Namal University, Mianwali

Software Requirements Specification

University Timetable Management System

Version 1.0

28-December-2025

Authors:

Hammad Shabbir
Ahmer Sultan
Hasnain Ali

Requirement Provider (Client):

Mr. Abdul Rafay

Table of Contents

1 Introduction

1.1 Overview

This Software Requirements Specification (SRS) document is a compilation of the overall requirement of the University Timetable Management System to be designed to be utilized at Namal University, Mianwali. It is mainly used to guide the development process and make it meet the expectations of the entire stakeholders, such as the developers, and end users.

1.2 Purpose

The following Software Requirements Specification document will outline and specify the functional requirements and non-functional requirements of the **University Timetable Management System**. This document specifies the system functions, constraints and quality features that shall be used by the development team in the development process.

1.3 Product Scope

1.3.1 Scope Coverage

- Basic scheduling and schedule control.
- Instructors and venue conflict detection and resolution.
- Multi-user access e.g. Admin, Faculty, Student etc.
- Timetable notification system via email.
- Basic reporting
- Adding, deleting and managing user accounts.
- Classroom, Labs Resource management.

1.3.2 Scope Exclusions

- Native mobile applications
- Full Support for more than one university.
- AI conflict resolution.

1.4 Definitions, Acronyms, and Abbreviations

Term	Definition
Conflict	Overlapping class schedules
Time Slot	Specific day/time for a class
Venue	Classroom/lab where class is held
Draft Timetable	Being created/edited, not yet published
Published Timetable	Timetable which is visible to users
Soft Delete	A data management technique where data is flagged as inactive rather than removing it physically.

Table 1: Definitions

1.4.0.1 Definitions

Acronym	Full Form
SRS	Software Requirements Specification
UTMS	University Timetable Management System
UI/UX	User Interface/User Experience
RBAC	Role-Based Access Control
SMTP	Simple Mail Transfer Protocol
CSV	Comma-Separated Values
SQL	Structured Query Language
HTTPS	Hypertext Transfer Protocol Secure
XSS/CSRF	Cross-Site Scripting/Cross-Site Request Forgery
FAQ	Frequently Asked Questions
ToS	Terms of Service

Table 2: Acronyms and Abbreviations

1.4.0.2 Acronyms and Abbreviations

1.5 Key Business Goals

1. **Arrange Scheduling Conflicts**: Eliminate scheduling conflicts in classes and venue conflicts by conflict detection.
2. **Centralize Timetable Management:** Have the system to replace email and informal coordination with a single source of information.

3. **Support Multiple Departments:** Manage schedules of various departments at the same time.

1.6 Intended Audience

- **Software Developers:** To understand the system and use requirements for development process.
- **Project Managers:** To track and manage the progress of the development process.
- **Quality Assurance Team:** Generate test cases with help of requirements explained in this document.
- **Stakeholders:** To ensure that system meets the expectation and needs of the organization and aligns with its policies.

1.7 References

Document	Date	Source
Project Proposal: University Timetable Management System	Nov 9, 2025	Namal University, CSC-225
IEEE Recommended Practice for Software Requirements Specifications (IEEE 830-1998)	1998	IEEE
IEEE Recommended Practice for Software Requirements Specifications	2021	Studocu
General Data Protection Regulation (GDPR)	2018	European Union

Table 3: References

2 General Description

2.1 Product Perspective

2.1.1 System Context

The University Timetable Management System is a web based application that consists of a self built application. It is intended to operate in isolation on behalf of the university.

2.1.2 System Boundaries

The system is an independent application, which has the following external interfaces:

- **Email Server:** To send notification emails to users.
- **Web Browsers**: Chrome, Firefox, Safari, Edge etc. to access by users.
- **Database:** Import/export functions of backup

2.2 Product Functions

2.2.1 Major System Functions

1. **Timetable Management**
    - Create/Manage timetables
    - Edit and manage previous timetables
    - Store timetables for historical records
    - Import/export timetables
2. **Conflict Detection & Resolution**
    - Detection and Resolution of Conflict
    - Suggest resolution options
    - Track conflict resolution status
    - Generate conflict reports
3. **User & Access Management**
    - Authenticate users with email and password)
    - Create/Manage user accounts
    - Role-based access control
    - Reset forgotten passwords
    - Mass user import through CSV/Excel.
4. **Notification System**
    - Send notification email for timetable updates
    - In app notifications for system updates
    - Scheduled reminders
5. **Reporting**
    - Generate summary reports
    - Report on conflict and resolution tracking.
    - Logs of all system activities
6. **Resource Management**
    - Maintain venue or classrooms database
    - Track lab availability
    - Prevent twice resource booking.

7. **Dashboard**
    - Admin dashboard
    - Faculty dashboard displaying assigned courses
    - Student dashboard showing class schedule
    - Search and filter functions
8. **System Configuration**
    - Set up academic calendar (semesters, holidays)
    - Configure time slots and class durations
    - Manage notification settings

2.3 User Classes and Characteristics

User Class Characteristics:

- **System Administrator:** Trained and have wide system knowledge.
- **Faculty:** Has technical skills, and he/she may request support.
- **Student:** Has technical skills, and primarily reads schedule.
- **Coordinator:** Has administrative background and is comfortable with data entry. He/She acts as bridge between faculty and admin.
- **IT Support:** Has technical skills.

2.4 General Constraints

In this section, the organizational, regulatory, technical, and schedule limitations are described that should define how the system is designed, developed, deployed, and maintained are outlined. These limits are required limits within which the system should be working.

2.4.0.1 Organizational and Regulatory Constraints

2.4.0.1.1 University Policies Compliance The system shall fully comply with all University policies, including but not limited to:

- Privacy and protection policy on data,
- Information security regulations,
- Policy of acceptable use and access control.

Any subsequent changes on University policies should be checked and incorporated in the system where a necessity exists.

2.4.0.1.2 Data Confidentiality and Data Privacy

- The system shall ensure the confidentiality and integrity of all academic, administrative and data related to the user.
- The access to the sensitive information shall be limited according to user roles and privileges.

2.4.0.1.3 Data Retention and Archival The system shall hold the records of historical timetables to utilize in auditing, compliance, and reference purposes.

- The history of timetables shall be archived **at least three (3) years** long.
- The archived information shall be accessible but shall not be edited by an user without appropriate authorization.

2.4.0.1.4 Legal and Regulatory Constraints The system shall comply with national and institutional data protection laws (e.g. data protection laws according to the jurisdiction of the institution).

2.4.0.2 Technical Constraints

2.4.0.2.1 Technology Stack Limitations The approved technologies that shall be used in developing the system include:

- Backend: Node.js
- Frontend: HTML, CSS, and JavaScript using either **React** or **Angular** framework
- Database: MySQL or PostgreSQL
- Web Server: Apache or Nginx
- UI/UX Design Tool: Figma

The application of alternative technologies shall have to be pre-approved by the project stakeholders.

2.4.0.2.2 System Architecture The system shall be based on a modular and scalable design and shall implement Role-based access control (RBAC) to provide secure and authorized access to various user groups that may include administrators, faculty, and students.

2.4.0.2.3 Design Constraints in Database

- The database shall be developed using **normalized relational schemas** to reduce redundancy and avoid occurrence of data anomalies.
- In order to enhance performance and data consistency, referential integrity, indexing and constraints shall be used.

2.4.0.2.4 Code Quality and Standards

- The source code shall follow consistent coding naming standards to ensure maintainability and readability.
- The source code shall be documented using comments and external documents to support future improvements.
- Secure coding practices shall be employed to avoid common system vulnerabilities (e.g.SQL injection).
- The system shall be reviewed for quality of code before deployement.

2.4.0.2.5 Compatibility and Deployment The system shall be compatible with the latest versions of major web browsers and shall be deployed in servers supported by University IT infrastructure environment with the University IT infrastructure support.

2.4.0.3 Schedule Constraints

2.4.0.3.1 Development Timeline The system development shall be completed in **five (5) major phases** that shall take a total of **12 months** to be completed as specified in the Section 2.5 of the proposal.

2.4.0.3.2 Phase Dependencies The various phases of development shall be done and verified before moving to the next stage. Slowdown of the previous phases can affect the project schedule.

2.5 Assumptions and Dependencies

The details below explains the assumptions that were used in the creation of the University Timetable Management System and the dependencies on which the system depends to run correctly.

2.5.1 Assumptions

2.5.1.1 User Data Availability The assumption is that the existing user information and detail such as faculty, student and course information can be imported either via the official records of the University or in case of Excel/CSV file.

2.5.1.2 Academic Calendar Provided The University will also provide the academic calendar to set the dates of the semester, holidays and other scheduling variables in the system.

2.5.1.3 Constant Network connectivity The system is assumed to be stable in terms of internet connectivity to access the system.

2.5.1.4 Email System Available The University has a running SMTP email server to send notifications, alerts, and reminders to users of the system.

2.5.1.5 Data Completeness All information needed in the generation of the timetable such as courses, instructors and places is supposed to be complete and correct before scheduling.

2.5.1.6 One University Operation The system will support one university. It is not supposed to work in multi-university environments.

2.5.1.7 Instructor Availability The teachers are supposed to inform the administration about being unavailable ahead of time to enable the administration to plan the schedule.

2.5.2 Dependencies

2.5.2.1 External Systems

- **Email Server:** This is required for sending notifications and alerts.
- **Operating System:** Linux is needed for server deployment.
- **Web Browser**: The user needs to have a newer version of a web browser which supports JavaScript.

2.5.2.2 Third-Party Libraries and Frameworks

- **Frontend Framework:** React/Angular for user interface.
- **Backend Framework:** Node.js or Node.js-based framework.
- **Database System**: Postgresql/MySQL as a database system.
- **Authentication Libraries:** bcrypt/Argon2 for secure password hashing and authentication.

2.5.2.3 Project Dependencies

- Stakeholder approval of scope and budget of project.
- University IT infrastructure for hosting, networking and server maintenance is available.
- Availability of client to provide feedback and clarify their requirements on a regular basis.
- Presence of Scrum Master to plan the sprints, handle backlog, and oversee development.

2.5.2.4 Data Dependencies

- Proper and complete information of the course.
- Available schedules of instructors.
- Classroom and lab capacity data.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Timetable Management Module

3.1.1.1 FR1: Create Timetable

3.1.1.1.1 Introduction The system shall allow administrators to create new timetables for departments with date, course, instructor, and venue assignments.

3.1.1.1.2 Inputs

- Department ID
- Course codes
- Instructor IDs
- Venue IDs
- Date ranges
- Time slots

3.1.1.1.3 Processing The system first checks the input data against the existing schedules. If it finds any conflicts, it uses the conflict detection rule to identify them. If all checks pass, it saves the new timetable as a draft. If the check fails, then the system shows specific error messages. These messages explain exactly which rules were broken.

3.1.1.1.4 Outputs

- New draft timetable saved in the database with a unique ID
- Confirmation message displayed to the administrator

3.1.1.2 FR2: Edit Timetable

3.1.1.2.1 Introduction The system shall allow administrators to modify existing timetable entries and save changes with version control.

3.1.1.2.2 Inputs

- Timetable ID
- Fields to modify (time, venue, instructor)
- Edit reason

3.1.1.2.3 Processing To make an edit, the system loads the current version, applies changes, and runs a conflict check. It then saves the update with a timestamp and user ID, maintaining a version history as required by FR10.

3.1.1.2.4 Outputs

- Updated timetable version
- Logged version history
- Success notification shown

3.1.1.3 FR3: Delete Timetable

3.1.1.3.1 Introduction The system shall allow administrators to mark draft or expired timetables as delete (soft-delete) as per **BR3**.

3.1.1.3.2 Inputs

- Timetable ID
- Confirmation of deletion

3.1.1.3.3 Processing The system first checks the timetable's status. If it is published, deletion is not allowed and an error message is shown. For draft or expired timetables, the system performs a soft delete. This means the timetable is marked as "deleted" and hidden from normal views. A record of when and by whom it was deleted is saved.

3.1.1.3.4 Outputs

- Deletion recorded in log
- Error if trying to delete published timetable
- Notification to the administrator

3.1.1.4 FR4: Conflict Detection

3.1.1.4.1 Introduction The system shall automatically detect and report scheduling conflicts in real-time as entries are added or modified, including instructor overlap, venue double-booking, and student schedule overlap.

3.1.1.4.2 Inputs

- New or modified timetable entry (time, venue, instructor, student)

3.1.1.4.3 Processing The system compares the new entry against existing schedules, checking for overlaps in instructor availability, venue bookings, and student enrollments. It then flags conflicts.

3.1.1.4.4 Outputs

- On-screen warning if a conflict is detected
- Details about conflict

3.1.1.5 FR5: Conflict Resolution Suggestions

3.1.1.5.1 Introduction The system shall suggest alternative time slots, instructors, or venues to resolve identified conflicts.

3.1.1.5.2 Inputs

- Conflict ID
- Current timetable constraints

3.1.1.5.3 Processing The system analyzes available resources and open slots to generate feasible alternatives ranked by suitability.

3.1.1.5.4 Outputs

- List of suggested alternatives
- Menu for the administrator to select

3.1.1.6 FR6: Publish Timetable

3.1.1.6.1 Introduction The system shall allow administrators to approve and publish final timetables, making them visible to authorized users. The system shall implement BR5 preventing publishing timetable if there is a conflict.

3.1.1.6.2 Inputs

- Timetable ID
- Administrator approval confirmation

3.1.1.6.3 Processing The system scans the timetable if it does not find any conflict, it marks the timetable as "published," notifies relevant users (faculty and students), and updates dashboards, otherwise it displays error message saying "Timetable with conflict can not be published".

3.1.1.6.4 Outputs

- Notification to relevant users via email or in-app
- Status changed to "Active"
- Error message if publishing timetable with conflict

3.1.1.7 FR7: Archive Timetable

3.1.1.7.1 Introduction The system shall automatically archive published timetables after the term ends for history.

3.1.1.7.2 Inputs

- Term end date
- Timetable status

3.1.1.7.3 Processing The system scans for timetables after their end date, if it a timetable is expired, it moves it to archive storage, and updates its access permissions to "read-only."

3.1.1.7.4 Outputs

- Archived timetable

3.1.1.8 FR8: Import and Export Timetable

3.1.1.8.1 Introduction The system shall support importing timetables from previous terms and exporting timetables to PDF or Excel formats.

3.1.1.8.2 Inputs

- For import: CSV or Excel file
- For export: Timetable ID and format selection

3.1.1.8.3 Processing For import, the system validates file format and then, runs full conflict check. if conflicts are found, it moves timetable to draft and flags issues for review. For export, the system generates a file in the selected format with proper formatting.

3.1.1.8.4 Outputs

- Import: Data loaded into draft timetable
- Export: File downloaded (PDF or Excel) to user's device

3.1.1.9 FR9: Bulk Operations

3.1.1.9.1 Introduction The system shall allow administrators to perform bulk edit operations on multiple timetable entries simultaneously.

3.1.1.9.2 Inputs

- Selection of multiple entries
- Operation type (change venue, shift time)
- New values

3.1.1.9.3 Processing The system applies the specified change to all selected entries, validates changes in batch, and updates the database.

3.1.1.9.4 Outputs

- Updated entries
- Summary report of changes
- Errors if any

3.1.1.10 FR10: Timetable Versioning

3.1.1.10.1 Introduction The system shall maintain version history of all timetables with timestamps and change logs.

3.1.1.10.2 Inputs

- Timetable ID
- User actions (create, edit, publish)

3.1.1.10.3 Processing The system logs each action with user ID, timestamp, changed fields, and previous values in a version history table.

3.1.1.10.4 Outputs

- Complete version history in administrator panel

3.1.2 Conflict Management Module

3.1.2.1 FR11: Conflict Report

3.1.2.1.1 Introduction The system shall generate detailed conflict reports with affected courses, instructors, venues.

3.1.2.1.2 Inputs

- Date range
- Department filter
- Conflict status filter

3.1.2.1.3 Processing The system searches through all conflicts based on the filters set by the user and compiles them into a structured report.

3.1.2.1.4 Outputs

- Downloadable or viewable report (PDF/Excel)
- Lists each conflict with details, and current status

3.1.2.2 FR12: Track Resolution Status

3.1.2.2.1 Introduction The system shall track the status of each conflict (Unresolved, In Review, Resolved, Acknowledged).

3.1.2.2.2 Inputs

- Conflict ID
- New status e.g. mark as "Resolved"

3.1.2.2.3 Processing The system updates the conflict's record with the new status, the time of the change, and the admin's ID. The admin can also add notes explaining how it was resolved.

3.1.2.2.4 Outputs

- Conflict status updated in the system
- Reflected in all views and reports

3.1.2.3 FR13: Automated Conflict Resolution

3.1.2.3.1 Introduction The system shall suggest automated resolution options for conflicts where applicable using constraint-based scheduling.

3.1.2.3.2 Inputs

- Specific conflict ID
- System's knowledge of available resources (free time slots, alternate venues/instructors)

3.1.2.3.3 Processing When a scheduling conflict is identified, the system analyzes the problem. It looks for open rooms and time slots, then checks each option against the rules and constraints. It presents these possible solutions. A typical suggestion might be as: "Fix: Move CS101 to Room B at 10 AM."

3.1.2.3.4 Outputs

- List of suggested resolution actions
- Presented to the administrator who can approve or ignore them

3.1.3 User Management Module

3.1.3.1 FR14: User Registration

3.1.3.1.1 Introduction The system shall support user account creation for Faculty, Students, and Course Coordinators with unique identification.

3.1.3.1.2 Inputs

- User details (name, email, role, department)
- Chosen password (must meet NFR15)

3.1.3.1.3 Processing During account creation, the system first validates the email address for uniqueness, then checks password strength. Then, password is hashed and system generates a unique identifier for the user and stores the complete profile record in the database.

3.1.3.1.4 Outputs

- New user account
- Confirmation email sent
- Error message if email is not valid or already exists or password does not meet requirements

3.1.3.2 FR15: User Login and Authentication

3.1.3.2.1 Introduction The system shall authenticate users via email/ID and password with secure password hashing (bcrypt/Argon2 or similar) as per NFR23.

3.1.3.2.2 Inputs

- Email or ID
- Password

3.1.3.2.3 Processing The system verifies credentials against the database, if they matches it opens a session. The session expires after 30 minutes of inactivity as per NFR22.

3.1.3.2.4 Outputs

- Successful login: access granted to the dashboard
- Failed login: error message shown
- Login attempt is logged

3.1.3.3 FR16: Role-Based Access Control

3.1.3.3.1 Introduction The system shall enforce role-based permissions: Admin (full access), Faculty (view own schedule), Student (view own schedule), Coordinator (view/-validate department data).

3.1.3.3.2 Inputs

- User role
- Requested action

3.1.3.3.3 Processing Before any action, system checks user's role. If user has appropriate role for the action, it allows it to be performed.

3.1.3.3.4 Outputs

- User is granted or denied access to specific system functions based on their role

3.1.3.4 FR17: User Profile Management

3.1.3.4.1 Introduction The system shall allow users to view and update their profile (name, email, department, contact information).

3.1.3.4.2 Inputs

- Updated profile fields submitted by the user

3.1.3.4.3 Processing The system checks new information for correct formats. If the email is being updated, it makes sure no one else is using it. It then saves the updates to the user's account.

3.1.3.4.4 Outputs

- Updated Profile

3.1.3.5 FR18: Password Reset

3.1.3.5.1 Introduction The system shall provide password reset functionality via email verification with validity of one hour as per NFR25.

3.1.3.5.2 Inputs

- User's registered email address

3.1.3.5.3 Processing The system generates a secure reset token, sends an email with a reset link. It checks for validity of link, if it has expired it shows an error message otherwise it lets user reset his/her password.

3.1.3.5.4 Outputs

- Password reset email
- Error message if link has expired or invalid

3.1.3.6 FR19: Bulk User Import

3.1.3.6.1 Introduction The system shall support bulk user creation via CSV or Excel file upload. Passwords shall be system-generated.

3.1.3.6.2 Inputs

- CSV or Excel file containing user details (name, email, role, department)

3.1.3.6.3 Processing An admin can bulk-create users by uploading a file. The system validates each user in the file, to make sure that emails are correct and unique. For valid entries, it creates accounts with secure passwords and sends welcome emails. Invalid entries are skipped, and an error report is generated.

3.1.3.6.4 Outputs

- Bulk users accounts
- Summary report for the administrator

3.1.3.7 FR20: User Deactivation

3.1.3.7.1 Introduction The system shall allow administrators to deactivate user accounts without deleting data.

3.1.3.7.2 Inputs

- User ID
- Administrator confirmation

3.1.3.7.3 Processing The system sets the user account status to "inactive". This blocks user's login but account data is still preserved (soft-delete).

3.1.3.7.4 Outputs

- User account deactivated
- Confirmation notification to the administrator

3.1.3.8 FR21: Multi-Factor Authentication

3.1.3.8.1 Introduction The system shall support optional MFA (OTP via email/SMS) for enhanced security.

3.1.3.8.2 Inputs

- User's email or phone number
- One-time password (OTP) entered by the user

3.1.3.8.3 Processing If MFA is enabled, the system generates and sends an OTP after password verification, then validates the entered OTP.

3.1.3.8.4 Outputs

- Successful MFA: grants login
- Failed OTP: blocks login

3.1.4 Notification and Communication Module

3.1.4.1 FR22: Email Notifications

3.1.4.1.1 Introduction The system shall send email notifications to faculty and students when timetables are published or updated.

3.1.4.1.2 Inputs

- Timetable ID
- List of affected users (faculty and students)
- Type of change (publish/update)

3.1.4.1.3 Processing The system composes an email with timetable details and change summary, then sends it via a configured email service.

3.1.4.1.4 Outputs

- Email notifications delivered to users' registered email addresses
- Delivery status logged

3.1.4.2 FR23: In-App Notifications

3.1.4.2.1 Introduction The system shall display in-app notifications for timetable changes, conflicts, and system updates.

3.1.4.2.2 Inputs

- Event trigger (timetable edit, conflict detection)
- Target user(s)

3.1.4.2.3 Processing The system creates a notification message, stores it in the user's notification queue (keeps for 30 days), and displays it on their dashboard when they next log in.

3.1.4.2.4 Outputs

- Notification appears in the user's notification panel

3.1.4.3 FR24: Scheduled Notifications

3.1.4.3.1 Introduction The system shall support scheduled notifications e.g. class reminders 1 hour before.

3.1.4.3.2 Inputs

- Class schedule data
- User-defined reminder rules (time before class)

3.1.4.3.3 Processing The system checks upcoming classes every 15 minutes against the current time. When a class matches the reminder criteria, the system triggers notifications in-app.

3.1.4.3.4 Outputs

- Notifications sent to users at the scheduled time

3.1.4.4 FR25: Notification Preferences

3.1.4.4.1 Introduction The system shall allow users to configure notification settings (frequency, channels).

3.1.4.4.2 Inputs

- User's selected preferences (e.g., disable email reminders, enable in-app alerts)

3.1.4.4.3 Processing The system saves preferences to the user's profile and applies them when generating notifications.

3.1.4.4.4 Outputs

- User's notification settings

3.1.5 Reporting Module

3.1.5.1 FR27: Timetable Summary Report

3.1.5.1.1 Introduction The system shall generate summary reports (courses per instructor, venue utilization, time slot usage).

3.1.5.1.2 Inputs

- Date range
- Department filter
- Report type

3.1.5.1.3 Processing The system aggregates timetable data to calculate metrics like courses taught per instructor, venue occupancy, and time slot usage percentages.

3.1.5.1.4 Outputs

- Summary report (PDF/Excel)

3.1.5.2 FR28: Resource Utilization Report

3.1.5.2.1 Introduction The system shall report on classroom and lab usage, capacity, and utilization rates.

3.1.5.2.2 Inputs

- Resource type (room/lab)
- Date range
- Department selected

3.1.5.2.3 Processing The system calculates usage hours for each resource, compares against total available hours, and determines utilization percentages per resource.

3.1.5.2.4 Outputs

- Utilization report showing each resource's usage

3.1.5.3 FR29: Audit Log Report

3.1.5.3.1 Introduction The system shall provide audit logs of all system activities (timetable changes, user actions, conflicts).

3.1.5.3.2 Inputs

- User ID (optional filter)
- Action type (optional filter)
- Date range filters

3.1.5.3.3 Processing The system retrieves log entries from the audit table, filters by criteria, and sorts in order.

3.1.5.3.4 Outputs

- Comprehensive audit log report

3.1.6 Resource Management Module

3.1.6.1 FR30: Venue and Classroom Management

3.1.6.1.1 Introduction The system shall maintain a database of all venues with capacity, facilities, and availability.

3.1.6.1.2 Inputs

- Venue details (name, capacity, facilities list, availability schedule)

3.1.6.1.3 Processing The system stores and updates venue information and checks availability against scheduled classes when timetables are created.

3.1.6.1.4 Outputs

- Updated venue records
- Errors for invalid data

3.1.6.2 FR31: Lab Management

3.1.6.2.1 Introduction The system shall track lab availability, and booking status.

3.1.6.2.2 Inputs

- Lab details (name, equipment, capacity)
- Booking schedules

3.1.6.2.3 Processing The system manages lab records separately from classrooms and validates bookings and prevents double-booking.

3.1.6.2.4 Outputs

- Lab availability
- Prevented booking conflicts

3.1.6.3 FR32: Resource Constraints

3.1.6.3.1 Introduction The system shall implement constraints e.g. classroom capacity, lab availability.

3.1.6.3.2 Inputs

- Class size
- Resource requirements
- Selected venue/lab during timetable creation

3.1.6.3.3 Processing The system validates that the selected resource meets all constraints (capacity, availability) before finalizing the schedule.

3.1.6.3.4 Outputs

- Scheduling fails with an error message if requirements are not met

3.1.6.4 FR33: Resource Conflict Detection

3.1.6.4.1 Introduction The system shall prevent double-booking of venues or labs through conflict detection.

3.1.6.4.2 Inputs

- Timetable entry with selected venue/lab and time slot

3.1.6.4.3 Processing The system checks the resource's booking schedule in real-time for double reservations during the same period.

3.1.6.4.4 Outputs

- Double-booking attempts are blocked
- Administrator receives an conflict alert with details

3.1.7 Course and Department Management

3.1.7.1 FR34: Course Registry

3.1.7.1.1 Introduction The system shall maintain a course database (code, name, instructor, number of sections).

3.1.7.1.2 Inputs

- Course details

3.1.7.1.3 Processing The system stores course information, and links it to the relevant department and instructor records.

3.1.7.1.4 Outputs

- Course added to the system
- Available for selection during timetable creation

3.1.7.2 FR35: Department Management

3.1.7.2.1 Introduction The system shall support management of multiple departments with separate course offerings.

3.1.7.2.2 Inputs

- Department details
- Associated courses

3.1.7.2.3 Processing The system organizes courses, instructors, and timetables under departments, implements department boundaries in scheduling to prevent cross-department conflicts unless allowed.

3.1.7.2.4 Outputs

- Departments are listed
- Scheduling and reporting are filtered by department

3.1.7.3 FR36: Course Constraints

3.1.7.3.1 Introduction The system shall allow specifying course-specific constraints e.g. preferred time slots.

3.1.7.3.2 Inputs

- Constraints defined per course
- Example: "Must be scheduled in 1st slot"

3.1.7.3.3 Processing The system stores constraints as structured metadata for each course and considers them during automated scheduling and conflict checking.

3.1.7.3.4 Outputs

- Constraints are followed during timetable generation

3.1.8 Dashboard Module

3.1.8.1 FR37: Admin Dashboard

3.1.8.1.1 Introduction The system shall display admin dashboard with timetable overview, pending actions, conflict summary, and user statistics.

3.1.8.1.2 Inputs

- Admin login
- System data (timetables, conflicts)

3.1.8.1.3 Processing The system gets data from multiple modules and presents it in widgets or panels on a dashboard.

3.1.8.1.4 Outputs

- Interactive dashboard

3.1.8.2 FR38: Faculty Dashboard

3.1.8.2.1 Introduction The system shall display faculty dashboard with their assigned courses, schedules.

3.1.8.2.2 Inputs

- Faculty member's user ID and role
- Their assigned courses and timetable data

3.1.8.2.3 Processing The system retrieves and displays only the courses, class times, and venues assigned to that faculty member. The data is displayed in calender view.

3.1.8.2.4 Outputs

- Dashboard showing weekly schedule, class details
- Class details like course name, students enrolled, venue, time
- Option to export schedule

3.1.8.3 FR39: Student Dashboard

3.1.8.3.1 Introduction The system shall display student dashboard with class schedule, venue information, and timetable updates.

3.1.8.3.2 Inputs

- Student's user ID
- Enrolled courses
- Latest timetable

3.1.8.3.3 Processing The system gets only published timetables, retrieves the student's enrolled courses, filters the timetable to show only their classes and displays it in calendar view.

3.1.8.3.4 Outputs

- Clear schedule showing classes, times, venues
- Notifications about schedule changes
- Option to export schedule

3.1.8.4 FR40: Filter and Sort

3.1.8.4.1 Introduction The system shall support filtering and sorting of timetables by department, day, instructor, venue.

3.1.8.4.2 Inputs

- User-selected filters
- Example: department = "Computer Science", day = "Monday"

3.1.8.4.3 Processing The system applies filters to the timetable data set, reorders results, and updates the display.

3.1.8.4.4 Outputs

- Filtered and sorted timetable

3.1.9 System Configuration Module

3.1.9.1 FR41: Academic Calendar Setup

3.1.9.1.1 Introduction The system shall allow admin to configure academic calendar e.g. semesters, holidays, exam periods.

3.1.9.1.2 Inputs

- Academic year start/end dates
- Semester periods
- Holidays
- Exam schedules

3.1.9.1.3 Processing The system saves calendar events and checks them for errors, like making sure end dates come after start dates and that semesters don't overlap. These dates are used to set boundaries for the schedule. For instance, no classes can be set on holidays.

3.1.9.1.4 Outputs

- Configured academic calendar applied system-wide
- Errors if date ranges are illogical

3.1.9.2 FR42: Time Slot Configuration

3.1.9.2.1 Introduction The system shall allow admin to define available time slots and class durations.

3.1.9.2.2 Inputs

- Start/end times for day
- Break periods
- Slot durations

3.1.9.2.3 Processing he system stores templates for time slots, which have names and set start/end times. It checks that these slots do not overlap with scheduled breaks. These saved templates can then be chosen when building a timetable.

3.1.9.2.4 Outputs

- Time slot templates displayed in timetable creation interface
- Error if slots overlap

3.1.9.3 FR43: System Settings

3.1.9.3.1 Introduction The system shall support configuration of notification settings, email templates, and system parameters.

3.1.9.3.2 Inputs

- Admin-defined settings (email server details, default notification content)

3.1.9.3.3 Processing The system checks that the email server is reachable before saving its settings in case of email settings. It then stores the configuration and applies it everywhere it's needed for sending notifications.

3.1.9.3.4 Outputs

- System behavior customized (e.g., email sender name, notification triggers, UI preferences)

3.1.9.4 FR44: Constraint Configuration

3.1.9.4.1 Introduction The system shall allow admin to set global constraints (max classes per instructor, max students per class).

3.1.9.4.2 Inputs

- Constraint rules e.g. "max 4 classes per instructor per day"

3.1.9.4.3 Processing The system checks that new constraint values are valid, such as making sure numbers are positive and within acceptable limits. It then saves them as official rules for the entire system. These rules are implemented during all timetable operations: creation, editing, and auto-generation.

3.1.9.4.4 Outputs

- Constraints actively preventing schedule violations
- Error messages when constraints are violated

3.1.10 Data Integrity and Security

3.1.10.1 FR45: Data Validation

3.1.10.1.1 Introduction The system shall validate all input data (dates, times, capacity) before accepting.

3.1.10.1.2 Inputs

- User-provided data (dates, times, numbers, text fields) during any data entry operation

3.1.10.1.3 Processing The system checks all data for accuracy before saving it. This includes verifying formats (like dates and times), ensuring numbers are in a valid range, and confirming that the information is logical (for example, a start date must be before an end date). It also checks that any scheduled time slots match the system's configured periods.

3.1.10.1.4 Outputs

- Invalid inputs are rejected with error messages
- Valid data is stored in the database

3.1.10.2 FR46: Data Consistency

3.1.10.2.1 Introduction The system shall ensure data consistency across timetables, courses, and instructor assignments.

3.1.10.2.2 Inputs

- Related data entries (course codes, instructor IDs, timetable references)

3.1.10.2.3 Processing The system prevents actions that would damage the data. It stops users from deleting things that are still in use, like an instructor with active classes. If an item is updated, connected information can update too. But the system will not automatically delete data in a way that breaks important links.

3.1.10.2.4 Outputs

- Consistent data across modules
- Database integrity maintained

3.1.10.3 FR47: Access Control

3.1.10.3.1 Introduction The system shall enforce role-based access control for all features.

3.1.10.3.2 Inputs

- User role
- Requested action

3.1.10.3.3 Processing The system checks a user's permissions according to its role before granting access to any feature or data. Every attempt to access something whether allowed or blocked is recorded in the system's audit log. If a user does not have permission, access is denied and they see an appropriate error message.

3.1.10.3.4 Outputs

- Users can only access functions permitted for their role
- Unauthorized requests are blocked with "Access Denied" message

3.1.10.4 FR48: Data Backup

3.1.10.4.1 Introduction The system shall perform automatic daily backups of all timetable and user data.

3.1.10.4.2 Inputs

- System trigger (scheduled time)
- Database contents

3.1.10.4.3 Processing The system creates secure backups by taking a snapshot of the database and sending it to a separate backup server as per DBR-3. These backup files are compressed and then encrypted for security as per NFR17. All backups are kept for a minimum of 30 days as per DBR-2.

3.1.10.4.4 Outputs

- Daily backup files
- Backup success/failure logged
- Notification to administrator if backup failed

3.1.11 Timetable Generation Module

3.1.11.1 FR49: Automatic Timetable Generation

3.1.11.1.1 Introduction The system shall automatically create draft timetable with courses, instructors, venues, time slots, and constraints that are available.

3.1.11.1.2 Inputs

- Course information (course codes, names, etc.)
- Instructor availability schedules
- Venue/classroom capacity data
- Available time slots
- Academic constraints
- Semester dates and academic calendar

3.1.11.1.3 Processing The system first ensures all setup is complete and the data is correct. It then uses an algorithm to build a schedule. This process follows all the rules to place courses into time slots, rooms, and with instructors. The algorithm will run until it finds a workable schedule or reaches its limit. Once a draft schedule is made, the system checks it thoroughly for conflicts and provides a report of any problems.

3.1.11.1.4 Outputs

- Draft timetable
- Conflict report indicating any pending problems
- Timetable generation logs

3.1.11.2 FR50: Draft Review and Approval

3.1.11.2.1 Introduction The system shall enable administrators to review and modify the automatically generated timetable and approve it for publication.

3.1.11.2.2 Inputs

- Auto-generated draft timetable
- Admin user credentials and permissions
- Modification requests
- Previous timetable data for comparison

3.1.11.2.3 Processing The system shows the draft timetable on a screen where conflicts are highlighted. When an admin makes a change, the system checks it against the rules right away. The system keeps a record of every change, saves different versions of the draft, and logs the review activity. It checks for new conflicts after every edit. Admins can also compare the current draft to older versions to see what was changed

3.1.11.2.4 Outputs

- Review interface for administrators
- Modified timetable version
- Change history
- Conflict warnings due to manual changes
- Final approved timetable ready for publication

3.2 External Interface Requirements

3.2.1 Hardware Interfaces

Server Hardware:

Component	Specification
Processor	Minimum: Quad-core CPU at 2.5+ GHz, Recommended: Octa-core at 3.0+ GHz
RAM	Minimum: 64 GB for up to 1000 concurrent users (NFR3), Scale: 8 GB per new 100 users
Storage	Minimum: 1 TB SSD, Scalable according to growth, RAID is recommended
Network Interface	1 Gbps Ethernet, Minimum 100 Mbps throughput
Operating System	Linux (Ubuntu 20.04+) according to dependencies (5.4.2.1)

Table 4: Primary Server Hardware Requirements

Backup Server Hardware (NFR11):

- Same specifications to primary server
- Separate physical location for disaster recovery (DBR-3)

Client Hardware:

- No specific hardware requirements except standard computer/mobile
- Minimum screen size: 320px (mobile phones) to 2560px (large displays)

Network Requirements:

- Bandwidth: 1 Mbps minimum per user, 5+ Mbps recommended for better experience
- Latency: less than 100ms for better user experience
- Protocols: TCP/IP, HTTPS (443), SMTP (587/465)

3.2.2 Software Interfaces

Software Stack (Technical Constraints 5.2.1):

Component	Specification
Backend Framework	Node.js 18 LTS or higher
Frontend Framework	React 18+ or Angular 15+
Database Management System	PostgreSQL 18.x or MySQL 8.4.x (DBR-1)
Web Server	Apache 2.4+ or Nginx 1.18+
Authentication	bcrypt or Argon2 for password hashing (NFR23)

Table 5: Software Stack Interfaces

Database Interface:

Aspect	Specification
DBMS	PostgreSQL 18.x or MySQL 8.4.x (DBR-1)
Connection Protocol	TCP/IP with SSL/TLS encryption
Connection Pool	Minimum: 10 connections, Scale: 10 connections per 100 concurrent users, Maximum: 200 connections
Connection Timeout	30 seconds
Query Timeout	60 seconds for complex reports (FR27, FR30)
Data Backup	Automated daily backups at 2:00 AM (FR52, DBR-2)
Backup Format	SQL dump with gzip compression and AES-256 encryption

Table 6: Database Interface Specifications

Data Exchange Formats:

Data Type	Format
Timetable Export	PDF (printable), Excel (editable)
User Import	CSV, Excel
Report Export	PDF, Excel

Table 7: Data Exchange Formats

CSV/Excel Import Format:

- Character encoding: UTF-8
- CSV delimiter: Comma (,)
- Excel format: .xlsx

- Maximum file size: 50 MB
- Maximum rows: 10,000 records per import
- Date format: YYYY-MM-DD

3.2.3 Communications Interfaces

Email Communication (SMTP):

Aspect	Specification
Protocol	SMTP (Simple Mail Transfer Protocol)
Port	587 or 465 (SSL/TLS)
Security	TLS 1.2 or higher encryption (NFR16)
Authentication	Username/password (configured in FR47)
Sender Address	System email but can be configured e.g. timetable@namal.edu.pk
Email Format	HTML
Retry Logic	Up to 3 retry attempts (FR22)
Bounce Handling	Failed deliveries are logged

Table 8: Email Communication Interface

3.2.4 User Interfaces

General UI/UX Principles:

1. **Responsive Design:** Interface shall be responsive to desktop, tablet and mobile screen size.
2. **Consistency:** All the screens shall have similar patterns of designs.
3. **Clarity:** Information architecture shall be user-friendly and new user shall complete basic tasks without training, as mentioned in **NFR26**.

UI Components & Standards:

Component	Standard
Color Scheme	Primary: University brand colors, Status colors: green for success, red for error, yellow for warning, blue for information
Typography	Sans-serif font e.g. Inter, Roboto, Poppins, Minimum 16px body text
Icons	Font Awesome icons. SVG format is preferred.
Buttons	Consistent sizing and spacing, Clear primary action, Dangerous actions e.g. delete require confirmation
Error Messages	Actionable language

Table 9: UI Components & Standards

Key User Interfaces:

1. **Admin Dashboard (FR39)**
    - Timetable status, conflicts, pending actions
    - Action buttons: Create timetable, manage users, view reports
2. **Timetable Creation/Edit View**
    - Two-panel layout: Left=course list, Right=timetable grid
    - Drag-and-drop or form-based assignment
    - Undo/redo buttons (NFR32)
3. **Conflict Detection & Resolution View**
    - List of conflicts
    - Suggested solutions for each conflict
4. **Faculty Dashboard (FR40)**
    - Calendar view of assigned courses
    - Course details e.g. students, venue
    - Issue reporting form
    - Request form for unavailability
5. **Student Dashboard (FR41)**
    - Calendar view
    - Course cards with instructor, venue, time
6. **User Management Interface**
    - User list with search/filter
    - Bulk import from CSV
    - User creation form
    - Permission assignment view

3.3 Non-Functional Requirements

3.3.1 Performance

ID	Requirement	Metric
NFR1	Response Time	System shall load timetable page in less than or equal to 2 seconds on average connection (5 Mbps).
NFR2	Conflict Detection Speed	Conflict detection algorithm shall complete for timetable with 500+ courses within 5 minutes.
NFR3	Concurrent Users	System shall support minimum 1000 concurrent users without performance issues.
NFR4	Database Query Response	Database queries shall return results within 1 second.
NFR5	File Import/Export	Mass import of 1000+ users or export of large timetables shall complete within 5 minutes.

3.3.2 Scalability

ID	Requirement	Metric
NFR6	Database Scalability	Database shall efficiently handle records from 1000 to 100,000+ without any performance loss.
NFR7	Multi-Department Support	System shall efficiently handle 10+ departments with separate timetables.
NFR8	User Growth	System shall accommodate growth from 500 users to 10,000+ users without any changes in its infrastructure.

3.3.3 Reliability & Availability

ID	Requirement	Metric
NFR9	System Uptime	System shall have 99.5% availability with maximum 3.6 hours downtime per month during academic term.

ID	Requirement	Metric
NFR10	Backup & Recovery	System shall support recovery to any backup point within the 30-day retention period (DBR-2). Restoration shall complete within 24 hours of request.
NFR11	Failover	System shall support automatic failover to backup server if primary server fails.
NFR12	Data Integrity	System shall ensure zero data loss during normal operations and failover operations.
NFR13	Error Handling	System shall gracefully handle errors and display user-friendly messages instead of system errors.

3.3.4 Security

ID	Requirement	Metric
NFR14	Authentication	System shall require strong password (minimum of 8 characters, mix of uppercase, lowercase, numbers, symbols).
NFR15	Encryption in Transit	All data transmitted shall be encrypted using HTTPS/TLS 1.2 or higher.
NFR16	Encryption at Rest	Sensitive data (passwords, personal information) shall be encrypted using AES-256 encryption.
NFR17	SQL Injection Prevention	System shall avoid SQL injection attacks using parameterized queries.
NFR18	Cross-Site Scripting (XSS) Prevention	System shall sanitize all user inputs to prevent XSS attacks.
NFR19	Cross-Site Request Forgery (CSRF) Prevention	System shall implement CSRF tokens for all state-changing operations.
NFR20	Audit Logging	System shall maintain audit logs of all user actions (login, timetable modification, access).
NFR21	Session Management	System shall implement secure session management with timeout after 30 minutes of inactivity.

ID	Requirement	Metric
NFR22	Password Storage	System shall store passwords using bcrypt or other strong hashing algorithm.
NFR23	Role-Based Access Control	System shall implement RBAC to ensure that users access only authorized data.
NFR24	Password Reset	The system shall expire password reset links after period of one hour.

3.3.5 Usability

ID	Requirement	Metric
NFR25	Intuitive Interface	System interface shall be intuitive, new users shall accomplish basic tasks without training.
NFR26	Responsive Design	System shall be fully responsive on desktop, tablet, and mobile devices.
NFR27	User Documentation	System shall include comprehensive user guides, video tutorials, and help documentation.
NFR28	Consistent UI/UX	System shall maintain consistent design patterns, terminology, and navigation across all modules.
NFR29	Error Messages	System shall display clear, actionable error messages that guide users to solutions.
NFR30	Undo/Redo Functionality	System shall support undo/redo operations for timetable modifications (up to last 50 actions).
NFR31	Dark Mode Support	System should support optional dark mode for user preference.

3.3.6 Compatibility & Integration

ID	Requirement	Metric
NFR32	Browser Compatibility	System shall work on latest versions of modern browsers.

ID	Requirement	Metric
NFR33	Mobile Compatibility	System shall be accessible via mobile browsers and responsive design.
NFR34	Email Integration	System shall integrate with university email system for sending notifications via SMTP.
NFR35	Database Compatibility	System shall use MySQL or PostgreSQL (as per proposal) and shall support database migration between versions.

3.3.7 Maintainability & Support

ID	Requirement	Metric
NFR36	Code Documentation	Codebase shall be well-documented with clear comments.
NFR37	Logging & Monitoring	System shall generate detailed logs for debugging.
NFR38	Database Maintenance	System shall be available 99.9% of time during database maintenance operations.
NFR39	Version Control	System development shall use version control (Git) for all code changes.
NFR40	User Support	System shall include in-app help, FAQ, and support ticket system for user issues.

3.3.8 Compliance & Legal

ID	Requirement	Metric
NFR41	Data Privacy	System shall comply with GDPR/local data privacy regulations.
NFR42	Terms of Service	System shall display and implement terms of service and privacy policy for all users.
NFR43	Regulatory Compliance	System shall align with university policies and regulatory requirements.

3.4 Other Non-Functional Requirements

3.4.1 Business Rules

Rule ID	Business Rule
BR-1	The system shall allow only administrators to create or publish timetables.
BR-2	The system shall allow only faculty to request unavailability for their own schedules.
BR-3	The system shall prevent deletion of published timetables, only archiving is permitted. (Soft-delete only)
BR-4	The system shall restrict student users to view only their own schedules.
BR-5	The system shall prevent publishing timetables with unresolved Critical conflicts and shall display a list of such conflicts.
BR-6	The system shall prevent assigning an instructor to more than one course in the same time slot.
BR-7	The system shall prevent double-booking of venues for overlapping times.
BR-8	The system shall prevent course enrollment from exceeding the assigned venue capacity and shall issue a warning when enrollment exceeds 80% of capacity.
BR-9	The system shall require user accounts to be deactivated before deletion (Soft-delete only).
BR-10	The system shall maintain audit logs as append-only records that cannot be deleted.
BR-11	The system shall send timetable change notifications only for timetables in Published status.

Table 18: Business Rules

3.5 Other Requirements

3.5.1 Database Requirements

Req ID	Requirement	Specification
DBR-1	Database choice	The system shall use PostgreSQL version 18.x or higher or MySQL version 8.4.x or higher for the database system.
DBR-2	Data backup	The system shall perform database backups on a daily at 2:00 AM with a minimum retention period of 30 days.
DBR-3	Backup location	The system shall store backups on a separate server and shall implement an offsite backup for disaster recovery.
DBR-4	Database normalization	The database shall be normalized to at Third Normal Form (3NF) to reduce data redundancy and anomalies.

Table 19: Database Requirements

3.5.2 Legal & Compliance Requirements

Req ID	Requirement	Specification
LEG-1	Terms of Service	The system shall display the Terms of Service (ToS) and require user to accept it on first login.
LEG-2	Privacy Policy	The system shall display a privacy policy that describes data collection, and usage.
LEG-3	Intellectual Property	The ownership of the system shall be defined according to university intellectual property policy.
LEG-4	Open Source Compliance	The system shall comply with all open-source software license terms if applicable e.g. MIT.

Table 20: Legal & Compliance Requirements

3.5.3 Deployment & Installation Requirements

Req ID	Requirement	Specification
DEP-1	Installation manual	The system shall consist of a stepwise installation guide to install the system within the university servers.
DEP-2	Configuration guide	The system shall provide a guide on how the database, email configurations, and system settings are configured.
DEP-3	System requirements	The system shall record the minimum server requirements as well as the software requirements.
DEP-4	Initial data load	The system shall provide scripts to import initial data e.g. users, courses from Excel/CSV files.
DEP-5	Rollback procedure	This system shall have a documented rollback procedure to recover the past state of the system in case of an unsuccessful deployment.

Table 21: Deployment & Installation Requirements

3.5.4 Documentation & Training Requirements

Req ID	Requirement
DOC-1	System shall include comprehensive user documentation (NFR27).
DOC-2	System shall include administrator training materials and guides.
DOC-3	System shall include in-app help system and FAQ section (NFR40).
DOC-4	System documentation shall be delivered in both PDF and web-based formats.

Table 22: Documentation & Training Requirements

A Use Case Diagram

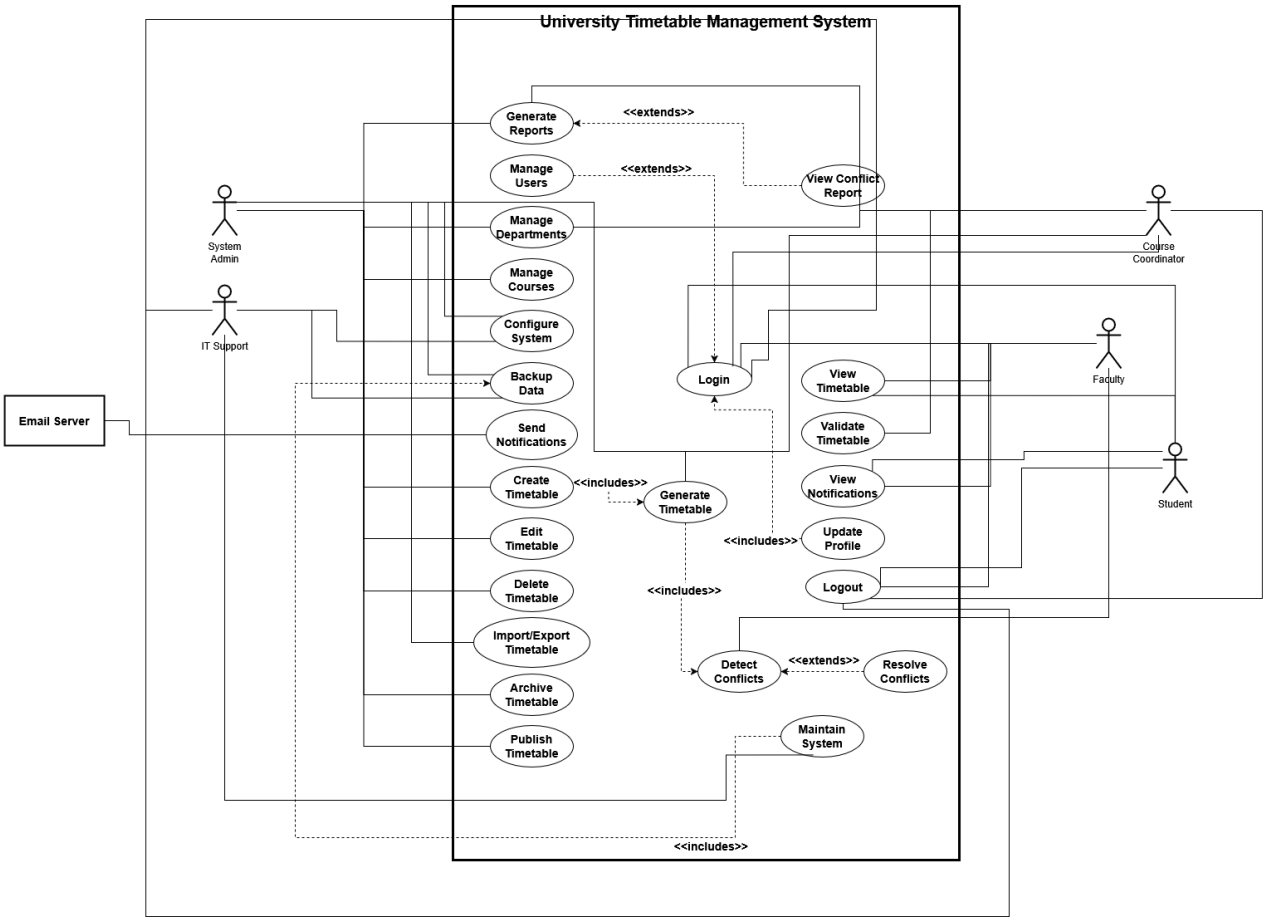

Figure 1: System Use Case Diagram

B Context Diagram

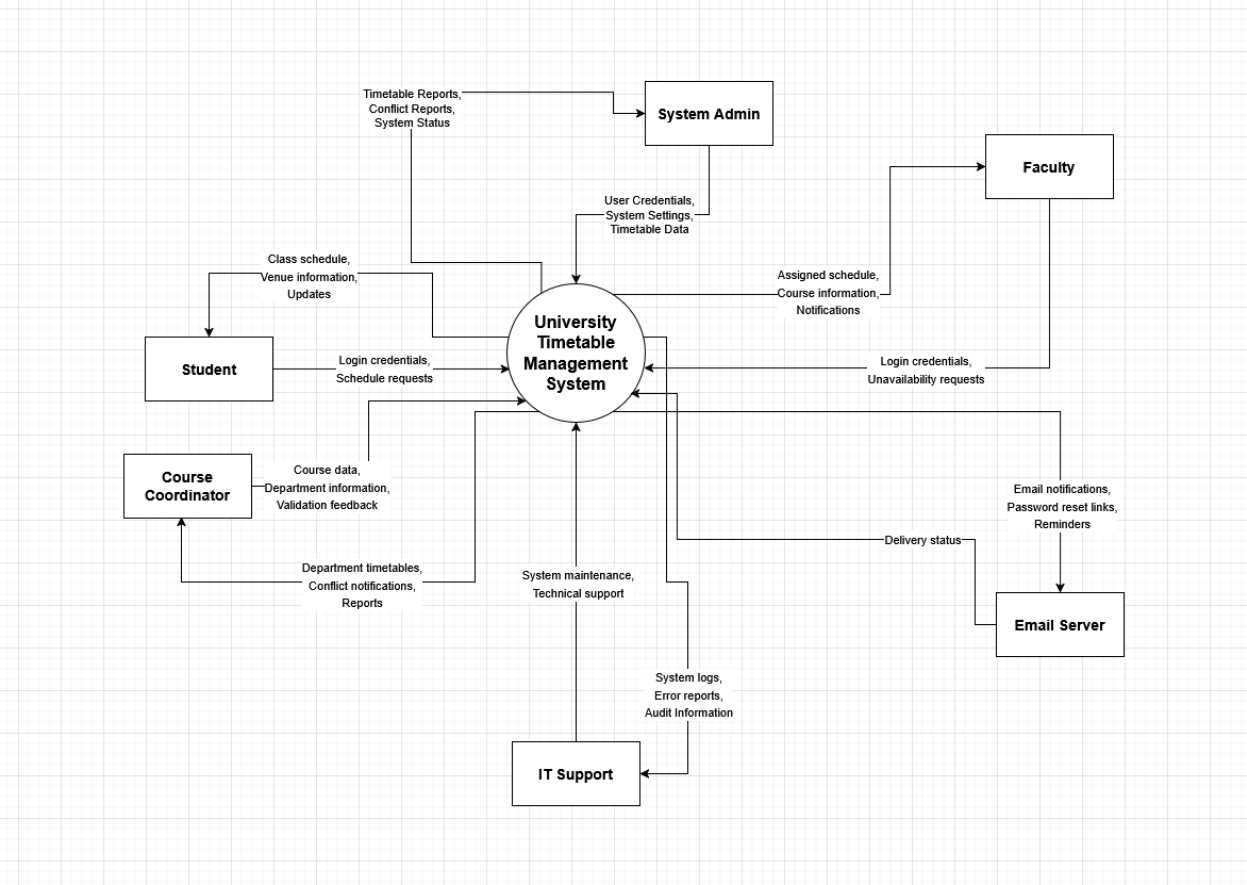

Figure 2: Context Diagram