

# Software Requirements Specification

for

## "Department Management System"

Prepared by:

Samia Alam (348)  
Abdus Salam (2148)  
Md Rakibul Haque (373)  
Niaz Rahman (401)  
Sakul Hasan(389)  
Taufiq Labib(383)

Jahangirnagar University, Savar  
*Software Engineering Lab*

14 September, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Intended Audience: . . . . .	2
1.3	Intended User . . . . .	2
1.4	Product Scope . . . . .	3
1.5	Risk Definition . . . . .	3
<b>2</b>	<b>Overall Description</b>	<b>5</b>
2.1	User Classes and Characteristics . . . . .	5
2.1.1	Class Representative (CR) . . . . .	5
2.1.2	Batch Members . . . . .	5
2.1.3	System Authority . . . . .	6
2.1.4	System (Automated Processes) . . . . .	6
2.2	System Requirements . . . . .	6
2.2.1	Hardware Platform . . . . .	6
2.2.2	Operating System and Versions . . . . .	6
2.2.3	Software Components and Applications . . . . .	6
2.2.4	Database Compatibility . . . . .	7
2.2.5	Interoperability . . . . .	7
2.2.6	Interoperability . . . . .	7
2.2.7	Security Considerations . . . . .	7
2.3	Constraints . . . . .	7
2.3.1	Technical Constraints . . . . .	7
2.3.2	Time Constraints . . . . .	7
2.3.3	Budget Constraints . . . . .	7
2.3.4	Regulatory and Compliance Constraints . . . . .	7
2.3.5	Resource Constraints . . . . .	8
2.4	Assumption of the Project . . . . .	8
2.4.1	User Roles and Access Control: . . . . .	8

2.4.2	Stable Internet Connection: . . . . .	8
2.4.3	Technical Infrastructure: . . . . .	8
2.4.4	Data Accuracy and Entry: . . . . .	8
2.4.5	User Training and System Usability: . . . . .	8
2.4.6	Regular System Maintenance: . . . . .	9
2.4.7	System Customization: . . . . .	9
<b>3</b>	<b>Requirements</b>	<b>10</b>
3.1	Functional Requirements: . . . . .	10
3.1.1	Register New Batch . . . . .	10
3.1.2	Validate A Batch for New Registration . . . . .	11
3.1.3	Manage Batch Information . . . . .	11
3.1.4	Edit User Profile Information . . . . .	12
3.1.5	Login A Registered Batch . . . . .	13
3.1.6	Add Semester . . . . .	14
3.1.7	Add Course . . . . .	15
3.1.8	View Course Details . . . . .	16
3.1.9	Maintain Department Notice Board . . . . .	17
3.1.10	Send Notices for Batch Updates . . . . .	18
3.1.11	Send Department Notice . . . . .	18
3.1.12	Handle Job and Internship Information . . . . .	19
3.2	Non-Functional Requirements (NFRs) . . . . .	20
3.2.1	Performance . . . . .	20
3.2.2	Scalability: . . . . .	21
3.2.3	Usability: . . . . .	21
3.2.4	Security: . . . . .	21
3.2.5	Reliability: . . . . .	21
3.2.6	Maintainability: . . . . .	21
3.2.7	Data Integrity: . . . . .	21
3.2.8	Availability: . . . . .	21

# Chapter 1

## Introduction

### 1.1 Purpose

The purpose of the **Batch Management System** project is to streamline batch-related operations and communication within the Department of Computer Science and Engineering at Jahangirnagar University. It aims to facilitate efficient management of student batches, including the distribution of notices, resource sharing, and communication between Class Representatives (CRs), faculty, and students. The system will ensure timely dissemination of important information, improving overall coordination and collaboration within the department. Additionally, it enhances user management and provides tools for department-specific resource sharing. The system is designed to be user-friendly and secure, addressing the department's operational needs.

### 1.2 Intended Audience:

The SRS document is intended for the following audiences:

- Project Managers
- System Architects and Developers
- Testers and Quality Assurance (QA) Teams
- Stakeholders and Department Heads
- End Users (Students, Faculty, Admins)
- Maintenance and Support Teams

### 1.3 Intended User

The SRS will be used for the following purposes:

- **Project Managers:**  
To oversee the project's scope, schedule, and resource allocation. They will use the SRS to plan and manage project timelines, allocate resources, and monitor progress against the requirements outlined in the document.

- **System Architects and Developers:**

To design and implement the system based on the detailed requirements specified in the SRS. They will refer to the SRS to guide system design and development, ensuring that all functionalities and constraints are met.

- **Testers and Quality Assurance (QA) Teams:**

To develop test cases and ensure the system meets quality standards. The SRS will be used to create and execute test plans, validating that the system performs according to the specified requirements.

- **Stakeholders and Department Heads:**

To verify that the system aligns with departmental goals and expectations. They will review and approve the requirements to ensure that the system supports the needs of the department and its users.

- **End Users (Students, Faculty, Admins):**

To understand how the system will meet their needs and expectations. They will interact with the system and provide feedback on its functionality and usability, ensuring it meets their requirements.

- **Maintenance and Support Teams:**

To provide ongoing support and system maintenance after deployment. The SRS will serve as a reference for maintaining and updating the system to ensure it continues to meet the defined requirements.

## 1.4 Product Scope

The **Batch Management System** will provide functionalities to manage batch-related operations and communication within the Department of Computer Science and Engineering at Jahangirnagar University.

Key features include:

- **User Management:** Account creation, role assignment, and profile management.
- **Batch Management:** Creation and updates of batch data, schedules, and communication.
- **Notice System:** Ability for CRs and Admins to send and manage notices, with a centralized notice board.
- **Sharing Resources:** Tools for sharing resources, books, notes, and reports.

Exclusions:

- The system will not include financial management, advanced analytics beyond basic reporting, or integration with external systems unless specified in future phases.

## 1.5 Risk Definition

Potential risks associated with the **Batch Management System** include:

- **Scope Creep:**

This risk can result in project delays, increased costs, and deviation from the original objectives, as additional features or changes are introduced without proper evaluation and adjustments.

- **Technical Challenges:**

Technical challenges refer to potential difficulties arising from the technology used in the system. These challenges can include software development issues, integration problems with existing systems, or performance limitations. Such challenges might emerge from unforeseen bugs, compatibility issues, or limitations inherent in the technology stack.

- **User Acceptance:**

User acceptance risk pertains to the likelihood that end users may not embrace or effectively use the system. This risk can arise if the system fails to meet user needs, lacks essential features, or is not intuitive and user-friendly. Poor user adoption can undermine the system's effectiveness and lead to a lack of engagement.

- **Data Security and Privacy:**

Data security and privacy risks involve the potential for unauthorized access, data breaches, or misuse of sensitive information. These risks can jeopardize the confidentiality, integrity, and availability of the data managed by the system, potentially leading to legal, financial, or reputational repercussions.

- **Resource Availability:**

Resource availability risk concerns the potential for insufficient resources—such as personnel, budget, or technology—to complete the project as planned. This risk can emerge from unexpected changes in team composition, budget constraints, or other factors affecting the availability of necessary resources.

- **Timeline and Deadlines:**

The risk associated with timelines and deadlines pertains to the potential failure to meet project deadlines. This risk can result from delays in development, unexpected issues, or changes in project scope. Such delays can impact the overall project schedule and timely delivery of the system.

## Chapter 2

# Overall Description

### 2.1 User Classes and Characteristics

#### 2.1.1 Class Representative (CR)

- **Batch Management:** Registers new batches, manages batch information, and ensures accuracy and completeness of batch details.
- **Course Management:** Adds, updates, or deletes course information, ensuring that all course details are accurate and up-to-date.
- **Semester Management:** Manages semester scheduling by adding, updating, or deleting semester details aligned with the academic calendar.
- **Notice Management:** Sends notices for batch updates such as exams and class leaves to keep batch members informed.
- **User Profile:** Can edit their profile information to maintain current and accurate personal details.
- **Dashboard Access:** Logs in to the registered batch and accesses a dashboard to manage batch activities and information.
- **Job and Internship Handling:** Coordinates and manages job and internship opportunities for batch members.
- **Authorization:** Must be authorized to perform these actions, with access controlled by role-based permissions.

#### 2.1.2 Batch Members

- **Login Access:** Logs into their registered batch to access batch-specific information and updates.
- **Course Details:** Views course details and semester schedules relevant to their batch.
- **Notices:** Receives and interacts with notices sent by the CR or Admin regarding batch updates.
- **Profile Management:** Can edit their own profile information to ensure accuracy.
- **Job and Internship:** Engages with job and internship opportunities provided by the batch management system.

### 2.1.3 System Authority

- **Batch Creation Permissions:** Grants or verifies permissions for users to create new batches. Ensures that only authorized users (CRs and Admins) can initiate the batch creation process.
- **Device Login Permissions:** Manages and verifies Permission Keys for users attempting to log in from new devices. Ensures secure and authorized access by validating these keys.
- **Validation and Token Generation:** Handles automated processes for generating real-time tokens during batch registration and ensuring the security of these tokens.
- **Error Handling:** Manages and logs errors related to unauthorized access attempts, validation failures, and system malfunctions.
- **Security Measures:** Implements security protocols to protect the system from unauthorized access and ensure that only validated and authorized actions are permitted.
- **Audit and Monitoring:** Monitors system activities related to batch creation and new device logins, maintaining logs for auditing and compliance purposes.

### 2.1.4 System (Automated Processes)

- **Validation Processes:** Handles automated validation of batch registration requests, email verifications, and token generation.
- **Error Handling:** Logs errors and provides relevant error messages to users in case of issues during batch registration, course management, or other functionalities.
- **Data Storage:** Manages storage of batch, course, semester, and notice details in the database.

## 2.2 System Requirements

### 2.2.1 Hardware Platform

- **Desktops:** Intel Core i5 processor or equivalent, 8GB RAM, 256GB SSD or higher.
- **Laptops:** Intel Core i3 processor or equivalent, 4GB RAM, 128GB SSD or higher.
- **Mobile Devices:** IOS devices (iPhone 6S and above), Android devices (running Android 8.0 and above).

### 2.2.2 Operating System and Versions

- **Windows:** Windows 10 (64-bit), Windows 11 (64-bit).
- **MacOS:** macOS Catalina (10.15) and above.
- **Linux:** Ubuntu 20.04 LTS and above, Fedora 33 and above.
- **Mobile Devices:** IOS 13 and above, Android 9.0 and above.

### 2.2.3 Software Components and Applications

- **Google Chrome:** Latest Stable Version.
- **Mozilla Firefox:** Latest Stable Version.
- **Microsoft Edge:** Latest Stable Version.
- **Safari:** Latest Stable Version.



### 2.2.4 Database Compatibility

- **Relational Database:** MySQL (version 8.0), PostgreSQL (version 13).
- **NoSQL Database:** MongoDB (version 4.4).

### 2.2.5 Interoperability

- The system utilizes RESTful APIs and adheres to JSON data interchange standards, ensuring compatibility with third-party services and applications.

### 2.2.6 Interoperability

- A stable internet connection with a minimum bandwidth of 5 Mbps.
- Support for both wired (Ethernet) and wireless (Wi-Fi) network connections.

### 2.2.7 Security Considerations

- **Encryption:** TLS encryption (SSL certificates).
- **Compatibility:** Industry-standard firewalls and security software.

## 2.3 Constraints

### 2.3.1 Technical Constraints

- Selection of appropriate frameworks and technologies for:
  1. Secure payment processing
  2. User authentication
- Consideration of scalability as the user base grows and the system needs to handle increased demand.

### 2.3.2 Time Constraints

- Establishing development phases with clear milestones for feature implementation and testing.
- Meeting the launch deadline based on market analysis and business objectives.

### 2.3.3 Budget Constraints

- Efficient allocation of resources for:
  1. Software development
  2. Hosting infrastructure
  3. Potential marketing and promotional efforts

### 2.3.4 Regulatory and Compliance Constraints

- Ensuring compliance with data protection laws (e.g., GDPR, HIPAA) concerning user data and payment details.
- Considering legal aspects of handling batch-related agreements, responsibilities, and liabilities.

### 2.3.5 Resource Constraints

- Availability of skilled development resources, including expertise in the necessary technologies and frameworks required for project execution.

## 2.4 Assumption of the Project

### 2.4.1 User Roles and Access Control:

- It is assumed that users (students, admins, and Class Representatives (CRs)) will have distinct roles and permissions defined through role-based access control. Each role will have specific privileges, such as students being able to view course details and admins being responsible for managing batches.
- Only authorized users (e.g., Admins and CRs) are allowed to create, update, and delete sensitive information like batches or notices.

### 2.4.2 Stable Internet Connection:

- The system is designed to function in an environment with a stable internet connection. It is assumed that users will access the system through web browsers or mobile devices with consistent internet availability.
- Performance and response times are based on the assumption of good network conditions. If users experience poor internet connectivity, response times may be slower, which is outside the system's control.

### 2.4.3 Technical Infrastructure:

- It is assumed that the institution will provide sufficient hardware resources (servers, databases, backup systems) to support the smooth functioning of the system. The system will be hosted on reliable infrastructure with enough capacity to handle the expected load.
- The system assumes that the server environment and database will be regularly maintained, with backups performed to prevent data loss in case of system failures.

### 2.4.4 Data Accuracy and Entry:

- It is assumed that data input by admins and CRs, such as course details, batch information, and notices, will be accurate and complete. The system will perform basic validation checks, but the quality of data relies on the users entering it correctly.
- Course and batch information will be regularly updated by the administrators to ensure the most current data is available to users.

### 2.4.5 User Training and System Usability:

- It is assumed that all users (students, admins, and CRs) will have received sufficient training or guidance on how to use the system effectively.
- The system will have a user-friendly interface, but users are expected to have basic computer literacy, such as navigating web pages and filling out forms.

### 2.4.6 Regular System Maintenance:

- The system assumes that there will be routine maintenance (e.g., software updates, bug fixes, server checks) to ensure its continuous operation.
- Scheduled maintenance will be planned during non-peak hours to minimize disruption for users. It is assumed that the institution will allocate resources for ongoing technical support.

### 2.4.7 System Customization:

- It is assumed that the system will require minimal customization for the institution it is being deployed in. Any additional features or custom workflows will be addressed during the development phase, with minimal changes required after deployment.

## Chapter 3

# Requirements

### 3.1 Functional Requirements:

#### 3.1.1 Register New Batch

**As a Class Representative (CR),**

I want to create, read, update, or delete (CRUD) batch information,  
So that I can manage batch details accurately for smooth academic operations.

#### Successful Attempt:

- The system verifies that the user is authorized (CR) using role-based access control (RBAC).
- The system validates the required inputs (Batch Email, Batch Name, Batch Session, Batch Profile Image, Batch Cover Image, Number of Students, Password).
- The batch details (Batch Email, Batch Name, Batch Session, Batch Profile Image, Batch Cover Image, Number of Students, Password) are stored in the *Batches* table.
- The system maps the batch to the relevant courses and faculty members using appropriate identifiers and foreign keys.
- A confirmation response is sent back to the user (e.g., "Batch added/updated/deleted successfully").
- The batch is visible in the batch listing section, and relevant faculty, students, and administrators can view it.

#### Final Outcome:

- The batch details are stored and visible to relevant users.
- CRs, faculty, and administrators can view the updated batch listing.
- Students and other users can access updated batch information, ensuring seamless academic operations.

#### Unsuccessful Attempt:

- If the user is not authorized (not a CR), the system returns an error response (e.g., "You are not authorized to manage batch information").

- If required fields (Batch Email, Batch Name, Batch Session, Batch Profile Image, Batch Cover Image, Number of Students, Password) are missing or invalid, the system returns a validation error message (e.g., "All batch details are required").
- If the database operation fails (due to connectivity issues or internal errors), the system logs the error for further investigation. The user receives a failure message (e.g., "Failed to create/update the batch. Please try again later"). No batch is created or updated.
- Any unexpected errors during the process are caught, and the user receives a generic error message (e.g., "An error occurred. Please try again later"). The error is logged for debugging purposes.

### 3.1.2 Validate A Batch for New Registration

**As a Admin,**

I want to validate, approve, or reject new user registrations,

So that I can ensure only authorized users gain access to the system for secure and controlled operations.

#### Successful Attempt:

- **Real-time Token:** A unique, time-limited token that grants the user access to create a new batch.
- **Confirmation Message:** A message informing the user that the validation was successful and they can proceed with batch creation.

#### Final Outcome:

- A user can successfully request access to create a new batch by providing their email address.
- The system validates the email address and determines if the user is authorized.
- If the user is authorized, a real-time token is generated and provided to them.
- If the user is not authorized or there is an error in the validation process, an appropriate error message is displayed.
- The generated token is used to authenticate the user during the batch creation process.
- The token expires after a specified time period to enhance security.

#### Unsuccessful Attempt:

- **Error Message:** A message indicating the reason for the validation failure, such as:
  - **Unauthorized User:** The user does not have the necessary permissions to create a new batch.
  - **Invalid Email Address:** The provided email address is not valid or does not exist.
  - **System Error:** A technical error occurred during the validation process.

### 3.1.3 Manage Batch Information

**As an Admin,**

I want to create, update, or delete batch information, So that student groups are properly organized by academic year, course, and department, ensuring effective batch management.

**Successful Attempt:**

1. The system verifies that the user is authorized (Admin) using role-based access control.
2. The system validates the required inputs (Batch Name, Department, Academic Year, Start Date, End Date).
3. The batch details (Batch Name, Department, Academic Year, Start Date, End Date) are stored in the Batches table.
4. If creating a new batch, the system checks for duplicate entries (e.g., batch name already exists in the department for the academic year).
5. Upon successful creation, update, or deletion, the batch list is updated, reflecting the change in the system.
6. A confirmation response is sent back to the admin (Batch successfully created/updated/deleted).

**Final Outcome:**

- The batch is successfully created, updated, or deleted in the system.
- The batch information is visible and can be managed by other relevant system features (e.g., student assignment, course allocation).
- The admin receives confirmation, ensuring smooth batch management without errors.

**Unsuccessful Attempt:**

1. If the user is not authorized (not an Admin), the system returns an error response (You are not authorized to manage batches).
2. If required fields (Batch Name, Department, Academic Year, Start Date, End Date) are missing or invalid, the system returns a validation error message (All batch details are required).
3. If the batch name already exists for the department and academic year, the system returns a validation error (Batch name already exists for this department and academic year).
4. If the database operation fails (due to connectivity issues or internal errors), the system logs the error for further investigation. The user receives a failure message (Failed to manage the batch. Please try again later). No batch is created, updated, or deleted.
5. Any unexpected errors during the process are caught, and the user receives a generic error message (An error occurred. Please try again later). The error is logged for debugging purposes.

**3.1.4 Edit User Profile Information**

**As a** Registered User,

**I want** to update my profile information (such as name, contact details, and password),  
**so that** my account remains accurate and up to date.

**Successful Attempt:**

- The system verifies that the user is logged in and authorized to make changes to their own profile using the session.
- The system validates the required inputs (such as name, contact details, email address, and password) for accuracy and format.
- The updated profile details (name, email, contact number, etc.) are stored in the Users table.
- A confirmation response is sent back to the user (e.g., "Profile updated successfully").
- The system ensures that email and password fields, if updated, follow security standards (e.g., password strength, email uniqueness).

**Final Outcome:**

- The user's profile is updated, and the changes are reflected immediately in the user's account.
- Any critical information changes (e.g., email or password) trigger notifications to the user for security purposes (e.g., "Your email was updated").
- The system maintains the integrity and accuracy of user data, and the updated information is available in future interactions (such as logging in or receiving notifications).

**Unsuccessful Attempt:**

- **If the user is not authenticated (not logged in):**
  - The system prompts the user to log in (e.g., "Please log in to edit your profile").
- **If required fields (e.g., name, email, password) are missing or invalid:**
  - The system returns a validation error message (e.g., "Name, email, and password are required").
  - The system ensures the email is unique and not already associated with another account.
  - The system validates the new password strength if the password is updated (e.g., "Password must contain at least 8 characters, one uppercase letter, and one number").
- **If the database operation (updating the profile) fails due to connectivity or internal error:**
  - The system logs the error for further investigation.
  - The user receives a failure message (e.g., "Failed to update profile. Please try again later").
  - No changes are applied to the profile.
- **Any unexpected errors during the process are caught:**
  - The user receives a generic error message (e.g., "An error occurred. Please try again later").
  - The error is logged for further debugging.

### 3.1.5 Login A Registered Batch

**Successful Attempt:**

- **Dashboard:** The user will be granted access to the batch dashboard, where they can manage their batch's activities and information.

### Unsuccessful Attempt:

- **Error Message:** The user will receive an error message indicating the reason for the unsuccessful login, such as:
  - **Invalid Credentials:** The provided email address or password does not match the registered information.
  - **Missing Permission Key:** A Permission Key is required for login on a different device, and the user did not provide one.

### Final Outcome:

- Users can log in with an email and password on the same device.
- A Permission Key is required to log in on a different device.
- Once logged in, users can access the dashboard to manage their batch.
- The system must handle secure and authorized access to the login process, especially for new devices.
- The system must verify the validity of the Permission Key before allowing login.
- The system must prevent unauthorized access by implementing appropriate security measures.

#### 3.1.6 Add Semester

**As a Class Representative (CR)**, I want to create, read, update, or delete (CRUD) semester details, so that semester scheduling is aligned with the academic calendar and ensures effective planning and organization.

### Successful Attempt:

- **User Authorization:** The system verifies that the user is authorized (CR) using role-based access control (RBAC).
- **Input Validation:** The system validates the required inputs: (Semester No, Start Date, End Date, PL Off Week, Tutorial Weeks, Final Semester Date).
- **Data Storage:** The semester details (Semester No, Start Date, End Date, PL Off Week, Tutorial Weeks, Final Semester Date) are stored in the Semesters table.
- **Mapping Semesters to Users:** The system maps the semester information to the relevant academic schedules and notifications using appropriate identifiers and foreign keys.
- **Confirmation Response:** A confirmation response is sent back to the user (e.g., "Semester added/updated/deleted successfully").
- **Semester List Visibility:** The semester is visible in the semester listing section, and relevant users can view the updated semester details.

### Final Outcome:

- The semester details are accurately stored and visible to the relevant users.
- CRs and faculty members can view the updated semester listings.
- Students and other users have access to the updated semester schedule, ensuring effective academic planning.



### Unsuccessful Attempt:

- **Unauthorized User:** If the user is not authorized (not a CR), the system returns an error response (e.g., "You are not authorized to manage semester information").
- **Missing or Invalid Inputs:** If required fields (Semester No, Start Date, End Date, PL Off Week, Tutorial Weeks, Final Semester Date) are missing or invalid, the system returns a validation error message (e.g., "All semester details are required").
- **Database Error:** If the database operation fails (e.g., due to connectivity issues or internal errors), the system logs the error for further investigation. The user receives a failure message (e.g., "Failed to add/update/delete the semester. Please try again later"). No semester is created or updated.
- **Unexpected Errors:** Any unexpected errors during the process are caught, and the user receives a generic error message (e.g., "An error occurred. Please try again later"). The error is logged for debugging purposes.

### 3.1.7 Add Course

**As a Class Representative (CR),** I want to create, read, update, or delete (CRUD) course information, so that all course details are accurate, ensuring effective academic planning and tutorial scheduling.

### Successful Attempt:

- **User Authorization:** The system verifies that the user is authorized (CR) using role-based access control (RBAC).
- **Input Validation:** The system validates the required inputs (Course Code, Course Name, Course Hours, Course Credit, Prerequisites, Course Teacher).
- **Data Storage:** The course details (Course Code, Course Name, Course Hours, Course Credit, Prerequisites, Course Teacher) are stored in the Courses table.
- **Mapping Courses to Users:** The system maps the courses to the relevant faculty and students using appropriate identifiers and foreign keys.
- **Confirmation Response:** A confirmation response is sent back to the user (e.g., "Course added/updated/deleted successfully").
- **Course Visibility:** The course is visible in the course listing section, and relevant faculty and students can view it.

### Final Outcome:

- The course details are stored and visible to the relevant users.
- CRs and faculty members can view the updated course listing.
- Students and other users can access updated course information, ensuring seamless academic operations.

### Unsuccessful Attempt:

- **Unauthorized User:** If the user is not authorized (not a CR), the system returns an error response (e.g., "You are not authorized to manage course information").

- **Missing or Invalid Inputs:** If required fields (Course Code, Course Name, Course Hours, Course Credit, Prerequisites, Course Teacher) are missing or invalid, the system returns a validation error message (e.g., "All course details are required").
- **Database Error:** If the database operation fails (e.g., due to connectivity issues or internal errors), the system logs the error for further investigation. The user receives a failure message (e.g., "Failed to add/update/delete the course. Please try again later"). No course is created or updated.
- **Unexpected Errors:** Any unexpected errors during the process are caught, and the user receives a generic error message (e.g., "An error occurred. Please try again later"). The error is logged for debugging purposes.

### 3.1.8 View Course Details

#### As a Student,

I want to view course details, so that I can access important information like course description, credits, instructor details, and schedule, ensuring that I have all necessary information to plan my academic activities.

#### Successful Attempt:

1. **User Authorization:** The system verifies that the user is authorized (Student) through role-based access control (RBAC).
2. **Course Search:** The student searches for a course by entering a course code or selecting from the list of available courses.
3. **Data Retrieval:** The system retrieves the course details (Course Title, Course Description, Credits, Instructor Name, Schedule) from the database.
4. **Course Display:** The course details are displayed in a structured format, including all relevant information for the selected course.
5. **Additional Information:** The system checks if the course is available for the current academic term and highlights important information such as registration deadlines or prerequisites.
6. **Confirmation Response:** A confirmation response is sent back to the student (e.g., "Course details loaded successfully").

#### Final Outcome:

- The course details are successfully retrieved and displayed to the student.
- The student can view all necessary information regarding the course, helping in making academic decisions or planning schedules.
- The interface is responsive and provides a seamless user experience.

#### Unsuccessful Attempt:

1. **Unauthorized User:** If the user is not authorized (not a student), the system returns an error response (e.g., "You are not authorized to view course details").
2. **Invalid Course Code:** If the student enters an invalid course code or no matching course is found, the system returns a validation error message (e.g., "Course not found. Please check the course code or try another course").

3. **Database Error:** If the database operation fails (e.g., due to connectivity issues or internal errors), the system logs the error for further investigation. The user receives a failure message (e.g., "Failed to load course details. Please try again later"). No course details are displayed.
4. **Unexpected Errors:** Any unexpected errors during the process are caught, and the user receives a generic error message (e.g., "An error occurred. Please try again later"). The error is logged for debugging purposes.

### 3.1.9 Maintain Department Notice Board

#### As a Class Representative (CR),

I want to create, update, or delete department notices, so that important academic information and announcements are effectively communicated to students and faculty members.

#### Successful Attempt:

1. **User Authorization:** The system verifies that the user is authorized (CR) using role-based access control (RBAC).
2. **Input Validation:** The system validates the required inputs: (Notice Title, Notice Description, Date, Expiry Date, Audience Type - Students).
3. **Data Storage:** The notice details (Notice Title, Notice Description, Date, Expiry Date, Audience Type) are stored in the `Notices` table.
4. **Notifications:** The system notifies the relevant faculty and students about the newly added notice via email or in-app notification.
5. **Notice Display:** The notice is displayed on the department notice board, accessible through the system interface, and visible to the intended audience (students).
6. **Confirmation Response:** A confirmation response is sent back to the user (e.g., "Notice posted successfully").

#### Final Outcome:

- The notice is posted and visible to all intended users (students).
- Relevant users receive timely updates and announcements, ensuring clear communication within the department.

#### Unsuccessful Attempt:

1. **Unauthorized User:** If the user is not authorized (not a CR), the system returns an error response (e.g., "You are not authorized to manage department notices").
2. **Invalid or Missing Inputs:** If required fields (Notice Title, Notice Description, Date, Expiry Date, Audience Type) are missing or invalid, the system returns a validation error message (e.g., "All notice details are required").
3. **Database Error:** If the database operation fails (e.g., due to connectivity issues or internal errors), the system logs the error for further investigation. The user receives a failure message (e.g., "Failed to post the notice. Please try again later"). No notice is created or updated.
4. **Unexpected Errors:** Any unexpected errors during the process are caught, and the user receives a generic error message (e.g., "An error occurred. Please try again later"). The error is logged for debugging purposes.

### 3.1.10 Send Notices for Batch Updates

**As a** Class Representative (CR),

**I want** to send notices about batch updates, such as exams, class leaves, and other important information, **so that** the relevant batch members are informed about these updates promptly.

**Successful Attempt:**

- The system verifies that the user is authorized (CR/Admin) using role-based access control (RBAC).
- The system validates the required inputs (title, content, batch identifier).
- The notice details (title, content, batch identifier, created\_by, timestamp) are stored in the Notices table.
- The system maps the batch identifier to the batch members using the relevant foreign key.
- A confirmation response is sent back to the user (e.g., "Notice sent successfully").
- The notice is visible in the notice board or batch-specific section for relevant batch members.

**Final Outcome:**

- The department members receive the notice and can view it in the department's notice board.
- The notice is accessible to the intended audience (batch students) as per the audience type.
- Department members are notified promptly, ensuring timely communication of critical information.

**Unsuccessful Attempt:**

- **If the user is not authorized (not a CR/Admin):**
  - The system returns an error response (e.g., "You are not authorized to send notices").
- **If required fields (title, content, batch identifier) are missing or invalid:**
  - The system returns a validation error message (e.g., "Notice title, content, and batch identifier are required").
- **If the database operation (inserting the notice) fails due to connectivity or internal error:**
  - The system logs the error for further investigation.
  - The user receives a failure message (e.g., "Failed to send the notice. Please try again later").
  - No notice is created or sent.
- **Any unexpected errors during the process are caught:**
  - The user receives a generic error message (e.g., "An error occurred. Please try again later").
  - The error is logged for further debugging.

### 3.1.11 Send Department Notice

**As a** Department Admin or Faculty,

**I want** to send department-wide notices, such as events, meetings, and other important information, **so that** all department members are informed in a timely manner.

**Successful Attempt:**

- The system verifies that the user is authorized (Department Admin/Faculty) using role-based access control (RBAC).
- The system validates the required inputs (title, content, audience type, department identifier).
- The notice details (title, content, audience type, department identifier, created\_by, timestamp) are stored in the Notices table.
- The system maps the department identifier to the relevant department members using the appropriate foreign key.
- A confirmation response is sent back to the user (e.g., "Department notice sent successfully").
- The notice is visible in the department notice board or department-specific section for all relevant members.

**Final Outcome:**

- Department members receive the notice and can view it on the department's notice board.
- The notice is accessible to the intended audience (faculty, staff, students) as per the audience type.
- Department members are notified promptly, ensuring timely communication of critical information.

**Unsuccessful Attempt:**

- **If the user is not authorized (not a Department Admin/Faculty):**
  - The system returns an error response (e.g., "You are not authorized to send department notices").
- **If required fields (title, content, audience type, department identifier) are missing or invalid:**
  - The system returns a validation error message (e.g., "Notice title, content, audience type, and department identifier are required").
- **If the database operation (inserting the notice) fails due to connectivity or internal error:**
  - The system logs the error for further investigation.
  - The user receives a failure message (e.g., "Failed to send the department notice. Please try again later").
  - No notice is created or sent.
- **Any unexpected errors during the process are caught:**
  - The user receives a generic error message (e.g., "An error occurred. Please try again later").
  - The error is logged for further debugging.

### 3.1.12 Handle Job and Internship Information

**As an Admin or Designated Staff,** I want to post, view, and manage job and internship opportunities related to the department, so that relevant users can access available opportunities and apply for them.

**Successful Attempt:**

- The system verifies that the user is authorized (Admin or designated staff) using role-based access control (RBAC).
- The system validates the required inputs (Job/Internship Title, Description, Application Deadline, Contact Information).
- The job or internship details (title, description, deadline, contact information) are stored in the system.
- A confirmation response is sent back to the user (e.g., "Job/Internship opportunity posted successfully").
- The system updates the list of available job/internship opportunities, making it visible to all relevant users.

**Final Outcome:**

- Users can view the posted job/internship opportunities in the job listings section.
- Job or internship opportunities are displayed with all relevant details, and users can access them to apply.
- The system ensures timely updates for the available opportunities, keeping the list current for department members.

**Unsuccessful Attempt:**

- If the user is not authorized (not an Admin or designated staff):
  - The system returns an error response (e.g., "You are not authorized to post job or internship opportunities").
- If required fields (e.g., title, description, deadline, contact information) are missing or invalid:
  - The system returns a validation error message (e.g., "Job/Internship title, description, deadline, and contact information are required").
- If the database operation (inserting the job/internship details) fails due to connectivity or internal error:
  - The system logs the error for further investigation.
  - The user receives a failure message (e.g., "Failed to post the job/internship opportunity. Please try again later").
  - No job or internship is created or posted.
- Any unexpected errors during the process are caught:
  - The user receives a generic error message (e.g., "An error occurred. Please try again later").
  - The error is logged for further debugging.

## 3.2 Non-Functional Requirements (NFRs)

### 3.2.1 Performance

- The system shall load course details and batch information within 2 seconds for any user request.
- The system should support up to 100 concurrent users without degradation in performance.

### 3.2.2 Scalability:

- The system shall be scalable to handle an increasing number of students, courses, and batches as the institution grows.
- The architecture should allow for easy expansion, such as adding more servers or upgrading storage.

### 3.2.3 Usability:

- The user interface shall be intuitive and user-friendly for both students and admins, with minimal learning curve.
- The system shall be compatible with major browsers (Chrome, Firefox, Safari, Edge) and provide a responsive design that works seamlessly on mobile devices and desktops.

### 3.2.4 Security:

- The system shall ensure that only authenticated users (students and admins) have access to the system features.
- Admin-level functionalities (e.g., managing batch information) shall be restricted to authorized personnel only.
- All sensitive data (such as passwords) shall be encrypted in transit and at rest.

### 3.2.5 Reliability:

- The system shall have an uptime of 99.9%, ensuring that it is available to users at all times, except during scheduled maintenance.
- It should have automatic backup and recovery mechanisms to prevent data loss in case of failure.

### 3.2.6 Maintainability:

- The system shall be designed with modularity to allow easy updates and bug fixes without affecting other parts of the system.
- All system components should be documented to ensure smooth handover and maintenance by developers.

### 3.2.7 Data Integrity:

- The system shall ensure that data consistency is maintained across the database, with validation rules preventing duplicate or conflicting entries.
- Course and batch information shall be updated in real time to reflect the most current data.

### 3.2.8 Availability:

- The system should be accessible 24/7 to allow students and administrators to access course and batch information at any time.
- Any scheduled downtime for maintenance should be notified at least 24 hours in advance.