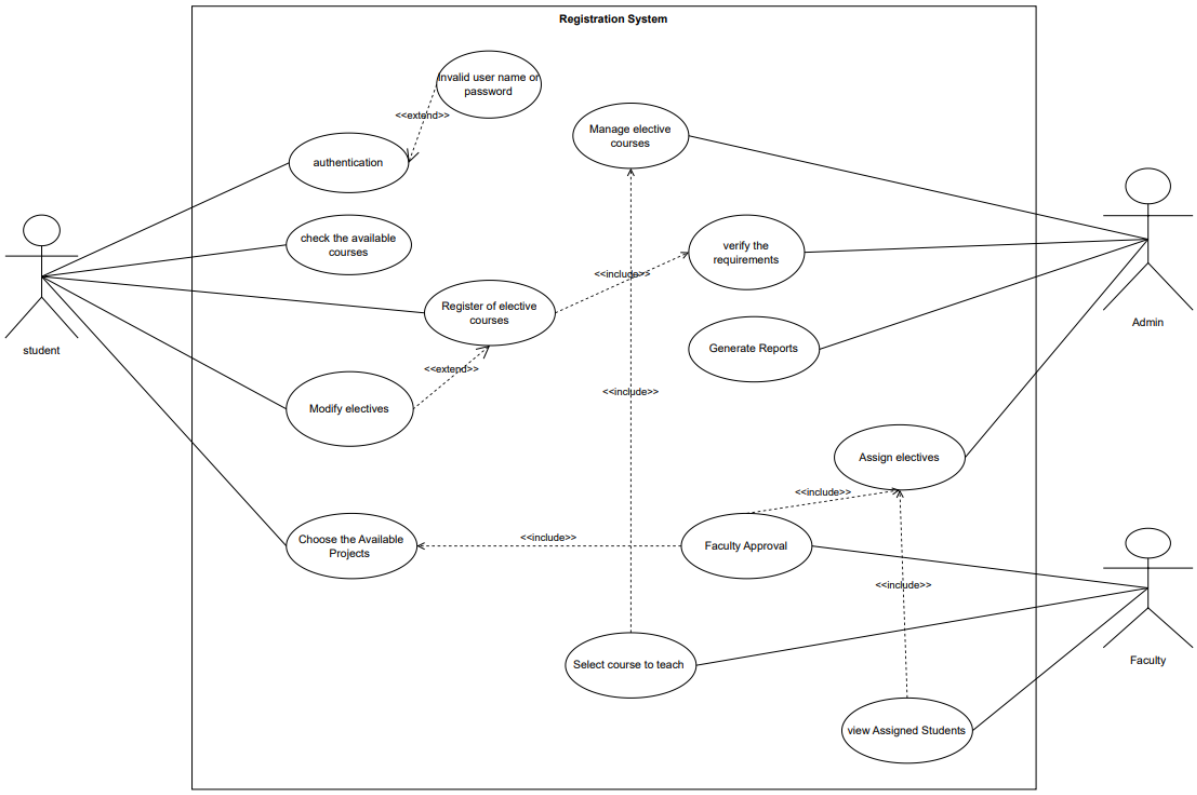
F1: Course Browsing
- The system shall allow students to browse available elective courses and projects.
- The system shall display course descriptions, prerequisites, and schedules.

F2: Prerequisite Checking
- The system shall automatically check if a student meets the prerequisites for a course before allowing registration.

F3: Course Registration
- The system shall allow students to register for elective courses or projects.
- The system shall provide confirmation of registration via email or on-screen notification.

F4: Administrative Tools
- The system shall allow faculty to add, modify, or remove courses.
- The system shall allow faculty to monitor enrolment and generate reports.

3.3 Use Case Model



3.3.1 Use Case #1 (Authentication U1)

Author: Darshan
Purpose: To allow students to log in securely and access the system.
Requirements Traceability: R1 – User authentication, R2 – Secure access
Priority: High
Preconditions:

- The student must have a registered account.
- The system must be online.

Postconditions:

- If successful, the student gains access to the system.
- If unsuccessful, the system displays an error message.

Actors: Student
Extends: None
Flow of Events:

1. **Basic Flow:**
    - The student enters a username and password.
    - The system verifies the credentials.
    - If valid, the student is granted access.
2. **Alternative Flow:**
    - If the student forgets the password, they can reset it via email.

3. **Exceptions:**
    - The system is down and cannot authenticate users.
    - The student enters incorrect credentials multiple times, leading to account lockout.

Includes: None
Notes/Issues: Implement CAPTCHA for added security.

3.3.2 Use Case #2 (Invalid Username or Password U2)

Author: Darshan
Purpose: To handle authentication failures due to incorrect login credentials.
Requirements Traceability: R3 – Error handling for login failures
Priority: High
Preconditions:

- The student attempts login.

Postconditions:

- The system displays an error message.
- The student is prompted to re-enter credentials.

Actors: Student
Extends: Authentication (U1)
Flow of Events:

1. **Basic Flow:**
    - The student enters an incorrect username or password.
    - The system displays an error message.
    - The student tries again.
2. **Alternative Flow:**
    - After multiple failed attempts, the account is locked.
3. **Exceptions:**
    - The system mistakenly locks the account due to a bug.

Includes: None
Notes/Issues: Implement an account recovery mechanism.

3.3.3 Use Case #3 (Check the Available Courses U3)

Author: Shiva Siddhartha
Purpose: Allows students to view a list of available courses before registration.
Requirements Traceability: R4 – Course listing functionality
Priority: Medium
Preconditions:

- The student is authenticated.

Postconditions:

- The system displays the available courses.

Actors: Student
Extends: None
Flow of Events:

1. **Basic Flow:**
    - The student selects the "View Courses" option.
    - The system fetches and displays the list of courses.
2. **Alternative Flow:**
    - The student applies filters to find specific courses.
3. **Exceptions:**
    - The system fails to retrieve courses due to a database issue.

Includes: None
Notes/Issues: Ensure courses are updated dynamically.

3.3.4 Use Case #4 (Register for Elective Courses U4)

Author: Shiva Siddhartha
Purpose: Allows students to register for elective courses.
Requirements Traceability: R5 – Course registration functionality
Priority: High
Preconditions:
- The student is authenticated.
- The student meets the course prerequisites.

Postconditions:
- The student is enrolled in the selected course.

Actors: Student
Extends: None
Flow of Events:
1. **Basic Flow:**
    - The student selects courses from the list.
    - The system checks eligibility.
    - The student confirms registration.
2. **Alternative Flow:**
    - The student changes their mind and selects a different course.
3. **Exceptions:**
    - The course is full.
    - The student does not meet the prerequisites.

Includes: Verify the Requirements (U5)
Notes/Issues: Handle cases where courses have limited seats.

3.3.5 Use Case #5 (Verify the Requirements U5)

Author: Shiva Ranjani
Purpose: Ensures students meet the prerequisites before course registration and has minimum CGPA.
Requirements Traceability: R6 – Eligibility validation
Priority: High
Preconditions:
- The student has selected a course.

Postconditions:
- The system verifies eligibility.
- If successful, the student can proceed with registration.

Actors: System
Extends: None
Flow of Events:
1. **Basic Flow:**
    - The system checks the student's academic history.
    - If eligible, the student is allowed to register.
2. **Alternative Flow:**
    - The system suggests alternative courses if requirements are not met.
3. **Exceptions:**
    - Database error prevents verification.

Includes: None
Notes/Issues: Ensure accurate validation logic.

3.3.6 Use Case #6 (Modify Electives U6)

Author: Dharani
Purpose: Allows students to change their elective course selections before the deadline.
Requirements Traceability: R7 – Modify course registration
Priority: Medium
Preconditions:
- The student is authenticated.
- The student is already registered for at least one elective.

Postconditions:
- The system updates the student's course registration.

Actors: Student
Extends: None
Flow of Events:
1. **Basic Flow:**
    - The student accesses their registered electives.
    - The student removes an existing elective.
    - The student selects a new elective.
    - The system verifies the eligibility and updates the record.
2. **Alternative Flow:**
    - If the student removes a course but does not select a new one, a warning is displayed.
3. **Exceptions:**
    - The elective cannot be modified after the deadline.
    - The new elective is full.

Includes: Verify the Requirements (U5)
Notes/Issues: Provide notifications for deadline reminders.

3.3.7 Use Case #7 (Choose the Available Projects U7)

Author: Shiva Ranjani
Purpose: Allows students to select from the list of available projects.
Requirements Traceability: R8 – Project selection feature
Priority: High
Preconditions:
- The student is authenticated.

Postconditions:
- The student’s project selection is recorded.

Actors: Student
Extends: None
Flow of Events:
1. **Basic Flow:**
    - The student navigates to the project selection section.
    - The system displays available projects.
    - The student selects a project.
    - The system requests faculty approval.
2. **Alternative Flow:**
    - If no projects are available, the student can request to add one.
3. **Exceptions:**
    - The project is already assigned to the maximum number of students.

Includes: Faculty Approval (U8)
Notes/Issues: Ensure fair distribution of projects among students.

3.3.8 Use Case #8 (Faculty Approval U8)

Author: Dharani
Purpose: Faculty members review and approve student requests for electives or projects.
Requirements Traceability: R9 – Faculty approval system
Priority: High
Preconditions:
- The student has submitted a request for a project or elective.

Postconditions:
- The faculty approves or rejects the request.

Actors: Faculty
Extends: None
Flow of Events:
1. **Basic Flow:**
    - The faculty member views pending approval requests.
    - The faculty either approves or rejects the request.
    - If approved, the student is notified.
2. **Alternative Flow:**
    - If the faculty is unavailable, another faculty member can review requests.
3. **Exceptions:**
    - System errors cause pending requests to disappear.

Includes: Assign Electives (U9)
Notes/Issues: Implement automatic notifications for approvals and rejections.

3.3.9 Use Case #9 (Assign Electives U9)

Author: Shiva Ranjani
Purpose: Allows the admin to assign elective courses to students.
Requirements Traceability: R10 – Course assignment system
Priority: High
Preconditions:
- The student has submitted elective preferences.
- Faculty approval is obtained (if required).

Postconditions:
- The student is assigned to the elective course.

Actors: Admin
Extends: None
Flow of Events:
1. **Basic Flow:**
    - The admin reviews student course preferences.
    - The admin assigns electives based on availability.
    - The system updates student records.
2. **Alternative Flow:**
    - If demand is high, students are placed on a waiting list.
3. **Exceptions:**
    - The system fails to update elective assignments.

Includes: Faculty Approval (U8)
Notes/Issues: Ensure fair allocation of elective courses.

3.3.10 Use Case #10 (Manage Elective Courses U10)

Author: Darshan
Purpose: Allows the admin to create, update, or remove elective courses.
Requirements Traceability: R11 – Elective course management
Priority: High
Preconditions:

- The admin is authenticated.

Postconditions:

- The system updates the list of available electives.

Actors: Admin
Extends: None
Flow of Events:

1. **Basic Flow:**
    - The admin adds new elective courses.
    - The admin modifies or removes existing electives.
    - The system updates course offerings.
2. **Alternative Flow:**
    - If no new courses are added, the system retains previous offerings.
3. **Exceptions:**
    - Data entry errors result in incorrect course details.

Includes: None
Notes/Issues: Validate course details before updating.

3.3.11 Use Case #11 (Generate Reports (U11)

Author: Shiva Siddhartha
Purpose: Generates reports related to student registrations, faculty assignments, and course enrolments.
Requirements Traceability: R13 – Reporting functionality
Priority: Medium
Preconditions:

- The admin is authenticated.

Postconditions:

- The system generates and stores reports.

Actors: Admin
Extends: None
Flow of Events:

1. **Basic Flow:**
    - The admin selects report parameters.
    - The system retrieves data and generates reports.
2. **Alternative Flow:**
    - The admin schedules automated reports.
3. **Exceptions:**
    - The report fails due to missing data.

Includes: None
Notes/Issues: Provide export options for reports.

3.3.12 Use Case #12 (Select Course to Teach U12)

Author: Darshan
Purpose: Allows faculty members to select which courses they want to teach.
Requirements Traceability: R14 – Faculty course selection
Priority: Medium
Preconditions:

- The faculty member is authenticated.

Postconditions:

- The faculty is assigned to the selected course.

Actors: Faculty
Extends: None
Flow of Events:

1. **Basic Flow:**
    - The faculty browses available courses.
    - The faculty selects a course to teach.
    - The system assigns the faculty member to the course.
2. **Alternative Flow:**
    - If a course is unavailable, the faculty can request assignment.
3. **Exceptions:**
    - The course is already assigned to another faculty.

Includes: None
Notes/Issues: Consider setting a deadline for course selection.

3.3.13 Use Case #13 (View Assigned Students U13)

Author: Shiva Siddhartha
Purpose: Allows faculty to view the list of students assigned to their courses.
Requirements Traceability: R15 – Faculty access to student records
Priority: Medium
Preconditions:

- The faculty member is authenticated.

Postconditions:

- The system displays assigned students.

Actors: Faculty
Extends: None
Flow of Events:

1. **Basic Flow:**
    - The faculty accesses student lists.
    - The system retrieves and displays data.
2. **Exceptions:**
    - The student list fails to load due to an error.

Includes: None
Notes/Issues: Ensure data privacy compliance.

4 Other Non-functional Requirements

4.1 Performance Requirements

- The system shall handle up to 1,000 concurrent users without performance degradation.
- The system shall respond to user requests within 2 seconds under normal load.

4.2 Safety and Security Requirements

- The system shall comply with institutional data security policies.
- The system shall require user authentication (login) for access.
- The system shall encrypt sensitive data during transmission.

4.3 Software Quality Attributes

4.3.1 Usability

- *The system shall provide an intuitive user interface for both students and faculty.*
- *The system shall include help documentation and tooltips for users.*
-

4.3.2 Reliability

- *The system shall have an uptime of 99.9% during the registration period.*
- *The system shall provide error messages for invalid user actions.*

5 Other Requirements

Legal and Compliance Requirements

- *The system shall ensure that all user data is encrypted during transmission and storage.*
- *The system shall provide an audit trail for all user actions (e.g., logins, registrations, changes to course offerings).*
- *The system shall include a terms-of-use agreement and privacy policy that users must accept before using the system.*

Database Requirements

- *The system shall use a relational database management system (RDBMS) such as MySQL or MongoDB to store data.*
- *The database shall store the following data:*

  Student information (e.g., StudentID, Name, Email, Program).

  Course information (e.g., CourseID, CourseName, Prerequisites, Schedule).

  Registration records (e.g., StudentID, CourseID, RegistrationStatus).
- *The database shall enforce referential integrity (e.g., a student cannot register for a course that does not exist).*

Testing Requirements

- The system shall undergo unit testing, integration testing, and user acceptance testing (UAT) before deployment.
- The system shall include automated test scripts for regression testing.
- The system shall be tested for compatibility with major web browsers (e.g., Chrome, Firefox, Safari, Edge).

Appendix A – Data Dictionary

Variables	Type	Description
Student ID	String	Unique identifier for each student
Course ID	String	Unique identifier for each course
Perquisite	String	Course prerequisites
Registration Status	String	Status of course registration

Appendix B - Group Log

Meetings	**Date**	**Time**	**Topics Discussed**
1	*31-01-2025*	*1hr*	*Project initiation and roles assignment*
2	*06-02-2025*	*1hr*	*Discussed the project functionalities*
3	*01-03-2025*	*1hr*	*Initiated UI*