IT-313 Software Engineering Project
Student Information System

Guide : Prof. Saurabh Tiwari
Mentor : Vivek ######

Software Requirements Specification

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

A Student Information System (SIS) is a software application dedicated to overseeing and organizing student information within educational institutions. It serves as a comprehensive solution for managing various student-related processes, such as course and semester enrollment, attendance tracking, grading, and other administrative tasks. Beyond offering convenient access to student data, the SIS also streamlines communication with parents by providing information on their child's grade card and sending reminders about fee payment status. This system is strategically crafted to serve as a unified platform for storing and administering all aspects of student information, empowering faculty and administrators to access and update academic processes seamlessly through a centralized database.

1.1 Purpose

The Student Information System (SIS) endeavors to offer a comprehensive and unified software application capable of efficiently overseeing all facets of student data and administrative responsibilities within educational institutions. The SIS is crafted to simplify and automate diverse academic and administrative procedures, encompassing student enrollment, course registration, tracking of grades and attendance, program and semester management. The primary objective is to enhance the efficiency of these processes, diminish administrative workloads, and furnish faculty, students, and parents with precise and timely information. It centralizes the management of resources that were previously handled by manpower or staff.

1.2 Document Conventions

To ensure uniformity and clarity across the Software Requirements Specification (SRS) document, the following conventions have been established:

- The term "SIS" will consistently refer to the "Student Information System" throughout the document.
- "DB" will be used to denote the "Database" component of the system.
- The term "Admin" will specifically indicate the "Administrator" role within the system.
- Each functional or non-functional requirement will be uniquely identified with a sequence number or a meaningful tag.

Acronyms defined for reference:

- SIS: Student Information System
- DB: Database
- JS: JavaScript
- HTML: Hyper Text Markup Language
- CSS: Cascading Style Sheets
- EJS: Embedded JavaScript
- HTTP: Hypertext Transfer Protocol
- TCP: Transmission Control Protocol
- IP: Internet Protocol

1.3 Intended Audience and Reading Suggestions

The target audience for the SIS is extensive and varied, with each segment possessing distinct requirements and anticipations. The system grants students the ability to access their grades, attendance records, and course registration, enabling them to strategically plan their studies. Additionally, administrators utilize the system to oversee academic performance, allocate resources, and leverage data-driven insights for enhancing student outcomes.

1.4 Project Scope

The product scope of the SIS should be comprehensive, encompassing a full range of features and functionalities essential for the efficient and effective management of student data and activities. The objective is to deliver an enriched learning experience that caters to the needs of students and teachers. The system aims to provide a robust platform that not only streamlines administrative tasks but also enhances the overall educational journey for all stakeholders involved.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

The SIS stands as a purpose-built software application meticulously crafted to oversee the intricacies of student data management. This includes but is not limited to personal information, academic records, grades, and attendance. Recognizing the dynamic nature of educational institutions and their unique requirements, the SIS is adaptable to cater to a spectrum of needs. While specific demands can differ across institutions, there are overarching common needs that the SIS addresses. This versatility allows the system to be seamlessly tailored to the distinctive requirements of each educational entity it serves.

2.2 Product Features

1. Management of Student Records: The system is equipped to efficiently store and handle comprehensive student information, encompassing demographic details, contact information, academic records, and general particulars.

2. Tracking Personal Details: Students have the capability to view and request updates to their personal information. Any modifications become effective after administrative verification and subsequent updates in the database.

3. Grade Monitoring: The system adeptly tracks student grades, computes SPI, and allows students to download their results.

4. Attendance Monitoring: This functionality enables the tracking of overall student attendance in specific courses, recording both attendance and absences from both the student and admin perspectives.

6. Program Management: A tool designed for the addition of new programs to the system.

7. Semester Management: A dedicated tool for incorporating new semesters within specific programs, including the assignment of faculties to courses within those programs.

8. Course Management: An essential tool for adding or updating courses seamlessly.

9. Announcement Management: A feature facilitating the dissemination of important announcements to relevant users.

11. Student Portal: A secure portal offering students access to their personal information, grades, and other academic records.

12. Online Registration: The system includes a user-friendly interface for online student registration, course enrollment, and the addition or removal of courses within specified timeframes.

2.3 User Classes and Characteristics

1. Administrator Role:
- *Full access to the Student Information System is granted to administrators.*
- *Responsibilities include managing user accounts, overseeing new admissions and instructors.*
- *Capabilities extend to configuring academic programs, and handling enrollment processes.*

2. Faculty Role:
- *They can view and update student information, including attendance and grades.*
- *They can access multiple courses they teach and handle the students.*

3. Student Role:
- *Students possess the ability to access their personal records, covering grades, attendance, schedules, and personal information.*
- *Registration for classes and tracking academic progress are integral functionalities available to students.*

2.4 Operating Environment

The architecture of the SIS system is meticulously crafted to seamlessly operate on any web-based platform, ensuring accessibility through standard web browsers with an active internet connection. This design philosophy empowers users to engage with the system on diverse operating systems, including but not limited to Windows, MacOS, and Linux. The flexibility of web-based functionality underscores the system's commitment to providing a universal and user-friendly experience across various computing environments.

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 Technologies used

1. Front-end Development:
- The front-end of the system is developed using JavaScript, CSS, and Bootstrap (CSS framework) to ensure a responsive and visually appealing user interface.

1. Back-end Development Dependencies:
- bcrypt: Employed for secure password hashing, ensuring the confidentiality of user credentials.
- bcryptjs: A library offering additional functionalities for password hashing in a secure manner.
- body-parser: Middleware essential for parsing request bodies in the Express framework, facilitating effective handling of incoming data.
- connect-flash: Middleware used for storing and displaying flash messages in Express, enhancing user communication.
- cookie-parser: A library facilitating the parsing of cookies in Express, crucial for managing user sessions.
- dotenv: Utilized for loading environment variables from a .env file, enhancing configuration management.
- ejs: A versatile template engine for generating dynamic HTML pages in Node.js, contributing to an interactive user interface.
- exceljs: A library enabling the generation and manipulation of Excel files in Node.js, enhancing data presentation capabilities.
- express: A widely-used web framework for Node.js, providing the foundational structure for building robust web applications and APIs.
- express-session: Middleware responsible for managing user sessions in Express, a critical component for user authentication and authorization.
- jsonwebtoken: Used for creating and verifying JSON Web Tokens (JWT), ensuring secure communication and authentication.
- mongoose: A MongoDB object modeling tool for Node.js that offers a schema-based solution for modeling application data, facilitating interaction with the database.
- morgan: A middleware used for logging HTTP requests in Express, aiding in debugging and monitoring.
- multer: Middleware designed for handling file uploads in Express, enhancing the system's capabilities for managing multimedia content.
- nodemailer: A library specifically employed for sending email messages in Node.js, crucial for system notifications and communication.
- nodemon: A utility that automatically restarts a Node.js application when changes are detected, streamlining the development and testing process.
- passport: Middleware dedicated to authentication in Express, essential for securing access to system resources.
- passport-local: A Passport strategy designed for authenticating users with a username and password in Express, contributing to a robust authentication process.
- passport-local-mongoose: An extension of Passport-local that simplifies user authentication by integrating with Mongoose, enhancing authentication capabilities.

- path: A core module in Node.js used for handling and transforming file paths, facilitating efficient file management.
- sessionstorage-for-nodejs: A library providing session storage functionality for Node.js applications, enhancing session management capabilities.
- validator: A library offering a set of validation functions, ensuring data integrity and security.
- xlsx: A library enabling the manipulation and generation of Excel files in Node.js, enhancing data handling capabilities.

3. Database:
- The system utilizes the Cloud platform Atlas-MongoDB to store data and query the database.

2.7 User Documentation

[1] Demo Video:

2.8 Assumptions and Dependencies

The software presupposes users have a reliable internet connection and a compatible web browser. Basic computer skills and familiarity with web-based applications are assumed. Users should possess the necessary permissions to access and perform tasks within the system. Accurate and up-to-date information about students is expected. The system assumes a maximum of 5 courses per semester.

Dependency on accessible third-party technologies and databases is vital. Availability and responsiveness of technical support may be crucial for user assistance.

3. System Features

3.1 Admin/Student/Instructor Login

3.1.1 Feature Overview and Priority
This functionality enables users to log in with valid credentials and access the system's user functions. The priority assigned to this feature is deemed as High.

3.1.2 User Interaction and System Response
I. User Interaction: On the home page, the user initiates the login process by clicking the login button based on their role.
- System Response: The system responds by displaying the login page, prompting users to input their email and password.

II. User Interaction: Upon entering their username and password, the user clicks the submit button.
- System Response: The system validates the credentials and directs the user to the dashboard if they are valid. In case of invalid credentials, the system displays an error message.

3.1.3 Functional Requirements
• REQ-1: The system should provide a registration page to add new admin, student and instructors.

• REQ-2: The system should display a success message on successful registration and an error message if no.

3.2 Course management

3.2.1 Feature Overview and Priority
This feature empowers the Admin to effectively manage courses, and its priority is classified as High.

3.2.2 User Interaction and System Response
I. User Interaction: The Admin initiates the process by clicking the course management button on the dashboard.
- System Response: The system promptly displays the course management page, presenting a list of existing courses.

II. User Interaction: Subsequently, the Admin clicks the add or delete course button as needed.
- System Response: The system reacts by presenting the add course page or update course page, prompting the Admin to input the requisite course information.

III. User Interaction: Upon entering or updating the course information, the Admin clicks the submit button.
- System Response: The system securely saves the new course information to the database and provides a success message.

3.3.3 Functional Requirements
- REQ-1: The system must facilitate the Admin in managing courses.
- REQ-2: New course information input by the Admin should be saved to the database.
- REQ-3: Following the successful saving of new course information, the system must display a success message.

3.3 New student registration

3.3.1 Feature Overview and Priority
This feature enables the admin to register new students, with a prioritization level of High.

3.3.2 User Interaction and System Response
I. User Interaction: The admin initiates the process by clicking the Registration button on the dashboard.
- System Response: The system promptly displays the registration form, prompting the admin to enter necessary information.

II. User Interaction: Upon entering students name, email, and other required details, the admin clicks the submit button.
- System Response: Subsequently, the system sends an email to the new user containing a system-generated password, accompanied by a success message.

3.3.3 Functional Requirements
- REQ-1: The system must furnish a registration page for the addition of new admins, students, and instructors.

- REQ-2: Upon successful registration, the system should display a success message; otherwise, an error message must be shown.

3.4 Program management

3.4.1 Feature Overview and Priority
This feature grants the Admin the capability to manage academic programs, with a prioritized status of High.

3.4.2 User Interaction and System Response
I. User Interaction: The Admin initiates the process by clicking the program management button on the dashboard.
- System Response: Promptly, the system displays the program management page, presenting a list of existing programs.

II. User Interaction: Subsequently, the Admin clicks the add program button.
- System Response: The system responds by presenting the add program page, prompting the admin to input the necessary program information.

III. User Interaction: Upon entering the program information, the Admin clicks the submit button.
- System Response: The system securely saves the new program information to the database and displays a success message.

3.4.3 Functional Requirements
- REQ-1: The system must facilitate Admins in adding new academic programs.
- REQ-2: New program information input by the Admin should be securely saved to the database.
- REQ-3: After the successful saving of new program information, the system should display a success/error message.

3.5 Degree management

3.4.1 Feature Overview and Priority
This feature grants the Admin the capability to manage academic degree, with a prioritized status of High.

3.4.2 User Interaction and System Response
I. User Interaction: The Admin initiates the process by clicking the program management button on the dashboard.
- System Response: Promptly, the system displays the degree management page, presenting a list of existing programs.

II. User Interaction: Subsequently, the Admin clicks the add degree button.
- System Response: The system responds by presenting the add program page, prompting the admin to input the necessary program information.

III. User Interaction: Upon entering the degree information, the Admin clicks the submit button.

- System Response: The system securely saves the new program information to the database and displays a success message.

3.4.3 Functional Requirements
- REQ-1: The system must facilitate Admins in adding new academic degree.
- REQ-2: New program information input by the Admin should be securely saved to the database.
- REQ-3: After the successful saving of new degree information, the system should display a success/error message.

3.6 Branch management

3.4.1 Feature Overview and Priority
This feature grants the Admin the capability to manage academic branch, with a prioritized status of High.

3.4.2 User Interaction and System Response
I. User Interaction: The Admin initiates the process by clicking the degree management button on the dashboard.
- System Response: Promptly, the system displays the degree management page, presenting a list of existing programs.

II. User Interaction: Subsequently, the Admin clicks the add degree button.
- System Response: The system responds by presenting the add program page, prompting the admin to input the necessary program information.

III. User Interaction: Upon entering the program information, the Admin clicks the submit button.
- System Response: The system securely saves the new degree information to the database and displays a success message. The admin can add/remove degree too.

3.4.3 Functional Requirements
- REQ-1: The system must facilitate Admins in adding new academic degree.
- REQ-2: New program information input by the Admin should be securely saved to the database.
- REQ-3: After the successful saving of new degree information, the system should display a success/error message.

3.7 Announcement

3.4.1 Feature Overview and Priority
This feature grants the Admin the capability to manage announcement, with a prioritized status of High, and students a prioritized status of medium.

3.4.2 User Interaction and System Response
I. User Interaction: The Admin initiates the process by clicking the announcement button on the dashboard.
- System Response: Promptly, the system displays the announcement page, presenting a dialog box to send announcement.

II. User Interaction: Subsequently, the Admin fills the boxes, sets date and clicks the add button.
- System Response: The system securely saves the new announcement information to the database and displays to student side.

3.4.3 Functional Requirements
- REQ-1: The system must facilitate Admins in adding announcements.
- REQ-2: New announcement input by the Admin should be securely saved to the database.
- REQ-3: After the successful saving of new announcement information, the system should display a success/error message.

3.8 Semester management

3.5.1 Feature Overview and Priority
This feature empowers the Admin to add new upcoming semesters along with their details, with a designated priority of High.

3.5.2 User Interaction and System Response
I. User Interaction: The Admin initiates the process by clicking the semester management button on the dashboard.
- System Response: Subsequently, the system displays the semester management page, providing an overview of existing semesters.

II. User Interaction: Following this, the Admin enters the semester name, selects programs, and clicks the next button.
- System Response: The system redirects the admin to a new page, where they are prompted to assign Instructors to each course in the semester of the selected program.

3.5.3 Functional Requirements
- REQ-1: The system should enable Admin and faculty to view a comprehensive list of all semesters.
- REQ-2: The system must incorporate measures to prevent overlapping semesters, ensuring accurate scheduling.

3.9 Courses Grade

3.6.1 Feature Overview and Priority
This feature enables the faculty to initiate to add all students grade card/report. Its priority is rated as High for the faculty side and Medium for the student side.

3.6.2 User Interaction and System Response
I. User Interaction: The faculty triggers the Course grade button on the dashboard.
- System Response: Consequently, the system presents a dialogue box, displaying a checkbox and upload file option. Though it faculty can add courses as per their teaching.

II. User Interaction: The faculty can add course grade as per their taught courses as per their requirement using the provided buttons and upload their excel sheets.

- System Response: The system offers functionality for adding, updating courses grades.

3.6.3 Functional Requirements
- REQ-1: The system must permit the faculty to add course grades based on course and degree criteria.

3.10 Course attendance

3.6.1 Feature Overview and Priority
This feature enables the faculty to initiate to add all student attendance report. Its priority is rated as High for the faculty side and Medium for the student side.

3.6.2 User Interaction and System Response
I. User Interaction: The faculty triggers the Course Attendance button on the dashboard.
- System Response: Consequently, the system presents a dialogue box, displaying a checkbox and upload file option. Though it faculty can add courses as per their teaching.

II. User Interaction: The faculty can add course grade as per their taught courses as per their requirement using the provided buttons and upload their excel sheets.
- System Response: The system offers functionality for adding, updating courses attendance.

3.6.3 Functional Requirements
- REQ-1: The system must permit the faculty to add course attendance based on course and degree criteria

3.11 Admin/Student/Instructor Change Password

3.1.1 Feature Overview and Priority
This functionality enables users to log in with valid credentials and access the system's user functions. The priority assigned to this feature is deemed as High.

3.1.2 User Interaction and System Response
I. User Interaction: The user initiates the login process by clicking the change password button on respective dashboard.
- System Response: The system responds by displaying the change password page, prompting users to input old password, new password and confirm password.

II. User Interaction: Upon entering their password, the user clicks the submit button.
- System Response: The system validates the credentials and show success dialog box. In case of invalid credentials, the system displays an error message.

3.1.3 Functional Requirements
• REQ-1: The system should display a success message on successful change password and an error message if no.

4. External Interface Requirements

4.1 User Interfaces

• Admin Interface: The Admin Interface offers access to administrative tasks, including the addition of new students and faculty members, course and semester management, and oversight of programs, branches, and degrees. The interface features a sidebar with standard buttons for common functions like Announcements, new course registration openings, Logout, etc. The screen layout is strategically organized to ensure convenient access to all functions with logically arranged menu options.

• Faculty Interface: The Faculty Interface provides access to course-related activities, such as entering grades and managing student attendance. The interface incorporates a dashboard with quick links to essential functions like Profile and Attendance. The screen layout prioritizes simplicity and cleanliness, facilitating easy navigation between different sections of the interface. Standard buttons, including Logout, are consistently available on every screen.

• Student Interface: The Student Interface grants access to personal information, grades, fee receipts, and course registration. The interface features a dashboard with quick links to key functions such as profile, semester registration, announcements, grades, and attendance. The screen layout is designed for clarity and simplicity, ensuring straightforward navigation across different sections of the interface. Standard buttons and functions, such as Help and Logout, are uniformly present on each screen.

Error message display standards across all interfaces adhere to clear and concise messages detailing the error and providing resolution suggestions. Additionally, alerts are implemented for confirmation, and success messages are displayed upon the successful completion of tasks.

4.2 Hardware Interfaces

The system can be run on any operating system supporting a web-based platform like Windows, MacOS or Linux with internet connectivity.

4.3 Software Interfaces

The Student Information System necessitates an operating system compatible with the development software, meeting performance and security criteria set by the institution. For accessibility, the system must align with widely used web browsers like Google Chrome, Mozilla Firefox, and Microsoft Edge. Compatibility with these browsers ensures seamless access and optimal performance for users.

4.4 Communications Interfaces

The system necessitates a web browser for accessing the Student Information System (SIS), supporting a user-friendly interface across major browsers like Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

For communication, the system will utilize the SMTP protocol to send email notifications, covering events such as registration confirmation, grade updates, and fee payment reminders.

To host the application and database, a network server is required, employing the TCP/IP protocol for client-server communication.

Emphasizing security, all communication between the client and server must be secure and encrypted. The system will implement the SSL/TLS protocol to ensure secure communication.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system should be able to manage the traffic of at least 800 users simultaneously. The system should respond quickly to user requests and have fast processing times, and minimum delay, even during peak usage times.

5.2 Safety Requirements

5.2.1 Data Privacy: The software is mandated to uphold stringent data privacy measures, guaranteeing the confidentiality of all student information. Access to such data is restricted solely to authorized personnel, reinforcing the security and privacy of sensitive student records.

5.2.2 Disaster Recovery: In anticipation of unforeseen events, the software is obligated to maintain a robust disaster recovery plan. This strategic framework ensures the safeguarding of student information, allowing for swift recovery in the event of a disaster or system failure. The software's commitment to comprehensive disaster recovery measures underscores its dedication to preserving the integrity and availability of crucial student data.

5.2.3 Backup and Restore: To mitigate the impact of potential data loss, the software must integrate a reliable backup and restore system. This system is pivotal in securing student information, providing a failsafe mechanism for data recovery. By implementing effective backup and restore protocols, the software ensures the resilience of student data against unforeseen circumstances, contributing to the overall reliability and stability of the system.

5.3 Security Requirements

The system should be secured with different levels of security for different users.

5.4 Software Quality Attributes

5.4.1 Maintainability: Emphasizing long-term sustainability, the software is designed for ease of maintenance and updates. Its codebase is characterized by clarity and comprehensive documentation, enabling swift issue identification and resolution. This commitment to maintainability ensures the software remains adaptable and responsive to evolving needs.

5.4.2 Reliability: With an unwavering commitment to reliability, the software strives for uninterrupted availability. Operating with minimal downtime or disruptions, it aims to

provide a seamless user experience. This dedication to reliability ensures consistent access to the software, fostering a dependable platform for continuous use.

5.4.3 Portability: Acknowledging diverse technology landscapes, the software is engineered for portability across various platforms and operating systems. This adaptability minimizes the need for significant modifications, ensuring the software's versatility and compatibility across different technological environments.

5.4.4 Usability: Prioritizing user experience, the software is crafted to be user-friendly and easily navigable. Clear and concise instructions, coupled with informative feedback, contribute to an intuitive interface. This commitment to usability enhances the accessibility and effectiveness of the software, promoting a positive and efficient user interaction.

5.5 Business Rules

5.5.1 Access Control: Upholding a principle of strict information control, access to student data is governed by a need-to-know basis. This meticulous approach to access authorization ensures that sensitive information is only disclosed to individuals with a legitimate need for such knowledge, bolstering data security and confidentiality.

5.5.2 Data Storage: Ensuring the utmost security, student information is stored with a robust commitment to safeguarding its integrity. The software implements secure data storage practices, emphasizing the importance of preserving the confidentiality and privacy of student records.

5.5.3 Student Privacy: A paramount concern is the protection of student privacy, necessitating the strict confidentiality of sensitive information like social security numbers, medical records, and disciplinary records. This unwavering commitment to student privacy underscores the software's dedication to ethical and responsible data handling.

5.5.4 Enrollment: The enrollment process demands that students fulfill specific criteria and furnish the requisite information to enroll in the school or program. This systematic approach ensures that enrollment is both comprehensive and aligned with the established standards.

5.5.5 Attendance: Precision in record-keeping is paramount, and the software maintains accurate attendance records for all students. This meticulous tracking system contributes to an effective monitoring process, aiding educational institutions in ensuring student engagement and compliance.

5.5.6 Grading: An imperative aspect of academic management is the accurate and equitable calculation of grades. The software is designed to uphold the standards of fairness and accuracy in grading, ensuring that students receive evaluations that genuinely reflect their academic performance.

6. User Stories

6.1.1 Login:

- Student Perspective:
- I want to log in with my student ID and password to access academic records and important information.
- Admin Perspective:
- I want to log in using my credentials to manage student and faculty accounts, security settings, and other tasks.
- Faculty Perspective:
- I want to log in using my credentials to access and update student records, including grades and attendance.

Confirmation:
- Success: All actors (Student, admin, and faculty) can log in and are directed to the home page.
- Failure: Display messages for various issues like incorrect credentials, unrecognized username, service unavailability, or expired accounts.

6.1.2 View Student Details:

- Student Perspective:
- I want to access personal information, grade cards, registered courses, and attendance.
- I want a digital identity card for emergency situations.

Confirmation:
- Success: Registered and verified students can see their personal and academic details.
- Failure: Admin entering wrong data during registration causes issues.

6.1.3 View Faculty Details:

- Faculty Perspective:
- I want to access personal information.
- I want a digital identity card for emergency situations.

Confirmation:
- Success: Registered and verified faculty can see their personal and academic details.
- Failure: Admin entering wrong data during registration causes issues.

6.1.4 Enrolled Course:

- Student Perspective:
- I want to enroll in courses with clear information about the course description.

Confirmation:
- Success: All students can enroll successfully.
- Failure: Display messages for enrollment denial or system errors.

6.1.5 Announcement:

- Student Perspective:
- I want to see announcements posted by admin.
- Admin Perspective:
- I want to post announcements visible to students until a specified date.

Confirmation:
- Success: Students can view admin announcements within the designated timeframe.

6.1.6 View Attendance:

- Student Perspective:
- I want to view my attendance record.

Confirmation:
- Success: All registered students can view their attendance.
- Failure: Incorrect details entered by faculty result in attendance unavailability.

6.1.7 Enter Attendance:

- Faculty Perspective:
- I want to enter student attendance data accurately.

Confirmation:
- Success: Successfully entering attendance displays the message "Attendance entered successfully."
- Failure: Display message "Unable to enter attendance, please try again."

6.1.8 Add Semester:

- Admin Perspective:
- I want to add a new semester with related courses.

Confirmation:
- Success: Confirmation message for adding a new semester.
- Failure: Display messages for duplicate semester names or general errors.

6.1.9 Add Programme:

- Admin Perspective:
- I want to add a new programme with related courses.

Confirmation:
- Success: Confirmation message for adding a new programme.
- Failure: Display messages for duplicate programme names or general errors.

6.1.1 Add Branch:

- Admin Perspective:
- I want to add a new branch with related courses.

Confirmation:
- Success: Confirmation message for adding a new branch.
- Failure: Display messages for duplicate branch names or general errors.

6.1.2 Add Degree:

- Admin Perspective:
- I want to add a new degree with related courses.

Confirmation:
- Success: Confirmation message for adding a new degree.
- Failure: Display messages for duplicate degree names or general errors.

6.1.3 Add Courses:

- Admin Perspective:
- I want to add a new course.

Confirmation:
- Success: Confirmation message for adding a new course.
- Failure: Display messages for duplicate course names or general errors.

6.1.4 Publish Grade:

- Faculty Perspective:
- I want to publish grades at the end of each semester.

Confirmation:
- Success: Published grades are available for students.
- Failure: Errors for incorrect search keys.

6.1.5 View Grade:

- Student Perspective:
- I want to view my grade card.

Confirmation:
- Success: All registered students can view their grades.

- Failure: Incorrect details entered by faculty result in grade unavailability.

6.1.6 Register Student Details:

- Admin Perspective:
- I want to register new students.

Confirmation:
- Success: Display the message "New student registered successfully."
- Failure: Display message "Student registration failed due to invalid details."

6.1.7 Change Password:

- Student Perspective:
- I want to change password in with my old password and new password and save it in database so that I can login.
- Admin Perspective:
- I want to change password in with my old password and new password and save it in database so that I can login.
- Faculty Perspective:
- I want to change password in with my old password and new password and save it in database so that I can login.

Confirmation:
- Success: All actors (Student, admin, and faculty) can log in with new password directed to the home page.
- Failure: Display messages for various issues like incorrect old password or not registered new password to database.

6.1.8 Performance:

- Admin Perspective:
- I want the system to handle 100 users simultaneously.

6.1.9 Scalability:

- User Perspective:
- I want the system to scale with increasing data and users.

6.1.10 Security:

- Admin Perspective:
- I want a high level of security for student information to prevent unauthorized access.

6.1.11 User-friendly:

- User Perspective:
- I want an interactive system for easy tracking of activities.

6.1.12 Reliability:

- User Perspective:
- I want a highly reliable system to trust my data's integrity.

6.1.13 Maintainability:

- Admin Perspective:
- I want the system to be easily maintainable for quick restoration after failures.

6.1.14 Portability:

- User Perspective:
- I want a portable system for access from any location without significant downtime.

6.1.15 Availability:

- User Perspective:
- I want a highly available system with no significant downtime.

6.1.16 Compatibility:

- User Perspective:
- I want a system compatible with all devices.

6.1.17 Usability:

- User Perspective:
- I want an easy-to-use system for quick navigation and information retrieval.

6.1.18 Database Consistency:

- User Perspective:
- I want a consistent system to avoid data ambiguity.

.