
Software Requirements Specification

for

University Course Registration System

Version 1.0 approved

Prepared by

Dhruv Varshney- A2305221157

Boddu Asmitha Bhavya A2305221386

Pratyaksha Sharma - A2305221652

V Sri Hrudya - A2305221030

7CSE6-X

Amity University

23/10/2024

Table of Contents

Table of Contents.....	ii
Revision History.....	ii
1. Introduction.....	3
1.1. Purpose.....	3
1.2. Document Conventions.....	2
1.3. Intended Audience and Reading Suggestions.....	2
1.4. Product Scope	2
1.5. References.....	3
2. Overall Description	3
2.1. Product Perspective.....	3
2.2. Product Functions	4
2.3. User Classes and Characteristics	4
2.4. Operating Environment.....	4
2.5. Design and Implementation Constraints	5
2.6. User Documentation	5
2.7. Assumptions and Dependencies	5
3. External Interface Requirements	6
3.1. User Interfaces	6
3.2. Hardware Interfaces	7
3.3. Software Interfaces	7
3.4. Communications Interfaces	8
4. System Features	8
4.1. Course Registration.....	8
4.2. Schedule Management	9
4.3. Grading and Student Management (Faculty)	10
4.4. User Management (Administrator)	11
5. Other Nonfunctional Requirements	12
5.1. Performance Requirements	12
5.2. Safety Requirements	12
5.3. Security Requirements	13
5.4. Software Quality Attributes	13
5.5. Business Rules	13
6. Other Requirements	14
6.1. Database Requirements.....	14
6.2. Internationalization Requirements	14
6.3. Legal and Compliance Requirements	14
6.4. Reusability Objectives	15
6.5. Future Growth and Scalability	15
Appendix A: Glossary.....	16
Appendix B: Analysis Models.....	18
Appendix C: To Be Determined List.....	21

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1. Purpose

This Software Requirements Specification (SRS) describes the requirements for the University Course Registration System (UCRS) version 1.0. The system enables candidates, students, faculty, and administrators to manage the processes related to course registration, scheduling, grading, and academic progress tracking. It includes functionality for secure user authentication, course management, and reporting, while ensuring compliance with security and data protection regulations.

1.2. Document Conventions

The SRS follows the IEEE standard for SRS documentation. The following conventions are used:

- **Bold** text for section headings.
- *Italics* for emphasis.
- Each requirement is assigned a unique identifier for easy reference.
- Role-based requirements are inherited across relevant user roles where applicable.

1.3. Intended Audience and Reading Suggestions

This document is intended for the following readers:

- **Developers:** For understanding system requirements and technical implementation.
- **Project Managers:** For planning and overseeing the development process.
- **Testers:** For validating that the system meets functional and non-functional requirements.
- **Client/ System Users:** To understand the system features based on their roles (candidates, students, faculty, administrators).

For a comprehensive understanding, readers should start with section 2 (System Requirements) and follow through based on their areas of interest.

1.4. Product Scope

The University Course Registration System (UCRS) facilitates seamless course registration, schedule management, and academic tracking for students. It enables faculty to manage course content, student enrollment, and grading. Administrators oversee user management, course management, and system reporting. The system supports secure data access, notifications, and communication, helping achieve efficient and effective academic operations within the university.

1.5. References

1. "IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1984 , vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205. keywords: {Software engineering;System analysis and design;software;requirements;specifications},
2. <https://studentprivacy.ed.gov/ferpa>
3. <https://www.meity.gov.in/writereaddata/files/Digital%20Personal%20Data%20Protection%20Act%202023.pdf>
4. Acharya, Kamal. (2023). ONLINE COURSE REGISTER SYSTEM PROJECT REPORT.. 10.13140/RG.2.2.36511.78249.
5. Y. Peng, N. Liu, Y. Li and Z. Shao, "Design and implementation of the online course registration system at Tsinghua University," 2012 International Conference on Systems and Informatics (ICSAI2012), Yantai, China, 2012, pp. 1179-1182, doi: 10.1109/ICSAI.2012.6223244.
6. W. Zhong, H. Chen, S. Xie, J. Yin and L. Dong, "Design and Construction of the New Web Learning System of Tsinghua University," 2022 12th International Conference on Information Technology in Medicine and Education (ITME), Xiamen, China, 2022, pp. 53-56, doi: 10.1109/ITME56794.2022.00022.

2. Overall Description

2.1. Product Perspective

The University Course Registration System (UCRS) is a new, self-contained product designed to facilitate course enrollment, schedule management, and academic progress tracking for

students, faculty, and administrators. It replaces manual or outdated registration systems currently in place. The URCS integrates with the university's existing Student Information System (SIS) and supports various user roles such as students, faculty, and administrators. The system also includes secure user authentication and data handling to comply with university policies and data protection laws.

The system interfaces with:

- **Student Information System (SIS)** for retrieving student records and academic progress.
- **Payment Gateways** for course fee payments (if applicable).

2.2. Product Functions

The main functions of the University Course Registration System include:

- **User Authentication:** Secure login, password recovery, and two-factor authentication (optional).
- **Student Module:** Course registration, schedule management, academic progress tracking, and GPA calculation.
- **Faculty Module:** Managing course details, student enrollment, grading, and attendance tracking.
- **Administrator Module:** Managing user accounts, courses, registration periods, and generating reports.
- **Notifications and Communication:** Sending email/SMS notifications and providing an in-app messaging system.
- **Security:** Role-based access control and secure data storage.

2.3. User Classes and Characteristics

- **Candidate:** Prospective students who have not yet registered for courses. Limited access to browse course offerings.
- **Student:** Registered students who can manage their course enrollments, academic schedules, and progress.
- **Faculty:** Professors and teaching staff responsible for course management, student enrollment, and grading.
- **Administrator:** System and academic administrators who manage users, courses, and system reports.

The most critical user classes are students and faculty, as they are the primary end users who will frequently interact with the system.

2.4. Operating Environment

The University Course Registration System will operate in the following environment:

- **Hardware:** Standard desktop and mobile devices used by students, faculty, and administrators.

- **Operating System:** Compatible with modern operating systems like Windows, macOS, iOS, Android, and Linux.
- **Software:** Compatible with the university's existing SIS, calendar systems (e.g., Google Calendar), and email systems.
- **Browser Support:** The system will support major web browsers such as Chrome, Firefox, Safari, and Edge.

2.5. Design and Implementation Constraints

The design of the UCRS must comply with the following constraints:

- **Security and Compliance:** The system must adhere to DPDP ACT and FERPA regulations for data protection and user privacy.
- **Integration with SIS:** The system must integrate seamlessly with the university's SIS to retrieve student records and academic data.
- **Performance Requirements:** The system must handle peak loads during registration periods without performance degradation.
- **Technologies:** Development will use modern web technologies like HTML5, CSS3, JavaScript, and frameworks like React or Angular for frontend, with a backend built using Node.js, Python, or a similar technology.
- **Databases:** The system will use an SQL or NoSQL database (e.g., MySQL, PostgreSQL, MongoDB) for data management.

2.6. User Documentation

The following user documentation will be provided with the system:

- **User Manuals** for students, faculty, and administrators outlining system features and usage.
- **Online Help** accessible within the system for quick guidance.
- **Video Tutorials** for students and faculty on how to register for courses, manage schedules, and use the system's features.

Documentation will be available in both PDF format and as web-based content for easy access.

2.7. Assumptions and Dependencies

- The UCRS assumes that users will have access to modern web browsers and internet connectivity.
- The system will rely on the existing SIS for retrieving accurate student records and academic information.
- The project assumes that all integrations with third-party systems, like payment gateways, will be functional and properly maintained.
- Any changes to regulatory data protection laws (such as DPDP ACT or FERPA) could require updates to the system's security features.

- The system depends on the continuous availability of email and SMS services for notifications and communication purposes.

3. External Interface Requirements

3.1. User Interfaces

The University Course Registration System (UCRS) will feature a user-friendly graphical user interface (GUI) that adheres to modern web design standards. The key characteristics of the user interfaces include:

- **Screen Layout:** Consistent layout across pages, with a top navigation bar for easy access to different sections such as Home, Profile, Courses, and Notifications.
- **Standard Buttons and Functions:**
 - **Help:** A dedicated help button available on every screen to access user manuals and FAQs.
 - **Back and Save Buttons:** Prominent on forms for user convenience.
 - **Error Messages:** Clear, concise error messages with suggestions for correction, displayed in red near the relevant input fields.

- **Keyboard Shortcuts:** Basic shortcuts for navigation (e.g., Ctrl+S for saving forms).
- **Sample screens would include:**
 - **Login Screen:** Username, password fields, "Forgot Password" link, and optional two-factor authentication.
 - **Course Registration Page:** Filterable list of courses, detailed course descriptions, and "Register" or "Drop" buttons.
 - **Student Dashboard:** Overview of registered courses, academic progress, and upcoming deadlines.

3.2. Hardware Interfaces

The URCS will be compatible with the following hardware devices:

- **Client Devices:**
 - **Desktop/Laptop:** Windows, macOS, and Linux operating systems.
 - **Mobile Devices:** Smartphones and tablets running iOS or Android.
- **Printers:** For users who need to print schedules, reports, or other documentation from the system.

No additional hardware interfaces are required beyond the standard input/output devices (keyboard, mouse, touchscreens for mobile users).

3.3. Software Interfaces

The URCS will integrate with several other software components, including:

- **Student Information System (SIS):**
 - Data exchanged: Student records, academic history, course offerings, and enrollment data.
 - Communication protocol: RESTful API for data retrieval and updates.
- **Calendar Systems** (e.g., Google Calendar, iCal):
 - Function: Exporting course schedules to personal calendars.
 - Communication protocol: iCal or Google Calendar API.
- **Email/SMS Gateway:**
 - Purpose: For sending notifications related to course registration, reminders, and system updates.
 - Communication protocol: SMTP for emails and an SMS API for text messages.
- **Payment Gateway** (if applicable for fee payment integration):
 - Function: Processing course fees.
 - Communication protocol: Secure payment APIs (e.g., PayPal, Stripe).
- **Database:**
 - Data exchanged: User information, course registrations, academic records.
 - The database will be an SQL-based system (e.g., MySQL, PostgreSQL).

3.4. Communications Interfaces

The URCS will require the following communication interfaces:

- **Web Browser:**
 - The system will be web-based, accessible via major browsers like Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
 - Protocols: HTTP/HTTPS will be used for secure web communications.
- **Email System:**
 - The system will send automatic email notifications for registration confirmations, reminders, and deadlines using SMTP.
- **Network Communication:**
 - The URCS will support LAN/WAN connectivity to communicate with other university systems, such as the SIS.
 - Secure communication protocols like SSL/TLS will be used to ensure data integrity and confidentiality.
- **Data Transfer:**
 - File exports (e.g., schedules or reports) will be available in formats such as PDF and CSV.
- **Security:**
 - Encryption mechanisms (e.g., SSL/TLS) will be used for all data transferred between the client and the server to protect sensitive information, such as login credentials and personal data.

4. System Features

4.1. Course Registration

4.1.1. Description and Priority

The Course Registration feature allows students to search, browse, and register for available courses during a specific registration period. This feature is critical for the system and is of High priority, as it is the core function for students.

- **Priority components:**
 - **Benefit:** 9
 - **Penalty:** 8 (If this feature is unavailable, students will be unable to register for courses, leading to delays in academic progress.)
 - **Cost:** 6
 - **Risk:** 4 (Moderate risk due to potential system overload during peak registration periods.)

4.1.2. Stimulus/Response Sequences

1. **Student Action:** Logs into the system and navigates to the "Course Registration" page.
System Response: Displays available courses based on department and program.
2. **Student Action:** Selects a course to view its details (e.g., description, schedule, prerequisites).
System Response: Displays course details.
3. **Student Action:** Registers for the selected course.
System Response: Confirms the registration and updates the student's course list.
4. **Student Action:** Attempts to register for a course that is full.
System Response: Adds the student to a waitlist and notifies them.
5. **Student Action:** Drops a course from their registered list.
System Response: Confirms the drop and updates the student's schedule accordingly.

4.1.3. Functional Requirements

- **REQ-1:** The system must allow students to browse courses by department, level, and availability status.
- **REQ-2:** The system must display course details, including prerequisites, credit hours, and schedule.
- **REQ-3:** The system must allow students to register for open courses and provide immediate confirmation of successful registration.
- **REQ-4:** If a course is full, the system must provide the option to join a waitlist.
- **REQ-5:** The system must allow students to drop courses before a specified deadline.
- **REQ-6:** The system must handle error conditions such as attempting to register for conflicting or restricted courses (e.g., missing prerequisites) by displaying appropriate error messages.

4.2. Schedule Management

4.2.1. Description and Priority

The Schedule Management feature enables students to view, export, and manage their academic schedule. It is of High priority because it helps students organize their coursework and avoid scheduling conflicts.

- **Priority components:**
 - **Benefit:** 8
 - **Penalty:** 6
 - **Cost:** 5
 - **Risk:** 3

4.2.2. Stimulus/Response Sequences

1. **Student Action:** Views their personal course schedule on the dashboard.
System Response: Displays the schedule, including class times and locations.
2. **Student Action:** Exports the schedule to their personal calendar.
System Response: Provides an option to download or sync with Google Calendar or iCal.
3. **Student Action:** Attempts to view schedule during system downtime.
System Response: Displays an error message notifying the student of maintenance.

4.2.3. Functional Requirements

- **REQ-1:** The system must allow students to view their personal course schedule in a user-friendly format.
- **REQ-2:** The system must support exporting the schedule to external calendar systems (e.g., Google Calendar, iCal).
- **REQ-3:** The system must automatically update the schedule when courses are added or dropped.
- **REQ-4:** The system must send reminders for upcoming classes and exams.

4.3. Grading and Student Management (Faculty)

4.3.1. Description and Priority

The Grading and Student Management feature allows faculty to manage course rosters, track attendance, and enter student grades. It is of Medium priority for the faculty user class but essential for maintaining academic records.

- **Priority components:**
 - **Benefit:** 7
 - **Penalty:** 5
 - **Cost:** 4
 - **Risk:** 3

4.3.2. Stimulus/Response Sequences

1. **Faculty Action:** Views the list of students enrolled in their course.
System Response: Displays the course roster.
2. **Faculty Action:** Inputs grades for assignments and exams.
System Response: Saves and calculates final grades based on predetermined weights.
3. **Faculty Action:** Tracks student attendance.
System Response: Updates the attendance record and displays alerts for excessive absences.

4.3.3. Functional Requirements

- **REQ-1:** The system must allow faculty to view and manage the list of enrolled students.
- **REQ-2:** The system must support the entry of student grades for assignments and exams.
- **REQ-3:** The system must automatically calculate final grades based on weights assigned to each assessment.
- **REQ-4:** The system must provide attendance tracking and alert faculty for students with low attendance rates.

4.4. User Management (Administrator)

4.4.1. Description and Priority

The User Management feature allows administrators to create, modify, and delete user accounts. This feature is of High priority as it is essential for system maintenance and role-based access control.

- **Priority components:**
 - **Benefit:** 9
 - **Penalty:** 7
 - **Cost:** 4
 - **Risk:** 3

4.4.2. Stimulus/Response Sequences

1. **Admin Action:** Creates a new user account (student, faculty, or administrator).
System Response: Adds the user and assigns a role.
2. **Admin Action:** Deletes a user account.
System Response: Removes the account and all associated data.
3. **Admin Action:** Updates a user's role from student to alumni.
System Response: Changes the user's role and updates permissions.

4.4.3. Functional Requirements

- **REQ-1:** The system must allow administrators to create new user accounts and assign roles.
- **REQ-2:** The system must allow administrators to modify user roles and permissions.
- **REQ-3:** The system must allow administrators to delete user accounts and associated data securely.
- **REQ-4:** The system must provide audit logs for all user account modifications made by administrators.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

- The system must support at least *1000 concurrent users* during peak registration periods, with a maximum response time of *3 seconds* for key transactions such as course search and registration.
- *Server uptime* must be maintained at *99.9%* during active registration periods to minimize downtime and ensure smooth operations.
- The system should be able to process *200 transactions per second* during peak periods to handle the expected volume of course registrations and updates.
- *Batch processing* (e.g., generating reports, running GPA calculations) should not impact the real-time performance of the course registration functionality and must be completed within *5 minutes* for each operation.

5.2. Safety Requirements

- The system must ensure the integrity of academic data, such as grades and course registrations, by implementing *regular data backups* every *24 hours* and ensuring a *disaster recovery system* with a recovery point objective (RPO) of *1 hour* and recovery time objective (RTO) of *2 hours*.
- Administrators must be alerted in the event of *system failures* or security breaches that could compromise user data or result in data loss.
- The system must follow the university's policies on *data retention and disposal*, ensuring that sensitive information is permanently deleted after the required retention period.

5.3. Security Requirements

- *Role-based access control (RBAC)* must be enforced to ensure that only authorized users (students, faculty, administrators) have access to appropriate system functions.
- The system must implement *strong encryption* (e.g., *AES-256*) for the storage and transmission of sensitive data, such as student personal information and login credentials.
- *Two-factor authentication* must be available for all users to enhance account security, especially for administrators and faculty members who handle critical data.
- The system must comply with *data protection regulations*, such as DPDP ACT for user privacy and FERPA for the protection of student records, ensuring that all access to sensitive data is logged and auditable.
- *Account lockout* must be triggered after *5 failed login attempts*, with an option for the user to recover their account through a secure password recovery process.

5.4. Software Quality Attributes

- **Usability:** The system should have an intuitive and user-friendly interface, ensuring that users can easily navigate through course registration, schedule management, and grading without extensive training.
- **Reliability:** The system must have a *mean time between failures (MTBF)* of at least *6 months*, ensuring stable and consistent operation during critical periods such as course registration and grading.
- **Maintainability:** The system should be easy to maintain, with well-documented code and architecture to allow updates or fixes to be deployed with *minimal downtime* (preferably during off-peak hours).
- **Interoperability:** The system must integrate seamlessly with the university's *Student Information System (SIS)*, allowing for real-time synchronization of data such as student enrolment status, grades, and course information.
- **Testability:** All system features must be thoroughly tested, with automated test suites for critical functionalities like course registration, grade management, and security features (e.g., login/authentication).
- **Scalability:** The system must be scalable to accommodate an increasing number of users, both in terms of student population growth and additional faculty or courses in the future.

5.5. Business Rules

- Only *students* can register for courses, view academic progress, and manage their schedule. *Faculty* are restricted to viewing and managing courses they are assigned to, entering grades, and tracking student attendance.
- *Administrators* have the exclusive right to create and manage courses, user roles, and system-wide settings, such as registration periods and course capacities.
- *Course registration* must follow university policies, including prerequisites and maximum enrollment limits. Students attempting to register for a course without

meeting prerequisites or when the course is full must be automatically placed on a waitlist.

- Students must only be allowed to drop courses within the *first two weeks of the semester* without penalty, as per university guidelines.

6. Other Requirements

6.1. Database Requirements

- The system must use a *relational database management system (RDBMS)*, such as *MySQL* or *PostgreSQL*, to store all user, course, and registration data.
- The database must support *ACID (Atomicity, Consistency, Isolation, Durability)* properties to ensure data integrity, particularly during registration and grading transactions.
- *Regular backups* of the database must be taken at intervals of *24 hours* during the semester and *every hour* during peak registration periods.
- The database schema must support *role-based access control* to ensure that students, faculty, and administrators have access only to the relevant data they are authorized to view or modify.
- The system must provide *encrypted storage* for sensitive information such as passwords, using a *secure hashing algorithm* like *bcrypt*.

6.2. Internationalization Requirements

- The system must support *multilingual* interfaces, with English as the default language. Additional language support, such as *Hindi* and *Tamil*, should be configurable via the user interface.
- Date formats, currency formats, and other region-specific data should follow the *locale settings* of the user, based on their geographic location or preference.
- The system should allow for the addition of new languages without major rework, by keeping *text strings external* from the core application code.

6.3. Legal and Compliance Requirements

The system must comply with *data protection laws*, including *FERPA* and *DPDP Act*, ensuring that students' personal data and academic records are securely managed and protected.

- The system should meet the university's internal policies and comply with national education standards as defined by government regulatory bodies such as the *University Grants Commission (UGC)* in India.

- The system must adhere to *accessibility standards*, such as *WCAG 2.1*, ensuring that the system is accessible to users with disabilities, including support for screen readers, keyboard navigation, and color contrast adjustments.

6.4. Reusability Objectives

- The system's architecture should allow for *modular components* (such as authentication, course management, and grading modules) to be reused in other university applications.
- The *APIs* developed for course registration and academic progress tracking should be designed for easy integration into other university systems, such as student mobile applications or alumni portals.

6.5. Future Growth and Scalability

- The system should be designed to scale both horizontally and vertically, supporting increased user load as the student population and course offerings expand.
- The system must allow for *easy expansion* of the database schema to accommodate new features or data types, such as new grading methods or additional types of academic programs.

Appendix A: Glossary

- **Authentication:** The process of verifying a user's identity before allowing access to the system, typically through a username and password, and optionally through two-factor authentication.
- **ACID:** A set of properties that guarantee reliable processing of database transactions (Atomicity, Consistency, Isolation, Durability).
- **API (Application Programming Interface):** A set of functions and procedures that allow other software systems to communicate and interact with the course registration system.
- **CAPTCHA:** A security measure used to prevent automated login attempts by requiring users to solve a challenge that is difficult for machines but easy for humans.
- **Course Registration:** The process by which students enroll in courses offered by the university for a given semester or academic term.
- **CSV (Comma-Separated Values):** A file format used to store tabular data, where each line is a record, and each field in the record is separated by a comma.
- **FERPA (Family Educational Rights and Privacy Act):** A U.S. law that governs the privacy of student education records.
- **GDPR (General Data Protection Regulation):** A European Union regulation that governs the privacy and security of personal data for individuals within the EU.
- **DPDP Act (Digital Personal Data Protection Act, 2023):** An Indian law that governs the processing of personal data, focusing on the privacy and protection of individuals' digital data. It requires consent for data processing, grants rights to data subjects, and outlines obligations for data controllers.
- **GPA (Grade Point Average):** A measure of a student's academic performance, calculated as a weighted average of grades received in courses.
- **GUI (Graphical User Interface):** The visual part of a software system that users interact with, typically involving buttons, menus, and forms.
- **iCal:** A calendar file format used to share and manage calendar data, often used by calendar applications like Google Calendar and Apple Calendar.
- **RBAC (Role-Based Access Control):** A security mechanism that restricts system access based on the roles of individual users, ensuring that only authorized users can access certain system features.
- **SRS (Software Requirements Specification):** A document that describes the functional and non-functional requirements of a software system, providing a detailed outline of what the system should do and how it should operate.
- **SIS (Student Information System):** A centralized system used by universities to manage student data, including registration, grades, and academic progress.
- **TA (Teaching Assistant):** A person, typically a graduate student, who assists faculty in teaching duties, including grading assignments, managing course materials, and answering student questions.
- **Two-factor Authentication (2FA):** An additional layer of security used to ensure that users are who they say they are by requiring two types of information—something they know (password) and something they have (a phone or token).
- **Waitlist:** A system feature that allows students to queue for enrollment in a course that is already full, with the possibility of registering if a seat becomes available.

Appendix B: Analysis Models

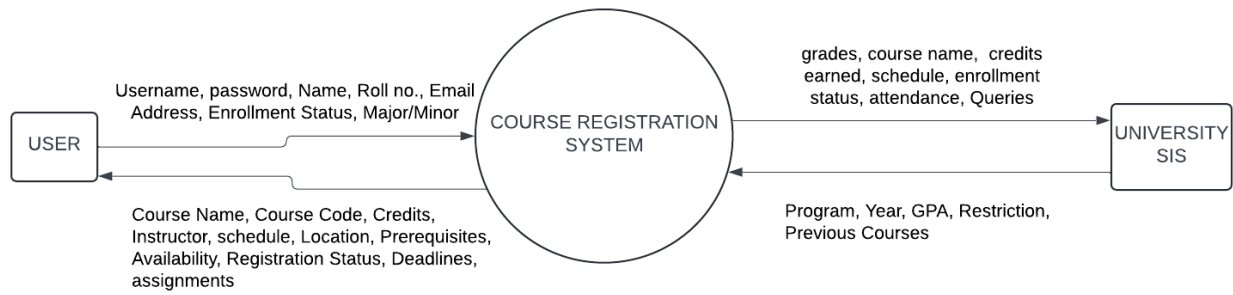


Fig 1 0-level Data Flow Diagram

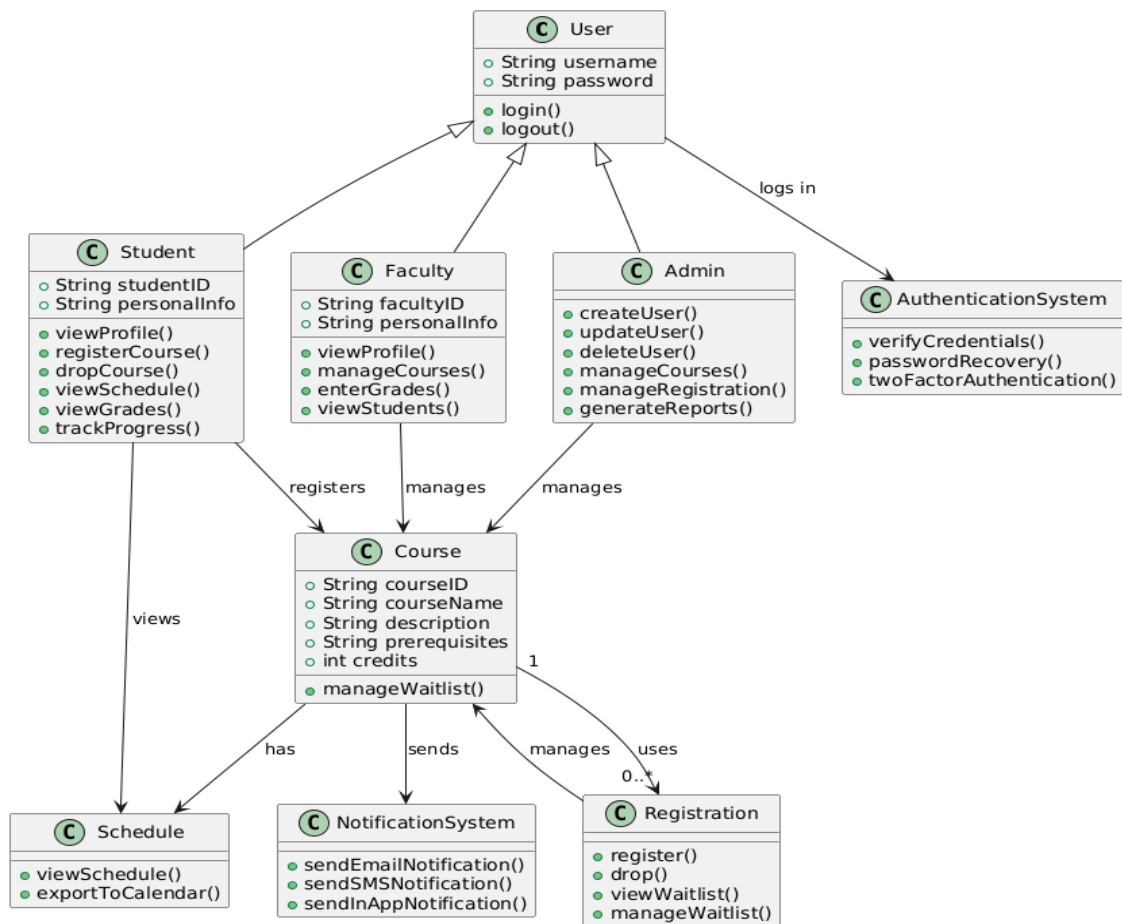


Fig 2 Class Diagram



Fig 3 Use Case Diagram

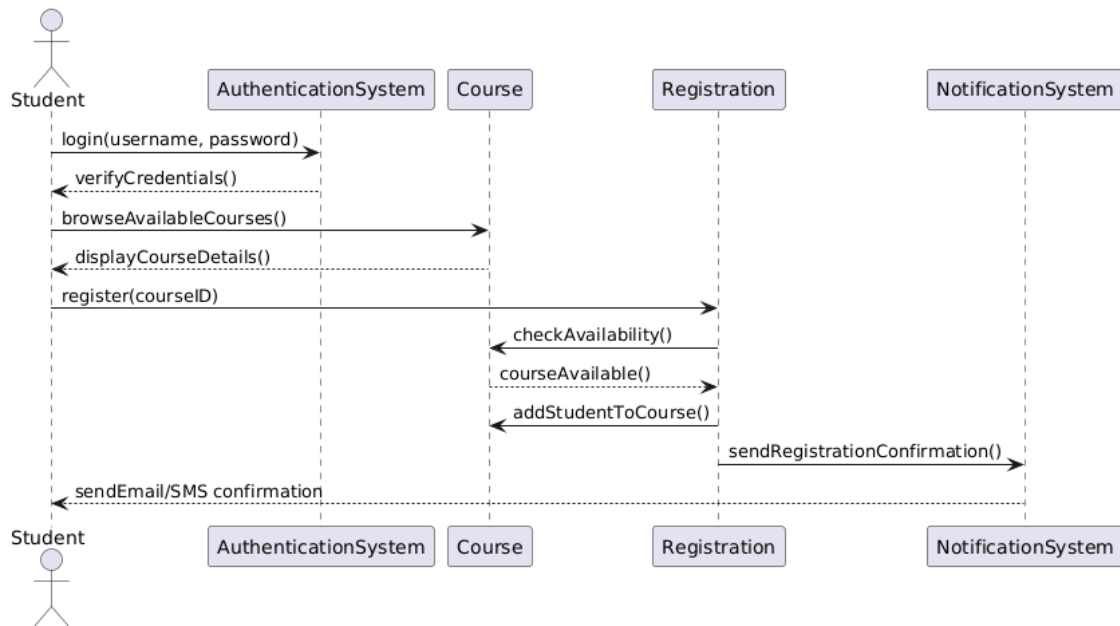


Fig 4 Sequence Diagram

Appendix C: To Be Determined List

1. Integration details with the university's payment gateway (Section 3.3).

2. Decision on supporting additional languages besides English (Section 6.2).
3. Confirmation of the final database management system (Section 6.1).
4. Final confirmation of the university's registration period rules (Section 5.5).
5. API documentation for Student Information System (SIS) integration (Section 3.3).