Software Requirements Specification

for

TimeTable Managment

Version

1.0

Prepared by

Group Name: Class Align

P.Vivek	199	se22ucse199
M.Anish	168	se22ucse168
P.Abhinav	201	se22ucse201
N.Vivek	175	se22ucse175
M.Srikar	164	se22ucse165
K.Prakash	147	se22ucse147
P.Harsha	313	se22ucse313
P.Nitin	186	se22ucse186

Instructor: D.Vijay Rao

Course: Software Engineering(CS2301) Lab Section: <place your lab section here> Teaching Assistant: B.Murali Krishna

Date: 07-03-2025

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft v1.0	P. Vivek, M. Anish, P. Abhinav, N. Vivek, M. Srikar, K. Prakash, P. Harsha, P. Nitin	Initial draft of the Software Requirements Specification (SRS) for Timetable Management, outlining system functionalities, use cases, and constraints based on the project scope.	11-03-2025

1.1 Document Purpose

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements for the **Timetable Management System (TMS)**. The TMS is designed to automate and optimize timetable scheduling by efficiently allocating time slots for courses, instructors, and classrooms while ensuring conflict-free scheduling.

This document details the system's capabilities, constraints, and user interactions, serving as a guide for developers, testers, and stakeholders. The scope of this SRS covers the core functionalities of the system, including user data input, timetable generation using the **Backtracking Algorithm**, and data management using **MySQL**. The document focuses on the **frontend (HTML, CSS, JavaScript)** and **backend (MySQL, server-side logic)**, ensuring clarity for implementation and future enhancements.:

1.2 Product Scope

The **Timetable Management System (TMS)** is a web-based application designed to automate and streamline the process of scheduling timetables for educational institutions. It allows users to input essential details such as course information, instructor availability, room capacity, and time preferences. Using a **Backtracking Algorithm**, the system efficiently allocates time slots while ensuring that there are no scheduling conflicts.

The primary objective of TMS is to eliminate the inefficiencies of manual scheduling, reducing errors and saving time. By leveraging **HTML, CSS, and JavaScript** for the frontend and **MySQL** for backend data management, the system provides a user-friendly interface with robust performance. The key benefits include improved accuracy, enhanced productivity, and flexibility in handling schedule adjustments, ultimately leading to better resource utilization and time management.

1.3 Intended Audience and Document Overview

This document is intended for various stakeholders involved in the development and evaluation of the **Timetable Management System (TMS)**, including:

- **Developers** – To understand the functional and technical requirements for implementation.
- **Testers** – To verify that the system meets the specified requirements and functions correctly.
- **Clients (Institution Administrators, Professors, and Staff)** – To review the system's capabilities and ensure it aligns with their scheduling needs.
- **Professor (Evaluator/Instructor)** – To assess the project for academic and functional completeness.

Document Overview:

This Software Requirements Specification (SRS) document is structured as follows:

- **Section 1: Introduction** – Provides an overview of the system, including its purpose, scope, intended audience, and key functionalities.

- **Section 2: Overall Description** – Outlines the system's user characteristics, constraints, dependencies, and design considerations.
- **Section 3: Specific Requirements** – Details the functional and non-functional requirements, including use cases and system interactions.
- **Section 4: System Features** – Describes the core features, such as timetable generation, user inputs, and database management.

For optimal understanding, readers should begin with **Section 1 (Introduction)** for context, followed by **Section 2 (Overall Description)** to grasp the system's architecture and constraints. **Developers and testers** should focus on **Sections 3 and 4** to ensure the correct implementation of system functionalities.

1.4 Definitions, Acronyms, and Abbreviations

Below is a list of key terms, acronyms, and abbreviations used in this document:

- **API** – Application Programming Interface
- **CSS** – Cascading Style Sheets (used for frontend design)
- **DBMS** – Database Management System
- **Frontend** – The user interface of the system, developed using HTML, CSS, and JavaScript
- **HTML** – HyperText Markup Language (used for structuring web pages)
- **MySQL** – A relational database management system used for backend data storage
- **SRS** – Software Requirements Specification
- **TMS** – Timetable Management System (the software being developed)
- **UI** – User Interface
- **UX** – User Experience
- **Backend** – The server-side logic and database interactions of the system

1.5 Document Conventions

This document follows standard formatting and structuring conventions to ensure clarity and consistency. The conventions used are as follows:

1.5.1 Formatting Conventions

- The document follows the **IEEE SRS formatting guidelines**.
- **Font:** Arial, size **11 or 12**, used consistently throughout the document.
- **Line Spacing:** Single-spaced text with **1-inch margins** on all sides.
- **Section and Subsection Titles:** Bold and numbered according to the template (e.g., **1. Introduction, 1.1 Document Purpose**).
- **Emphasis:** *Italics* are used for comments or instructions.
- **Code or Commands:** If any code snippets are included, they are formatted in a monospaced font (e.g., SELECT * FROM timetable;).

1.5.2 Naming Conventions

- Database table names follow **snake_case** (e.g., course_schedule).
- Variables and function names in JavaScript follow **camelCase** (e.g., generateTimetable()).
- Acronyms such as **TMS (Timetable Management System)** are defined once and used consistently throughout the document.

By maintaining these conventions, the document remains professional, readable, and easy to reference for all stakeholders.

1.6 References and Acknowledgments

1.C.1 References

The following resources were used as references in the development of this SRS document and the **Timetable Management System (TMS):**

1. IEEE Standard for Software Requirements Specifications (IEEE 830- 1998)
2. MySQL Official Documentation – https://dev.mysql.com/doc/
3. HTML, CSS, and JavaScript Documentation – https://developer.mozilla.org/
4. Backtracking Algorithm Overview – https://www.geeksforgeeks.org/backtracking-algorithms/

1.C.2 *Acknowledgments*

We would like to acknowledge:

- **Professor [Your Professor's Name]** for their guidance and feedback throughout the project.
- **Team Members** for their collaboration in designing and implementing the system.
- **Online Resources and Documentation** that provided valuable insights into database management, algorithm implementation, and web development.

These references and contributions have played a crucial role in ensuring the successful development of the Timetable Management System.

2.1 Product Overview

The **Timetable Management System (TMS)** is a **new, self-contained** software solution designed to automate and optimize scheduling for educational institutions. The system eliminates the inefficiencies of manual timetable creation by allowing administrators to input course details, instructor availability, and room capacities. Using a **Backtracking Algorithm**, the system generates an optimized, conflict-free timetable, significantly reducing human error and saving time.

The TMS is designed as a **standalone web-based application** but can be integrated with **existing Learning Management Systems (LMS)** or **institutional databases** if needed. It provides a user-friendly interface for schedule creation and modification while ensuring real-time database updates. The system consists of the following major components:

- **User Interface (Frontend)** – Built with **HTML, CSS, and JavaScript**, allowing administrators and faculty members to input scheduling data and view generated timetables.
- **Database (Backend)** – Uses **MySQL** to store course schedules, faculty details, and room allocations securely.
- **Scheduling Engine** – Implements a **Backtracking Algorithm** to allocate time slots while resolving conflicts dynamically.
- **Admin Panel** – Allows authorized users to modify, update, and manage timetables efficiently.

Below is a high-level diagram illustrating how the system interacts with users and external components:

System Architecture Diagram

```
sql CopyEdit
+--------------------------------------
                                     +          +-------------------------------------+
|      User Interface            | -----> |          Scheduling Engine         |
|    (HTML,  CSS,  JS)       |              |  (Backtracking       Algo)     |
+--------------------------------------+          +-------------------------------------+
                 |                                                  |
                 v                                                  v
+--------------------------------------+          +-------------------------------------+
|            MySQL DB            | <-----> |            Admin Dashboard       |
| (Stores schedules)  |                        |  (Manage Timetables) |
+-------------------------------------+              +-------------------------------------+
```

This diagram provides a **high-level view** of how the system functions. Users interact with the **frontend**, which communicates with the **scheduling engine** to generate a timetable. The **MySQL database** stores all the necessary information, and an **admin panel** allows authorized users to manage schedules effectively.

The **Timetable Management System** ensures seamless timetable creation, minimizes conflicts, and improves the overall efficiency of scheduling in educational institutions.

2.2 Product Functionality

The **Timetable Management System (TMS)** provides the following major functions:

- **User Authentication s Role Management**
  - Allows **administrators** and **faculty members** to log in with secure credentials.
  - Different user roles (Admin, Faculty) with appropriate access levels.
- **Data Input s Management**
  - Users can **input course details, instructor availability, room capacities, and time preferences**.
  - Edit, update, or delete existing schedule data.
- **Automated Timetable Generation**
  - Uses a **Backtracking Algorithm** to efficiently allocate time slots while resolving conflicts dynamically.
  - Ensures that **no two courses overlap** in the same room or with the same instructor.
- **Manual Adjustments s Overrides**

    - Admins can manually adjust generated schedules if necessary.
    - System provides **conflict alerts** for manual changes that create overlaps.
- **Database Management**
    - Stores and retrieves schedule-related data using **MySQL**.
    - Ensures data consistency and prevents duplication.
- **User-Friendly Interface**
    - Built with **HTML, CSS, and JavaScript** for an intuitive experience.
    - Interactive timetable view for easy navigation and understanding.
- **Export s Print Functionality**
    - Users can export timetables in **PDF, Excel, or printable formats** for distribution.
- **Error Handling s Notifications**
    - Provides real-time **alerts for conflicts or missing data**.
    - Ensures smooth error handling to prevent crashes or incorrect scheduling.

These core functionalities make the **TMS an efficient and reliable tool for educational institutions**, streamlining the entire timetable scheduling process.

2.3 Design and Implementation Constraints

The development of the **Timetable Management System (TMS)** is subject to several design and implementation constraints, including technological requirements, performance limitations, and integration guidelines. These constraints ensure the system is developed efficiently while adhering to best practices.

2.3.1 Technological Constraints

- The system must be developed using **HTML, CSS, and JavaScript** for the frontend.

- The backend must utilize **MySQL** for database management.
- The scheduling logic must be implemented using a **Backtracking Algorithm** to ensure conflict-free timetables.

2.3.2 Hardware & Performance Constraints

- The system must be optimized for **web browsers**, ensuring responsiveness on desktops and tablets.
- Memory usage must be optimized to prevent excessive database queries that could slow down performance.
- The application should handle at least **100 concurrent users** without major performance degradation.

2.3.3 Security Constraints

- User authentication must be implemented to prevent **unauthorized access**.
- **Data validation and sanitization** must be performed to avoid SQL injection attacks.
- The system should ensure **role-based access control (RBAC)** to differentiate between admin and faculty permissions.

2.3.4 Integration & Compatibility Constraints

- The system must be compatible with **modern web browsers** such as Chrome, Firefox, and Edge.
- Future integration with **Learning Management Systems (LMS)** should be possible using **REST APIs**.
- The application must follow **responsive design principles** for usability across different screen sizes.

2.3.5 *Design Methodology Constraints*

- The **COMET Method** must be used for software design, ensuring **a structured and model-driven approach**.
    - **Reference:** Gomaa, H. (2011). *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures.* Cambridge University Press.
- The system design must use the **UML (Unified Modeling Language) modeling approach** for:
    - **Use Case Diagrams** to define user interactions.
    - **Class Diagrams** for system structure representation.
    - **Sequence Diagrams** to illustrate process flow.
    - **Reference:** Object Management Group (OMG), *UML 2.5 Specification*.

These constraints define the boundaries within which the **TMS** must be designed and developed, ensuring a **scalable, secure, and well-structured** software solution.

2.4 Assumptions and Dependencies

The development and deployment of the **Timetable Management System (TMS)** are based on several assumptions and dependencies. Any changes to these factors may impact the system's performance, functionality, or overall feasibility.

2.4.1 *Assumptions*

- **Stable Internet Connection** – The system is assumed to be used in an environment with a stable internet connection for seamless access and data synchronization.
- **Web-Based Access** – The application will primarily be accessed via modern web browsers such as **Google Chrome, Mozilla Firefox, and Microsoft Edge**.

- **Predefined User Roles** – It is assumed that the system will have **two main user roles**: **Administrators (who manage schedules) and Faculty Members (who view and suggest changes)**.
- **Consistent Data Entry** – The accuracy of generated timetables depends on **correct and complete input** from administrators (e.g., correct instructor availability and course details).
- **Limited Number of Concurrent Users** – The system is designed to handle **up to 100 concurrent users** at peak usage times without performance issues.

2.4.2 *Dependencies*

- **Technology Stack** – The system depends on **HTML, CSS, JavaScript (Frontend), and MySQL (Backend)** for development. Any changes in technology selection may affect implementation.
- **Hosting Environment** – The system requires a web server (e.g., **Apache, Nginx**) for deployment and access over the internet or an institutional intranet.
- **Security Protocols** – The system relies on **HTTPS encryption** and **SQL injection prevention techniques** to protect user data and maintain integrity.
- **Algorithm Performance** – The **Backtracking Algorithm** used for timetable generation is expected to work efficiently under normal data loads. If the dataset grows significantly (e.g., thousands of courses and faculty members), **alternative optimization techniques** may be required.

These **assumptions and dependencies** provide a foundation for system development, ensuring smooth implementation **as long as these conditions hold true**.

3.1 External Interface Requirements

3.1.1 User Interfaces

The **Timetable Management System (TMS)** provides a **web-based graphical user interface (GUI)** that allows administrators and faculty members to interact with the system efficiently. The interface is designed to be **intuitive, responsive, and user-friendly**, ensuring a seamless experience across different devices.

Main User Interface Components:

1. **Login Page** – Users authenticate with **username and password** to access their respective dashboards.
2. **Dashboard** – Displays an **overview of schedules, notifications, and options** for timetable management.
3. **Timetable Input Form** – Users can **enter course details, faculty availability, and room constraints** through dropdown menus and text fields.
4. **Generated Timetable View** – The system **automatically displays a conflict-free timetable** in a grid-based format.
5. **Manual Adjustment Panel** – Allows **drag-and-drop modifications** for administrators to refine the schedule.
6. **Export s Print Options** – Users can **export the timetable as PDF, Excel, or print it directly**.

User Interaction Features:

- **Navigation Menus** – A **sidebar or top menu** for quick access to different system sections.
- **Interactive Drag-and-Drop** – Allows users to **manually move** time slots if needed.
- **Color-Coded Timetable** – Helps users easily distinguish between different subjects, faculty members, and conflicts.

- **Search s Filters** – Users can **search for specific courses or faculty schedules** for quick modifications.

Basic User Interface Mockup

(*Note: A high-level mockup image will be included here if needed*)

Feature	Interaction Method
Login Page	Username C Password Input Fields
Navigation	Sidebar or Top Menu for Quick Access
Timetable Input	Form Fields, Dropdowns, Checkboxes
Auto-generated Timetable	Grid Display with Color Coding
Manual Adjustments	Drag-and-Drop, Click-to-Edit
Export C Print	Buttons for PDF, Excel, Print

The **TMS interface** is designed to be simple yet powerful, ensuring that users can efficiently **manage, modify, and generate timetables** with minimal effort.

3.1.2 Hardware Interfaces

The **Timetable Management System (TMS)** is a **web-based software** and primarily interacts with standard computing hardware. Since TMS does not rely on specialized hardware sensors, its hardware interface requirements are minimal. However, it must support interaction with the following devices:

Supported Hardware Interfaces:

1. **User Devices (Client-Side)**
   a. **Desktop Computers s Laptops** (Windows, macOS, Linux)

b. **Tablets** (iPads, Android Tablets) – For viewing and managing timetables on the go

c. **Smartphones** (Android, iOS) – Limited support for timetable viewing (responsive design)

2. **Server-Side Infrastructure**
   a. **Web Server** (Apache/Nginx) – Hosts the application and processes user requests
   b. **Database Server** (MySQL) – Stores all timetable-related data
3. **Printers s External Storage**
   a. **Printers** – Used to print schedules in hard copy format (supports standard A4 printing)
   b. **External Drives s Cloud Storage** – Users can export timetables as PDF/Excel and save them to USB drives or cloud platforms (Google Drive, OneDrive)

Nature of Data and Control Interactions:

- **Input Data** – User interactions via keyboard/mouse or touchscreen for schedule entry
- **Data Processing** – The backend processes inputs and generates optimized timetables
- **Output Display** – The system renders **timetables on screens, exports them as files, or sends them to a printer**

Since TMS is designed as a **cross-platform web application**, it does not have **direct hardware dependencies** beyond standard computing devices, ensuring **maximum compatibility and ease of access.**

3.1.3 Software Interfaces

The **Timetable Management System (TMS)** interacts with various software components to ensure seamless operation and data management. These interfaces facilitate data processing, user authentication, and system integration.

Primary Software Interfaces:

1. **Web Browser Interface**
    a. The system is accessed via modern web browsers such as **Google Chrome, Mozilla Firefox, and Microsoft Edge**.
    b. Uses **HTML, CSS, and JavaScript** for frontend rendering.
2. **Database Interface (MySQL)**
    a. TMS interacts with a **MySQL relational database** to store and retrieve timetable data.
    b. Queries are executed via **SQL statements**, ensuring data integrity and efficient retrieval.
3. **Backend Communication**
    a. The backend handles scheduling logic, conflict detection, and manual overrides.
    b. Interacts with the frontend through **AJAX requests or REST APIs** for real-time updates.
4. **Export s Printing Software**
    a. Users can **export schedules in PDF and Excel formats** for external use.
    b. The system integrates with built-in OS printing services to allow **direct printing**.
5. **Future Mobile App Interface (Planned)**
    a. A future **mobile application** may be developed for faculty members and students to view timetables.
    b. The mobile app will interact with the system through a **REST API**, enabling users to:
        i. View real-time schedules.
        ii. Receive notifications for changes or cancellations.
        iii. Send timetable change requests (if applicable).

By maintaining **modular and well-structured software interfaces**, the TMS ensures **smooth interoperability** across different platforms and potential future extensions, such as **mobile apps or LMS integrations**.7

3.2 Functional Requirements

Functional requirements define the specific behaviors and capabilities that the **Timetable Management System (TMS)** must provide. These requirements ensure the system operates effectively and meets user expectations.

3.2.1 F1: User Authentication and Access Control

- The system shall **require users to log in** with a **username and password** before accessing features.
- The system shall **support multiple user roles**: **Administrators (full access) and Faculty Members (restricted access).**
- The system shall encrypt passwords and **ensure secure authentication mechanisms**.

3.2.2 F2: Course and Faculty Data Management

- The system shall allow administrators to **add, edit, and delete course details**, including course name, instructor, room number, and duration.
- The system shall enable faculty members to **input their availability** and **request schedule changes** if needed.

3.2.3 F3: Automatic Timetable Generation

- The system shall use a **Backtracking Algorithm** to generate a conflict-free timetable.
- The system shall **automatically allocate classrooms** based on availability and capacity constraints.
- The system shall provide **real-time conflict detection and resolution suggestions**.

3.2.4 F4: Manual Schedule Adjustment

- The system shall provide an **interactive drag-and-drop interface** for administrators to manually adjust schedules.
- The system shall allow administrators to **override auto-generated timetables** when necessary.

3.2.5 F5: Timetable Display and Accessibility

- The system shall display timetables in a **grid format**, categorized by faculty, course, and time slots.
- The system shall allow users to **search for specific schedules** using filters such as faculty name, course, or time slot.

3.2.C FC: Exporting and Printing

- The system shall allow users to **export timetables in PDF and Excel formats**.
- The system shall provide a **print-friendly view** for hard copy generation.

3.2.7 F7: Notifications and Alerts

- The system shall send **notifications to faculty members** regarding schedule updates or changes.
- The system shall provide alerts for **conflicting schedules, room availability issues, or missing inputs**.

3.2.8 F8: Data Backup and Recovery

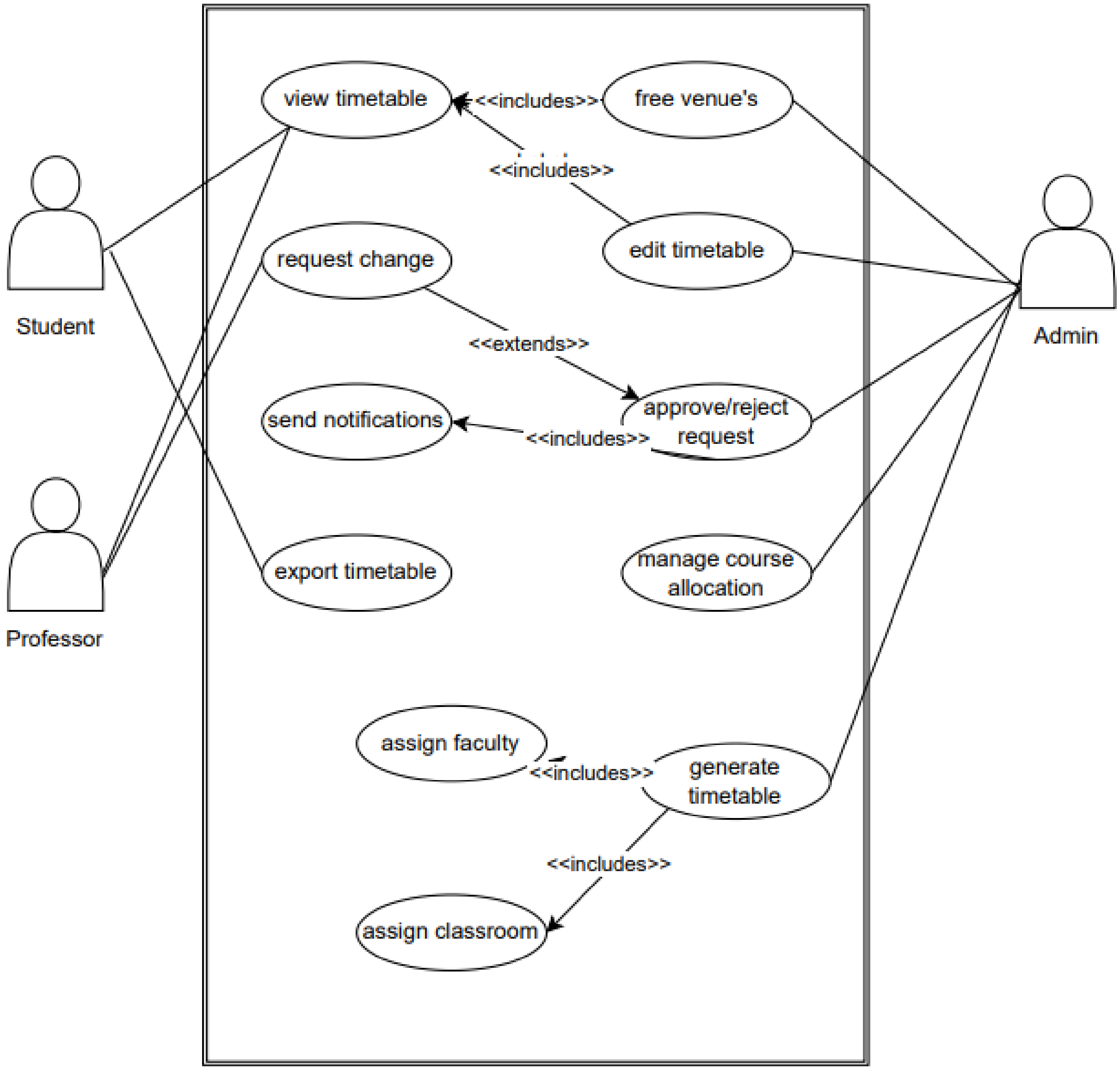
- The system shall **automatically back up timetable data** periodically.
- The system shall support **data recovery in case of accidental deletions or system failures**.

By implementing these **functional requirements**, the TMS ensures **efficient scheduling, conflict resolution, and user-friendly interactions**, making timetable management seamless for institutions.

3.3 Use Case Model

The **Use Case Diagram** provides an overview of the interactions between different actors and the **Timetable Management System (TMS)**.

Use Case Diagram



Actors in the System

- **Administrator** – Manages timetables, faculty schedules, and system settings.
- **Faculty Member** – Views assigned schedules and submits availability.
- **System Database** – Stores timetable information and processes scheduling data.

3.3.1 Use Case #1: Login s Authentication (U1)

Author

[Your Name]

Purpose

The objective of this use case is to ensure **secure access control**, allowing only **authorized users** to access the system.

Requirements Traceability

- **F1: User Authentication and Access**

Control Priority

High – Essential for system security.

Preconditions

- The user must have a valid **username and password**.
- The system must be **connected to the authentication database**.

Postconditions

- Successful login grants access to the user's dashboard.
- Failed login attempts may result in **temporary account lockout**.

Actors

- **Administrator**
- **Faculty**

Member Flow of

Events

1. **Basic Flow**
   a. User navigates to the **Login Page**.
   b. User enters **username and password**.
   c. System validates credentials against the database.
   d. If valid, the system redirects the user to their respective **dashboard**.
2. **Alternative Flow**
   a. If incorrect credentials are entered, the system displays an **error message** and allows the user to retry.
3. **Exceptions**

a. If **3 failed attempts** occur, the system **locks the account for 5 minutes**.

Notes/Issues

- Implement **password encryption** for security.
- Consider **multi-factor authentication (MFA)** for better protection.

3.3.2 Use Case #2: Generate Timetable (U2)

Author

[Your Name]

Purpose

To automatically generate a **conflict-free timetable** based on faculty availability, course details, and room constraints.

Requirements Traceability

- **F3: Automatic Timetable Generation**

Priority

High – Core functionality of the system.

Preconditions

- **Course details**, **faculty availability**, and **room data** must be entered.

Postconditions

- A **finalized, conflict-free timetable** is generated and stored in the database.

Actors

- **Administrator**
- **System**

Database Flow of

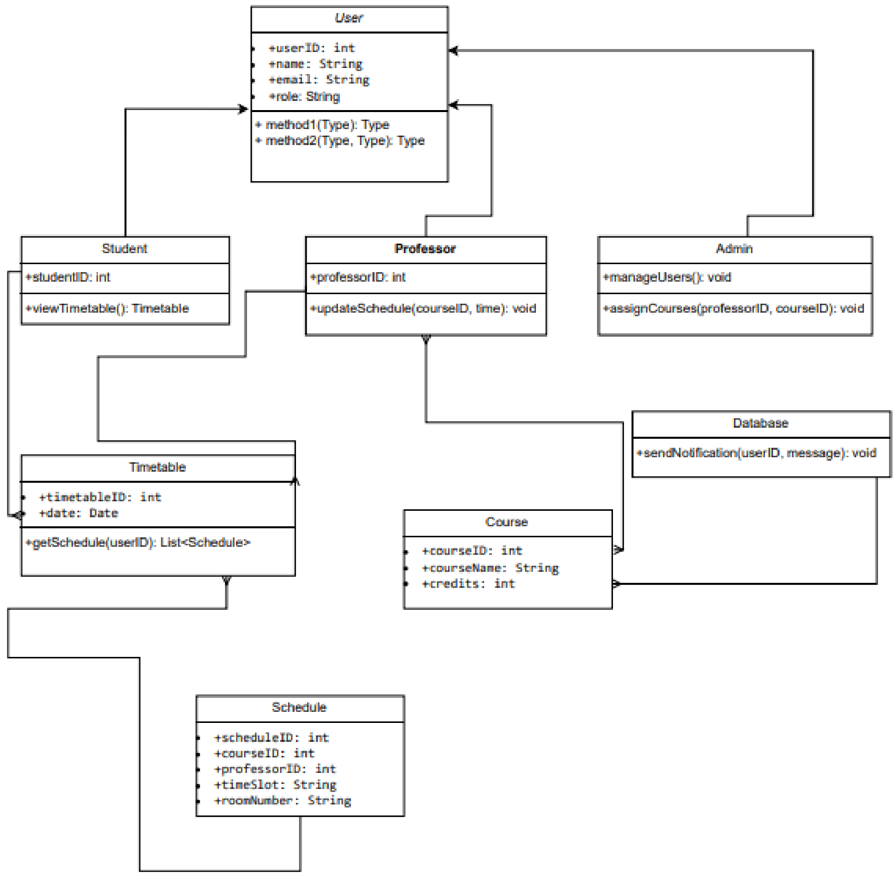
Events

1. **Basic Flow**
    a. Administrator selects **"Generate Timetable"** from the dashboard.
    b. The system retrieves all necessary data.
    c. The system applies the **Backtracking Algorithm** to create an optimized schedule.
    d. The generated timetable is displayed for review.
2. **Alternative Flow**
    a. If conflicts are detected, the system **suggests manual adjustments**.
3. **Exceptions**
    a. If faculty availability data is missing, the system prompts an **error message**.

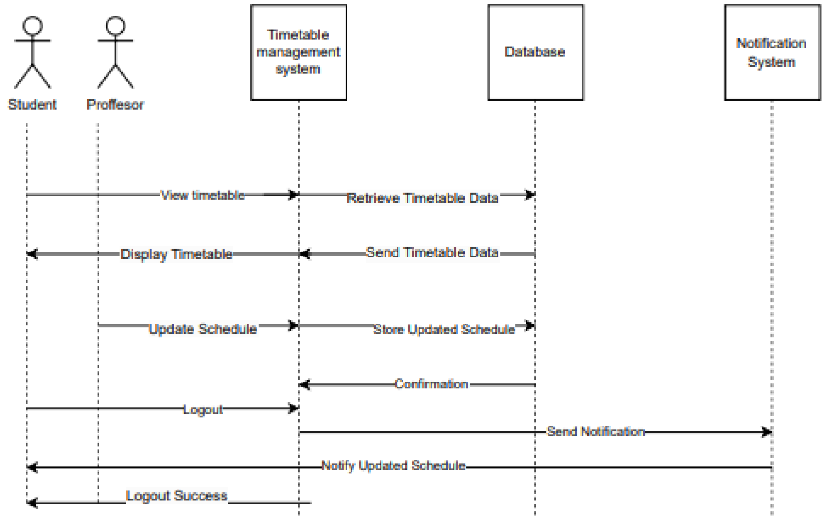
Notes/Issues

- Ensure **real-time conflict detection**.
- Optimize the algorithm for **faster timetable generation**.

USER DIAGRAM



SEQUENCE DIAGRAM



4 Other Non-functional Requirements

4.1 Performance Requirements

The **Timetable Management System (TMS)** must meet the following **performance criteria** to ensure **efficient operation** under varying loads and conditions.

- **P1.** The system shall **generate a complete timetable within 5 seconds** of receiving all input data.
- **P2.** The system shall support **at least 50 concurrent users** (faculty members and administrators) without significant performance degradation.
- **P3.** The timetable display shall **load within 2 seconds** when accessed by users.
- **P4.** The system shall **process updates to faculty availability** and reflect changes in the timetable **within 3 seconds**.
- **P5.** The system shall handle **at least 1,000 course entries and 500 faculty members** without a decline in responsiveness.
- **P6.** Database queries for retrieving schedules shall **execute within 1 second** under normal conditions.
- **P7.** The system shall send notifications for schedule changes **within 1 second** of an update.

4.2 Safety and Security Requirements

The **Timetable Management System (TMS)** must incorporate safety and security measures to **protect user data, prevent unauthorized access, and ensure system reliability**.

4.2.1 Data Security & Privacy

- **S1.** All **user credentials and sensitive data** shall be **encrypted** using **AES-256 encryption** to prevent unauthorized access.

- **S2.** The system shall enforce **role-based access control (RBAC)**, restricting users to only the information and functionalities relevant to their roles (e.g., administrators can modify schedules, faculty can only view their assigned timetables).
- **S3.** User **passwords must follow a strong password policy** (minimum 8 characters, at least one uppercase letter, one number, and one special character).
- **S4.** The system shall **automatically log out inactive users after 15 minutes** of inactivity.
- **S5.** The system shall support **multi-factor authentication (MFA)** for administrator accounts to enhance security.

4.2.2 System Security

- **S6.** All **database transactions must be logged**, and unauthorized access attempts shall trigger alerts to the system administrator.
- **S7.** The system shall be protected against **SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF)** attacks.
- **S8.** All API communications (if applicable) between the front end and back end shall be secured using **HTTPS (SSL/TLS encryption)**.
- **SG.** The system shall include **automatic backups** of timetable data **every 24 hours** to prevent data loss.

4.2.3 Safety Considerations

- **S10.** The system must **prevent schedule conflicts** by enforcing logical constraints (e.g., ensuring no two classes are scheduled in the same room at the same time).
- **S11.** If a critical failure occurs, the system shall **enter safe mode**, allowing only read-only access to the last stable timetable version.
- **S12.** The system shall comply with **educational data privacy regulations**, such as **FERPA (for the US) or GDPR (for Europe)**, to ensure student and faculty data protection.

Appendix A – Data Dictionary

Name	**Type**	**Description**	**Possible Values/Format**	**Related Operations/Requirements**
User_ID	String	Unique identifier for each user	University Email (e.g., se22ucse175@mu.edu)	User Registration & Authentication (F1)
Username	String	User's chosen name	Alphanumeric (max 20 chars)	User Registration & Authentication (F1)
Email	String	User's email address (university-verified)	Valid email format	User Registration (F1)
Password	String	Encrypted password for authentication	Hashed string	User Authentication (F1)
User_Role	Enum	Defines user type	Admin, Faculty, Student	User Access Control (F1)
Course_ID	Integer	Unique identifier for a course	Auto-incremented	Course Management (F2)
Course_Name	String	Name of the course	Alphanumeric (max 50 chars)	Course Management (F2)
Faculty_ID	Integer	Unique identifier for a faculty member	Auto-incremented	Faculty Management (F2)
Faculty_Name	String	Name of the faculty member	Alphanumeric (max 50 chars)	Faculty Management (F2)
Schedule_ID	Integer	Unique identifier for a schedule entry	Auto-incremented	Schedule Management (F3)
Schedule_Tim	Timesta	Assigned	Date-Time format	Schedule

Name	Type	Description	Possible Values/Format	Related Operations/Requirements
e	mp	time slot		Management (F3)
Room_ID	Integer	Unique identifier for a classroom	Auto-incremented	Room Management (F4)
Room_Name	String	Name of the classroom	Alphanumeric (max 20 chars)	Room Management (F4)
Room_Capacity	Integer	Maximum number of students allowed	Numeric	Room Management (F4)
Timetable_ID	Integer	Unique identifier for a generated timetable	Auto-incremented	Timetable Generation (F5)
Timetable_Status	Enum	Status of the timetable	Active, Inactive, Modified	Timetable Management (F5)
Notification_ID	Integer	Unique identifier for a notification	Auto-incremented	Notifications & Alerts (F6)
Notification_Type	Enum	Type of notification	Schedule Update, Conflict Alert	Notifications & Alerts (F6)
Export_Format	Enum	Format in which timetable is exported	PDF, Excel, Print	Exporting & Printing (F7)

Appendix B – Group Log

Date	Attendees	Discussion Topics	Time Spent	Assigned Tasks
06-03-25	P. Vivek, M. Anish, P. Abhinav, N. Vivek, M. Srikar, K. Prakash, P.	SRS sections assigned, formatting, content finalization	120-150 min	1.1-1.3 : K. Prakash 1.4-1.6 : M. Srikar

Date	Attendees	Discussion Topics	Time Spent	Assigned Tasks
	Harsha, P. Nitin			2.1-2.2 : N. Vivek 2.3-2.4 : P. Harsha 3.1-3.2 : P. Nitin
08-03-25	P. Vivek, M. Anish, P. Abhinav, N. Vivek, M. Srikar, K. Prakash, P. Harsha, P. Nitin	Review of functional requirements, data dictionary	140-160 min	Finalizing system design details
10-03-25	P. Vivek, M. Anish, P. Abhinav, N. Vivek, M. Srikar, K. Prakash, P. Harsha, P. Nitin	Final proofreading and submission	130-150 min	Last-minute refinements before submission