



North South University

Department of Electrical & Computer Engineering

**Software Requirements Specification
for
Intelligent Task Planner for Students**

Submitted to:

Dr. Mohammad Rezwanul Huq(MRH1)

Associate Professor

Group No. - 06

Group Members:

| ID | Name | Contribution |
|-------------------|----------------------------|-------------------------|
| 2312561642 | Pujon Chakraborty | SRS |
| 2212650042 | Ifrat Afroze Niniya | SRS |
| 2222756042 | Sudipto Roy | UI Mockup |
| 2311539042 | Arafat Zaman Ratul | Use Case Diagram |

Contents

| | | |
|-----|---|---|
| 1 | Introduction | 2 |
| 1.1 | Purpose | 2 |
| 1.2 | Document Conventions | 2 |
| 1.3 | Project Scope | 2 |
| 1.4 | References | 2 |
| 2 | Overall Description | 2 |
| 2.1 | Product Perspective | 2 |
| 2.2 | Product Features | 2 |
| 2.3 | User Classes and Characteristics | 3 |
| 2.4 | Operating Environment | 3 |
| 2.5 | Design and Implementation Constraints | 3 |
| 2.6 | Assumptions and Dependencies | 3 |
| 2.7 | Stakeholders and Expected Benefits | 3 |
| 3 | System Features and Requirements | 4 |
| 3.1 | Functional Requirements | 4 |
| 3.2 | System Features | 5 |
| 3.3 | Non-Functional Requirements | 6 |
| 4 | System Use-Case Diagram | 7 |
| 5 | System Use-Case Diagram | 8 |

1. Introduction

1.1 Purpose

This SRS describes the functional and non-functional requirements for the Intelligent Task Planner for Students, a web-based productivity tool to assist university students in managing their tasks, assignments, exams, and study schedules.

1.2 Document Conventions

- Bold section headers.
- Requirements are labeled as REQ-x.

1.3 Project Scope

The project aims to create a scheduling assistant that utilizes rule-based or AI-driven logic to recommend optimal plans based on priority, deadlines, and user availability. It will provide visual analytics, reminders, and an intuitive UI.

1.4 Intended Audience and Reading Suggestions

This document is intended for developers, project supervisors, testers, and stakeholders. Readers should first review the Introduction and Overall Description before moving on to detailed features and requirements.

1.5 References

- TickTick: https://ticktick.com/?language=en_US
- IEEE SRS Standard

2. Overall Description

2.1 Product Perspective

The Intelligent Task Planner for Students is an independent web-based system. It shares conceptual similarities with productivity tools like and adds academic-specific features and intelligent time allocation logic.

2.2 Product Features

- Task creation and prioritization
- Smart scheduling recommendations
- Visual dashboard (calendar, charts)
- Notifications and reminders
- Pomodoro timer and Google Calendar sync

2.3 User Classes and Characteristics

- Students: Primary users
- Guest User

2.4 Operating Environment

- Web browser (Chrome, Firefox, Brave etc.)
- Backend: Node.js / FastAPI
- Database: PostgreSQL

2.5 Design and Implementation Constraints

- Must use relational DBMS
- Optional AI scheduler must be modular
- Web technologies only

2.6 Assumptions and Dependencies

It is assumed that the Intelligent Task Planner for Students will be developed as a web-based application using standard technologies such as HTML, Tailwind CSS, and JavaScript for the frontend, with either Node.js or FastAPI as the backend framework. The backend will be connected to a relational database system such as MongoDB or PostgreSQL to store user data, tasks, schedules, and preferences. The application requires users to have access to a stable internet connection to ensure smooth interaction with the system, particularly for features such as schedule generation and notification delivery. It is also assumed that users will provide accurate inputs regarding their tasks and available time blocks, as the scheduling logic depends on this data. The initial version of the scheduler will follow rule-based logic; however, there may be optional enhancements using AI techniques, which will depend on the availability of relevant libraries and training data. Integration with external APIs, such as Google Calendar, is optional and subject to proper API access and authentication credentials. Additionally, since the web application is designed primarily for student use, it is expected to handle moderate-scale user loads without the need for enterprise-grade infrastructure.

2.7 Stakeholders and Expected Benefits

- Students (Primary Users):
 - Role: Input tasks, deadlines, priorities, and available time; receive and adjust optimized schedules; access reminders and optional analytics.
 - Benefits: Improved task organization, reduced stress, and enhanced productivity through personalized schedules, timely notifications, and insights into time management efficiency.
- System Scheduler (AI/Rule-Based Engine):
 - Role: Processes user inputs to generate optimized task schedules and adapts to manual changes.
 - Benefits: Automates scheduling to deliver efficient, tailored plans, minimizing user effort and improving task completion rates.

3. System Features and Requirements

3.1 Functional Requirements

3.1 User Authentication

Description: Register new users and allow secure login/logout.

Priority: High

- REQ-1: The user shall be able to register and log in securely using a unique email and password.

3.2 Task Management

Description: Add/edit/delete tasks with deadlines, priority, and tags.

Priority: High

- REQ-2: The user shall be able to create, edit, and delete tasks with attributes such as title, description, deadline, and priority.

3.3 Time Block Management

Description: Define user availability to assist the scheduling engine.

Priority: High

- REQ-3: The user shall be able to specify available time blocks for scheduling tasks.

3.4 Scheduler Engine

Description: Suggests an optimized schedule based on task data and time availability.

Priority: High

- REQ-4: The system shall generate a recommended task schedule using rule-based or AI-assisted logic.

3.5 Schedule Adjustment

Description: Enable manual modification of generated schedules.

Priority: Medium

- REQ-5: The user shall be able to accept, adjust, or override the system-generated schedule.

3.6 Reminders and Notifications

Description: Notify users about upcoming tasks.

Priority: Medium

- REQ-6: The system shall send reminders 15 minutes before scheduled tasks.

3.7 Visualization Dashboard

Description: Display task distribution and time usage analytics.

Priority: Low

- REQ-7: The user shall be able to view daily and weekly visualizations such as calendars and charts.

3.2 System Features

Authentication

- Secure login/logout
- Session management for active users

Task Manager

- CRUD operations on tasks
- Task fields: Title, Description, Deadline, Priority, Category

Scheduler Engine

- Generates a daily/weekly schedule
- Respects available time, deadlines, and task priorities
- Allows manual edits by the user

Reminder System

- Configurable task notifications
- Browser/email notifications

Pomodoro Timer

- Timed intervals for task execution
- Visual indicator and logging for completed Pomodoros

Analytics Dashboard

- Pie chart of time spent on tasks
- Weekly trend line for focus/productivity

3.3 External Interface Requirements

User Interfaces

- Responsive web interface
- Calendar view, task input form, analytics dashboard

Hardware Interfaces

- Not applicable (client-side browser)

Software Interfaces

- Google Calendar API (optional)
- Database: MySQL/PostgreSQL

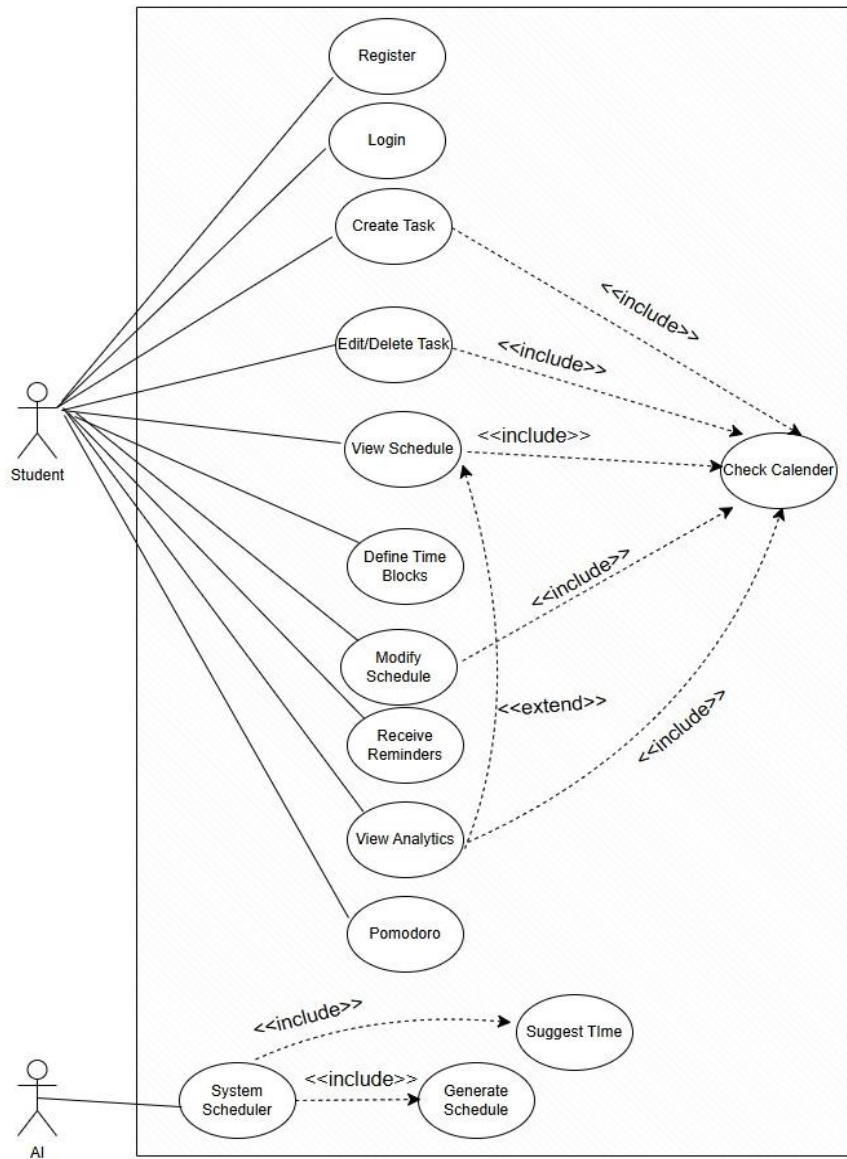
Communications Interfaces

- HTTP/HTTPS for client-server communication

3.3 Non-Functional Requirements

- **Security:** Password hashing, access control
- **Performance:** Task schedule generated under 2 seconds
- **Usability:** Simple, student-friendly UI
- **Scalability:** Must support multiple users concurrently
- **Reliability:** Auto-save and error recovery mechanisms
- **Maintainability:** Modular design for easy update
- **Compatibility:** Support multiple devices and browsers

4. System Use-Case Diagram



5. System Interface Mockups

