

Team Members

- Sarthak Kulkarni – 23101B0019
- Ambar Petkar – 23101B0010
- Pulkit Saini – 23101B0021

SEWDL Experiment - 2

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) details the requirements for Timetable Buddy: Lecture & Classroom Scheduling System. The product enables digital seat reservation, lecture scheduling, and classroom management, supporting students, teachers, and administrators. The SRS covers the system as a whole for the initial release.

1.2 Document Conventions

Requirement statements are tagged and numbered for clarity. All role-based terms (e.g., Student, Teacher, Admin) are capitalized for consistency. Priority levels High/Medium/Low are indicated where applicable.

1.3 Intended Audience and Reading Suggestions

This SRS is intended for the development team, testers, project managers, academic administrators, and institutional stakeholders. For an overview, begin with Sections 1 and 2, then proceed to System Features and Nonfunctional Requirements for detailed guidance.

1.4 Product Scope

Timetable Buddy provides a web-based solution for managing lecture schedules, classroom bookings, and seat reservations. The platform facilitates efficient allocation of resources, prevents overcrowding, and improves administrative visibility, aligning with modern digitalization strategies in educational institutions.

1.5 References

- IEEE SRS Template, Wiegers 1999
-

2. Overall Description

2.1 Product Perspective

Timetable Buddy is a new, self-contained software system designed to replace manual and unstructured attendance and scheduling processes in academic institutions. It draws conceptual inspiration from event booking platforms integrated with institutional-specific requirements. The software is web-based, intended to operate independently or alongside existing administrative systems.

2.2 Product Functions

- Lecture Viewing and Booking: **Students can view scheduled lectures and reserve seats in advance.**
- Schedule and Classroom Management: **Teachers can manage lecture schedules and assign classrooms.**
- Role-Based Access: **System access is differentiated for Students, Teachers, and Admins.**
- Occupancy Tracking: **Classroom occupancy monitoring avoids booking clashes and overcapacity.**
- Reporting: **Generation of usage and attendance reports for administrators.**
- Admin Tools: **Admin dashboard for managing users, classrooms, and schedules.**

2.3 User Classes and Characteristics

- Students: **Users who view timetables and reserve lecture seats, requiring basic web literacy.**
- Teachers: **Manage classes, create or modify lecture events, require moderate system familiarity.**
- Admins/Academic Admins: **Oversee class scheduling, manage users and resources; expected to have advanced administrative rights.**
- Institutional Stakeholders: **Require access to high-level usage and trend data for resource planning.**

2.4 Operating Environment

- Hardware: **Standard computer or laptop with internet access.**
- OS: **Windows, Linux, or macOS.**
- Browser: **Chrome, Firefox, or Safari (latest stable versions).**
- Database: **SQLite for development; PostgreSQL for production.**
- Framework: **Django (Python-based).**

2.5 Design and Implementation Constraints

- **Must use Django for backend development.**
- **Deployment intended on cloud platforms such as PythonAnywhere, Render, or Heroku.**
- **No integration with hardware-based attendance (e.g., biometrics).**
- **Web-based only; mobile implementation is explicitly out-of-scope for v1.0.**

2.6 User Documentation

- **Online User Manual HTML/PDF**
- **Quick Start Guide**
- **FAQ and Troubleshooting Section within the web portal**

2.7 Assumptions and Dependencies

- **All users have institutional credentials for authentication.**
- **Network connectivity is reliable within campus premises.**
- **The institution provides classroom inventory and lecture schedules for system initialization.**

3. External Interface Requirements

3.1 User Interfaces

- **Responsive web interface for Students, Teachers, and Admins.**
- **Lecture list views, interactive seat maps for selection, booking confirmation dialogs.**
- **Admin dashboard for classroom and schedule management.**
- **Standard navigation bar, consistent form controls, error message displays.**

3.2 Hardware Interfaces

- **None, as the web system does not directly interface with hardware such as card readers or biometric systems.**

3.3 Software Interfaces

- **Database: SQLite (development), PostgreSQL (production).**
- **Django Auth System for user authentication.**
- **Potential for integration with institutional SSO (Single-Sign-On) in future versions.**

3.4 Communications Interfaces

- **HTTP/HTTPS protocols for all web traffic.**
- **SSL encryption enforced for all authenticated sessions.**

4. System Features

4.1 Lecture Viewing & Seat Reservation

4.1.1 Description and Priority

Enables students to browse lectures and reserve seats online. Priority: High

4.1.2 Stimulus/Response Sequences

- **Student logs in**
- **Student views lecture list**
- **Student selects lecture, views available seats**
- **Student chooses seat and confirms booking**
- **System updates database and provides confirmation**

4.1.3 Functional Requirements

- RE Q 1 **System must display upcoming lectures with seat availability.**
- REQ 2 **System must allow authenticated students to reserve a seat for a lecture.**
- RE Q 3 **System must not allow overbooking beyond room capacity.**

4.2 Lecture and Classroom Management (Teacher/Admin)

4.2.1 Description and Priority

Teachers and Admins can add, update, and remove lectures, assign classrooms. Priority: High

4.2.2 Stimulus/Response Sequences

- **Teacher/Admin logs in**
- **Views dashboard**
- **Schedules new lecture or edits existing**
- **System validates availability and confirms changes**

4.2.3 Functional Requirements

- RE Q 4 **Teachers must have access only to assigned courses.**
- REQ 5 **Admins can manage all lectures, classrooms, and user accounts.**

4.3 Role-Based Access

4.3.1 Description and Priority

Ensures only authorized users access specific modules. Priority: High

4.3.2 Functional Requirements

- RE Q 6 **System enforces authentication and role-based permissions at login.**

4.4 Reporting

4.4.1 Description and Priority

Generates reports on room utilization and booking history. Priority: Medium

4.4.2 Functional Requirements

- RE Q 7 **Admins can export classroom and participation statistics.**

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Seat booking confirmation within 2 seconds under typical load.**
- **System to support concurrent logins by at least 200 users.**

5.2 Safety Requirements

- **No physical or operational safety hazards related to software use.**

5.3 Security Requirements

- **All personal data is encrypted in transit.**
- **User passwords are securely hashed and never stored in plaintext.**
- **Session timeouts after inactivity of 30 minutes.**

5.4 Software Quality Attributes

- **Usability: Intuitive UI with online help resources.**
- **Reliability: System uptime target of 98% during academic terms.**
- **Maintainability: Modular codebase following Django best practices.**

5.5 Business Rules

- **Only students officially enrolled in a course can reserve seats in its lectures.**
- **Lectures are limited to classroom capacity as defined during scheduling.**
- **Only admins can create or deactivate user accounts.**

6. Other Requirements

- All screens must be accessible (WCAG 2.1 AA compliant).
- All system logs must be retained for a minimum of 90 days.

Appendix A Glossary

- **Lecture**: A scheduled class session.
- **Classroom**: Physical teaching space with defined capacity.
- **Admin**: User with rights over all scheduling, users, and system configurations.

Team Members

- Sarthak Kulkarni – 23101B0019
- Ambar Petkar – 23101B0010
- Pulkit Saini – 23101B0021

—