



# Software Requirements Specification

## Student Productivity Platform

Version 1.0

### **Team Members:**

Hadia Yasir - NUM-BSCS-2024-23

Muhammad Bilal - NUM-BSCS-2024-47

Waseem Ullah - NUM-BSCS-2024-78

### **Client Representatives:**

Shazia Nazir & Maryam Noor ul Ain

**Course:** Software Engineering

**Submitted To:** Department of Computer Science

**Semester:** Fall 2024

December 28, 2025

Department of Computer Science  
Namal University Mianwali

*Prepared in accordance with IEEE Standard 830-1984*

## Revision History

---

Version	Date	Description	Author(s)
1.0	December 28, 2025	Initial draft	Hadia, Bilal, Waseem

Table 1: Document Revision History

# Contents

<b>Revision History</b>	<b>2</b>
<b>List of Figures</b>	<b>4</b>
<b>List of Tables</b>	<b>5</b>
<b>1 Introduction</b>	<b>6</b>
1.1 Purpose . . . . .	6
1.2 Scope . . . . .	6
1.3 Definitions, Acronyms, and Abbreviations . . . . .	8
1.4 References . . . . .	8
1.5 Overview . . . . .	8
<b>2 Overall Description</b>	<b>10</b>
2.1 Product Perspective . . . . .	10
2.2 Product Functions . . . . .	10
2.3 User Characteristics . . . . .	11
2.4 Operating Environment . . . . .	11
2.5 Constraints . . . . .	12
2.6 Assumptions and Dependencies . . . . .	13
2.7 Apportioning of Requirements . . . . .	13
<b>3 Specific Requirements</b>	<b>14</b>
3.1 External Interfaces . . . . .	14
3.1.1 User Interfaces . . . . .	14
3.1.2 Hardware Interfaces . . . . .	14
3.1.3 Software Interfaces . . . . .	15
3.1.4 Communications Interfaces . . . . .	15
3.2 Functional Requirements . . . . .	16
3.2.1 Centralized Dashboard Module . . . . .	16
3.2.2 Gamification Engine . . . . .	18
3.3 Non-Functional Requirements . . . . .	19
3.3.1 Performance Requirements . . . . .	19
3.3.2 Security Requirements . . . . .	20
3.3.3 Usability Requirements . . . . .	20
<b>A Context Diagram</b>	<b>22</b>
<b>B Use Case Diagram</b>	<b>23</b>
<b>C Glossary</b>	<b>23</b>

## List of Figures

1	Enter Caption . . . . .	22
2	System Context Diagram showing system boundaries and external entities	22
3	Use Case Diagram showing system actors and their interactions . . . . .	23

## List of Tables

1	Document Revision History . . . . .	2
2	Definitions, Acronyms, and Abbreviations . . . . .	8
3	User Characteristics . . . . .	11
4	Future Requirements . . . . .	13
5	Software Interfaces . . . . .	15

# 1 Introduction

---

This document provides a comprehensive Software Requirements Specification (SRS) for the Student Productivity Platform. It describes the functional and non-functional requirements, constraints, and assumptions that will guide the design, development, and testing of the system.

## 1.1 Purpose

This Software Requirements Specification (SRS) document defines the complete requirements for the Student Productivity Platform, version 1.0. The platform is a centralized web and mobile application designed to help university students efficiently manage their academic activities, reduce stress, and improve productivity by consolidating multiple disconnected systems into a single unified solution.

The purpose of this SRS is to:

- Define the functional and non-functional requirements for the Student Productivity Platform
- Establish the basis for agreement between stakeholders (students, client representatives, and the development team)
- Provide a baseline for validation and verification activities
- Facilitate the estimation of project costs and schedules
- Serve as a reference for future development and maintenance

**Intended Audience:**

- **Development Team:** Hadia Yasir, Muhammad Bilal, and Waseem Ullah
- **Client Representatives:** Shazia Nazir and Maryam Noor ul Ain
- **Project Evaluators:** Department of Computer Science faculty
- **Quality Assurance Team:** For testing and validation
- **Future Stakeholders:** University administrators and potential users

## 1.2 Scope

**Product Name:** Student Productivity Platform

**Product Description:**

The Student Productivity Platform is a comprehensive web and mobile application that addresses the challenge of fragmented academic task management faced by university students. Currently, students juggle multiple disconnected systems including Learning Management Systems (LMS), email clients, messaging applications, calendars, and physical planners to track assignments, announcements, and deadlines. This fragmentation leads to missed tasks, poor time management, and increased academic stress.

The platform serves as a centralized hub that aggregates academic information from the university's LMS and integrates it with personal task management capabilities. It goes beyond traditional to-do lists by incorporating workload visualization, intelligent reminders, gamification elements, collaboration tools, and wellness features specifically designed for the student experience.

The system empowers students to take control of their academic journey by providing a clear overview of their commitments, helping them prioritize effectively, encouraging healthy study habits, and fostering peer collaboration and support.

**Benefits:**

- **Reduced Cognitive Load:** Eliminates the need to monitor multiple platforms by providing a single source of truth for all academic tasks and deadlines
- **Improved Time Management:** Visual workload indicators and intelligent prioritization help students allocate their time effectively
- **Enhanced Academic Performance:** Timely reminders and progress tracking ensure students stay on top of assignments and maintain consistent study habits
- **Stress Reduction:** Wellness features and workload awareness help prevent burnout and promote healthy study-life balance
- **Better Collaboration:** Built-in tools for study groups and peer support facilitate effective teamwork on group projects
- **Increased Motivation:** Gamification elements make task completion more engaging and rewarding

**Objectives:**

- **Centralization:** Aggregate academic data from university LMS and provide a unified interface for all student tasks
- **Usability:** Create an intuitive, fast, and accessible interface that students will actively choose to use daily
- **Engagement:** Implement gamification and social features that make productivity enjoyable and sustainable
- **Wellness:** Integrate features that help students maintain mental health and prevent academic burnout
- **Scalability:** Build a robust system that can grow from initial deployment to support thousands of students
- **Privacy:** Ensure student data remains secure, private, and fully under their control

**Out of Scope:**

- Modification of university LMS data or institutional records
- Automated grade calculation or GPA prediction
- Direct messaging with instructors (existing university systems will be used)
- Integration with non-academic personal calendar events
- Financial management or tuition payment features

### 1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
SRS	Software Requirements Specification
IEEE	Institute of Electrical and Electronics Engineers
LMS	Learning Management System (e.g., Canvas, Moodle)
API	Application Programming Interface
REST	Representational State Transfer
UI	User Interface
SSO	Single Sign-On
LDAP	Lightweight Directory Access Protocol
SAML	Security Assertion Markup Language
GPA	Grade Point Average
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
RSVP	Répondez s'il vous plaît (Please respond)
PDF	Portable Document Format
TLS	Transport Layer Security
WCAG	Web Content Accessibility Guidelines
RBAC	Role-Based Access Control
Workload Indicator	Color-coded visual representation (green/yellow/red) of student's current academic pressure level
Streak	Consecutive days of task completion
Badge	Digital achievement award unlocked by completing specific milestones

Table 2: Definitions, Acronyms, and Abbreviations

### 1.4 References

1. IEEE Std 830-1984, IEEE Guide to Software Requirements Specifications
2. Namal University Student Information System Documentation
3. Canvas LMS API Documentation, Instructure Inc.
4. WCAG 2.1 Guidelines, World Wide Web Consortium (W3C)

### 1.5 Overview

This document is organized as follows:

- **Section 1 - Introduction:** Provides an overview of the SRS, including purpose, scope, definitions, and references.
- **Section 2 - Overall Description:** Describes the general factors that affect the product and its requirements.
- **Section 3 - Specific Requirements:** Details the functional and non-functional requirements, external interfaces, and other specific requirements.



- **Appendices:** Contains supporting diagrams including Context Diagram and Use Case Diagram.

## 2 Overall Description

---

This section provides a high-level overview of the Student Productivity Platform, its context, and the factors that influence its requirements.

### 2.1 Product Perspective

The Student Productivity Platform is a centralized, web- and mobile-based application designed to help university students efficiently manage their academic activities, reduce stress, and improve productivity. Currently, students rely on multiple disconnected systems such as Learning Management Systems (LMS), email, messaging applications, calendars, and physical planners to track assignments, announcements, and deadlines. This fragmented workflow often leads to missed tasks, poor time management, and academic burnout.

The proposed system acts as a single unified platform that aggregates academic information from the university's LMS (e.g., Canvas) and allows students to manage both academic and personal tasks in one place. The system enhances traditional task management with workload visualization, reminders, gamification, collaboration tools, and wellness features, creating a holistic productivity solution tailored for students.

The platform will integrate with existing university systems for read-only access to academic data, ensuring that institutional data remains unchanged while being presented in a student-friendly format.

#### **System Context:**

The Student Productivity Platform interacts with the following external entities:

- **University LMS:** Read-only integration to fetch assignments, grades, and announcements
- **Students:** Primary users who access the system via web browsers and mobile applications
- **System Administrators:** Maintain system availability, security, and integrations
- **Cloud Infrastructure:** Hosting environment for backend services and databases
- **Notification Services:** Third-party services for push notifications and email reminders

**Note:** A detailed Context Diagram is provided in Appendix A.

### 2.2 Product Functions

At a high level, the Student Productivity Platform provides the following major functions:

1. **Centralized Dashboard:** Displays upcoming deadlines, new grades, and important announcements in a unified view
2. **Task Management:** Enables task creation, prioritization, deadline setting, reminders, estimated completion time, and progress tracking for both academic and personal tasks

3. **Workload Visualization:** Uses color-coded indicators (green, yellow, red) to help students understand their current academic pressure levels
4. **Gamification System:** Implements points, badges, streaks, levels, and optional leaderboards to motivate consistent task completion
5. **Collaboration Features:** Facilitates study groups, group chat, file sharing, and task assignment for group projects
6. **Course Discussion Forums:** Provides course-based forums where students can ask questions, receive peer answers, and mark solutions
7. **Peer Mentorship Matching:** Connects senior and junior students based on courses, interests, and availability
8. **Wellness and Burnout Prevention:** Includes break reminders, stress level tracking, productivity reports, and mindfulness suggestions
9. **Secure Authentication:** Ensures user data privacy and protection through secure login and authorization mechanisms

## 2.3 User Characteristics

The Student Productivity Platform is designed for the following user classes:

User Type	Characteristics	Technical Expertise
Undergraduate Students	Aged 18-25; regular users of smartphones and web applications; experience high academic workload and stress, especially during midterms and finals; primary target users	Moderate to high technical proficiency; require simple, fast, and intuitive interfaces
Postgraduate Students	Similar usage patterns to undergraduates but with more focus on research tasks and long-term planning; considered for future scope	High technical proficiency; comfortable with complex features and integrations
System Administrators	Responsible for maintaining system availability, security, and integrations; do not access student personal or wellness data	Expert technical proficiency; familiar with system architecture, deployment, and troubleshooting

Table 3: User Characteristics

## 2.4 Operating Environment

The Student Productivity Platform will operate in the following environments:

### Web Application:

- Accessible through modern web browsers (Chrome, Firefox, Edge, Safari)
- Responsive design for desktops, laptops, and tablets
- Minimum screen resolution support: 320px width

**Mobile Application:**

- Android platform (Android 8.0 Oreo and higher)
- iOS platform (iOS 13.0 and higher)
- Optimized for performance and low startup time

**Backend Environment:**

- Cloud-based infrastructure
- Scalable server architecture to support growing number of users
- Secure database systems for storing user tasks and preferences
- RESTful API architecture for client-server communication

## 2.5 Constraints

The following constraints will limit design and implementation options:

- **Regulatory Policies:** The system must comply with university data privacy rules and applicable data protection regulations
- **Integration Constraints:** The system will access university LMS data in read-only mode to comply with institutional policies
- **Performance Requirements:** Pages must load within strict time limits (2 seconds for main pages) even under high user load
- **Scalability Constraints:** Development must allow for future scalability to 200,000 users without major architectural changes
- **Network Reliability:** Internet connectivity may be unstable; the system must handle temporary network failures gracefully
- **Platform Compatibility:** The system must support multiple browsers and mobile platforms simultaneously
- **Data Privacy:** Student wellness data must remain private and accessible only to the individual student
- **Security Standards:** All data transmission must be encrypted, and the system must protect against common web vulnerabilities

## 2.6 Assumptions and Dependencies

The system design is based on the following assumptions and dependencies:

### Assumptions:

- Students have regular access to smartphones or internet-enabled devices
- The university LMS provides APIs or integration methods for fetching student data
- Students are willing to actively use task management and productivity tools if they are engaging and rewarding
- University approval is granted for accessing academic data in a secure, read-only manner
- Users will maintain updated web browsers and mobile operating systems

### Dependencies:

- Availability and reliability of the university LMS APIs
- Stable cloud infrastructure for hosting and data storage
- Notification services (push notifications, email) for reminders and alerts
- Third-party authentication services (if used for single sign-on)
- Continuous internet connectivity for real-time features

## 2.7 Apportioning of Requirements

The following requirements have been identified for future versions of the Student Productivity Platform to allow for phased development and deployment:

Requirement ID	Description	Target Version
FR-Future-1	Integration with external calendar systems (Google Calendar, Outlook)	v2.0
FR-Future-2	AI-powered study recommendations based on learning patterns	v2.0
FR-Future-3	Video conferencing integration for virtual study sessions	v2.0
FR-Future-4	Mobile offline mode with full functionality	v2.0
FR-Future-5	Postgraduate student features including re-search task tracking	v2.0
FR-Future-6	Integration with university library system for resource discovery	v2.5
FR-Future-7	Advanced analytics dashboard for students to analyze study patterns	v2.5
FR-Future-8	Multi-language support (Urdu, Arabic, English)	v3.0

Table 4: Future Requirements

## 3 Specific Requirements

---

This section contains all the detailed requirements for the system. Each requirement is uniquely identified and written in a clear, unambiguous manner.

### 3.1 External Interfaces

#### 3.1.1 User Interfaces

The Student Productivity Platform shall provide intuitive and responsive user interfaces across web and mobile platforms:

- **UI-1: Login Screen** - Clean interface with university branding, email/password fields, "Remember Me" option, "Forgot Password" link, and SSO button
- **UI-2: Dashboard** - Centralized view displaying upcoming deadlines (7-day default), recent grade releases, workload indicator, streak counter, and quick-add task button
- **UI-3: Task Management Page** - List/grid view of tasks with filtering options, status indicators, priority colors, and detailed task cards showing description, due date, attachments
- **UI-4: Gamification Page** - Visual display of points, badges (locked/unlocked), current streak with flame icon, leaderboards (global/friends/course), and active challenges
- **UI-5: Collaboration Hub** - Study groups list, group creation interface, chat windows, forum threads, and mentor matching interface
- **UI-6: Wellness Center** - Stress logging widget, workload breakdown chart, wellness resource library, break reminder settings, and weekly report archive
- **UI-7: Profile & Settings** - User profile display with edit capability, notification preferences panel, privacy controls, and account management options
- **UI-8: Mobile Navigation** - Bottom navigation bar with 5 main sections: Dashboard, Tasks, Collaborate, Wellness, Profile
- **UI-9: Responsive Design** - All interfaces shall adapt seamlessly to screen sizes from 320px (mobile) to 2560px (desktop) width
- **UI-10: Accessibility Features** - High contrast mode, screen reader compatibility, keyboard navigation support, and color-blind friendly palettes

#### 3.1.2 Hardware Interfaces

- **HI-1: Mobile Device Sensors** - The system shall utilize device notification services for push notifications and may access device storage for file attachments
- **HI-2: Network Interface** - The system shall communicate with university LMS servers and cloud infrastructure over HTTPS connections

- **HI-3: Storage Requirements** - Client devices shall have minimum 100MB free storage for mobile app installation and local caching

### 3.1.3 Software Interfaces

Interface ID	Software Component	Description
SI-1	University LMS (Canvas)	REST API integration for read-only access to assignments, grades, announcements, and course enrollment data
SI-2	Cloud Database (PostgreSQL/MongoDB)	Storage and retrieval of user tasks, profiles, gamification data, and wellness logs
SI-3	Authentication Service (LDAP/SAML)	University SSO integration for secure user authentication and authorization
SI-4	Push Notification Service (Firebase/APNs)	Delivery of push notifications to mobile devices (Android and iOS)
SI-5	Email Service (SMTP)	Sending of email reminders, reports, and account notifications
SI-6	File Storage (AWS S3/Cloud Storage)	Storage of user-uploaded files (attachments, profile pictures, group icons)
SI-7	Antivirus Scanner API	Malware scanning of uploaded files before storage
SI-8	Web Browsers	Chrome 90+, Firefox 88+, Safari 14+, Edge 90+ for web application
SI-9	Mobile OS	Android 8.0+, iOS 13.0+ for mobile applications

Table 5: Software Interfaces

### 3.1.4 Communications Interfaces

- **CI-1: HTTPS Protocol** - All client-server communication shall use HTTPS (TLS 1.2 or higher) for encrypted data transmission
- **CI-2: RESTful API** - Backend services shall expose RESTful APIs using JSON data format for client applications

- **CI-3: WebSocket Protocol** - Real-time features (chat, leaderboard updates, synchronization) shall use WebSocket connections
- **CI-4: Email Protocol** - The system shall use SMTP protocol for sending emails, with SPF/DKIM authentication
- **CI-5: Push Notification Protocols** - Firebase Cloud Messaging (FCM) for Android and Apple Push Notification Service (APNs) for iOS
- **CI-6: Data Format** - API requests and responses shall use JSON format with UTF-8 encoding
- **CI-7: API Rate Limiting** - Client applications shall respect rate limits: 100 requests per minute per user, with exponential backoff for LMS API calls

## 3.2 Functional Requirements

This section specifies all functional requirements for the Student Productivity Platform. Each requirement is uniquely identified using the format FR-[Module]-[Number] and written in clear, testable language.

### 3.2.1 Centralized Dashboard Module

#### FR-1.1 LMS Data Aggregation

- **FR-1.1.1** The system shall retrieve assignments, grades, and announcements from university LMS via REST API
- **FR-1.1.2** The system shall complete data synchronization within 5 minutes of content posting in LMS
- **FR-1.1.3** The system shall poll LMS API every 15 minutes for updated information
- **FR-1.1.4** The system shall handle API rate limits by queuing requests and implementing exponential backoff
- **FR-1.1.5** The system shall log all API communication attempts including timestamps and response codes

#### FR-1.2 Deadline Display

- **FR-1.2.1** The system shall display upcoming assignment deadlines in chronological order on main dashboard
- **FR-1.2.2** The system shall show next 7 days of deadlines by default
- **FR-1.2.3** The system shall include course name, assignment title, due date/time, and submission status for each deadline
- **FR-1.2.4** The system shall visually distinguish between past-due, due-today, and future deadlines using color coding



- **FR-1.2.5** The system shall allow users to expand view to show deadlines beyond 7 days

### **FR-1.3 Grade Release Notification**

- **FR-1.3.1** The system shall detect new grade postings within 15 minutes of LMS update
- **FR-1.3.2** The system shall display grade releases with course name, assignment title, score, and posting timestamp
- **FR-1.3.3** The system shall mark new grades with "NEW" indicator until user views them
- **FR-1.3.4** The system shall maintain history of all grade releases for current semester
- **FR-1.3.5** The system shall calculate and display semester GPA based on released grades

### **FR-1.4 Notification Categorization**

- **FR-1.4.1** The system shall categorize all notifications into four types: Academic, Administrative, Social, Wellness
- **FR-1.4.2** Academic notifications shall include: assignments, grades, course announcements, exam schedules
- **FR-1.4.3** Administrative notifications shall include: registration deadlines, university announcements, policy updates
- **FR-1.4.4** Social notifications shall include: study group invitations, forum replies, mentor matches
- **FR-1.4.5** Wellness notifications shall include: break reminders, stress check-ins, challenge completions
- **FR-1.4.6** The system shall allow users to assign custom categories to manually created items

### **FR-1.5 Dashboard Filtering**

- **FR-1.5.1** Users shall be able to filter dashboard content by course using dropdown selector
- **FR-1.5.2** Users shall be able to filter dashboard by date range (today, this week, this month, custom range)
- **FR-1.5.3** Users shall be able to filter dashboard by notification type (Academic, Administrative, Social, Wellness)
- **FR-1.5.4** The system shall apply multiple filters simultaneously using AND logic
- **FR-1.5.5** The system shall persist user's last-used filter settings across sessions
- **FR-1.5.6** The system shall display active filters with option to clear all filters

### FR-1.6 Cross-Platform Synchronization

- **FR-1.6.1** The system shall synchronize dashboard data between web and mobile applications within 10 seconds of changes
- **FR-1.6.2** The system shall use push synchronization for real-time updates when devices are online
- **FR-1.6.3** The system shall detect conflicts when same item modified on multiple devices and apply last-write-wins resolution
- **FR-1.6.4** The system shall queue local changes when offline and sync automatically upon reconnection
- **FR-1.6.5** The system shall display sync status indicator showing last successful sync timestamp

### 3.2.2 Gamification Engine

#### FR-2.1 Point Award System

- **FR-2.1.1** The system shall award 10 points when user completes task before or on deadline
- **FR-2.1.2** The system shall award 5 points when user completes task after deadline
- **FR-2.1.3** The system shall award 15 bonus points for completing task more than 48 hours before deadline
- **FR-2.1.4** The system shall award 20 points for achieving 7-day completion streak
- **FR-2.1.5** The system shall award 2 points for daily login (maximum once per 24 hour period)
- **FR-2.1.6** The system shall award 5 points for creating study group or joining forum discussion
- **FR-2.1.7** Points shall be immediately reflected in user's total score and visible on profile

#### FR-2.2 Daily Streak Tracking

- **FR-2.2.1** The system shall maintain counter of consecutive days user completes at least one task
- **FR-2.2.2** Streak shall increment by 1 when user completes task on calendar day different from previous completion
- **FR-2.2.3** Streak shall reset to 0 if user fails to complete any task for 24 consecutive hours after last completion
- **FR-2.2.4** The system shall display current streak prominently on dashboard with flame icon

- **FR-2.2.5** The system shall notify user when streak reaches milestones: 3, 7, 14, 30, 60, 100 days
- **FR-2.2.6** The system shall allow users to "freeze" streak once per month to preserve it during planned breaks

### **FR-2.3 Badge Unlocking System**

- **FR-2.3.1** The system shall unlock "Early Bird" badge after user completes 10 tasks before deadline
- **FR-2.3.2** The system shall unlock "Week Warrior" badge after user maintains 7-day completion streak
- **FR-2.3.3** The system shall unlock "Semester Champion" badge after user completes 50 tasks in single semester
- **FR-2.3.4** The system shall unlock "Collaboration Master" badge after user participates in 5 study groups
- **FR-2.3.5** The system shall unlock "Wellness Advocate" badge after user logs stress level for 30 consecutive days
- **FR-2.3.6** The system shall unlock "Mentor" badge after user successfully mentors 3 students
- **FR-2.3.7** The system shall unlock "Forum Expert" badge after user receives 25 upvotes on forum answers
- **FR-2.3.8** The system shall display badge unlock notification with animation and congratulatory message
- **FR-2.3.9** Badges shall be displayed in user profile with locked/unlocked visual states

## **3.3 Non-Functional Requirements**

This section specifies all non-functional requirements for the Student Productivity Platform.

### **3.3.1 Performance Requirements**

- **NFR-P-1** The system shall load main pages (dashboard and task list) in less than 2 seconds under normal load conditions
- **NFR-P-2** The system shall respond to user interactions (button clicks, task marking) within 0.5 seconds
- **NFR-P-3** The mobile application shall launch and be fully operational within 3 seconds
- **NFR-P-4** The system shall support up to 5,000 concurrent users without performance degradation
- **NFR-P-5** The system architecture shall be designed to scale to 50,000 total registered students

### 3.3.2 Security Requirements

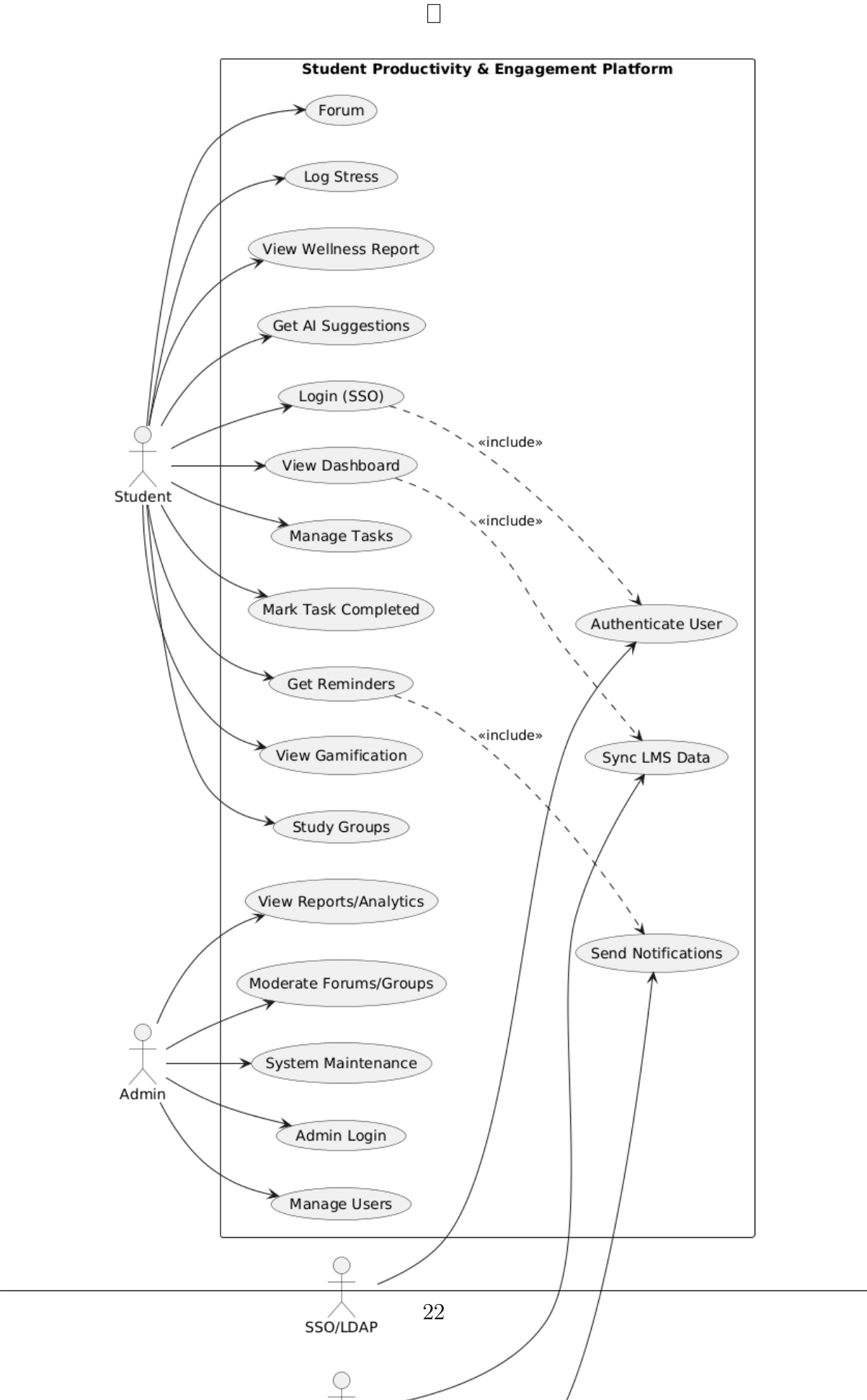
- **NFR-S-1** The system shall encrypt all data transmitted between client and server using TLS 1.2 or higher
- **NFR-S-2** The system shall enforce password requirements of minimum 8 characters including uppercase letters, lowercase letters, numbers, and special symbols
- **NFR-S-3** The system shall automatically log out users after 30 minutes of inactivity
- **NFR-S-4** The system shall lock user accounts for 30 minutes after 5 consecutive failed login attempts
- **NFR-S-5** The system shall restrict access to student wellness information exclusively to the individual student
- **NFR-S-6** The system shall scan all uploaded files for viruses and malware before storage
- **NFR-S-7** The system shall encrypt all stored passwords using industry-standard hashing algorithms (bcrypt or Argon2)
- **NFR-S-8** The system shall protect against common web vulnerabilities including SQL injection, XSS, and CSRF

### 3.3.3 Usability Requirements

- **NFR-U-1** The system shall enable new users to understand the interface and complete their first task within 5 minutes
- **NFR-U-2** The system shall provide a responsive user interface that adapts to smartphones, tablets, and desktop computers
- **NFR-U-3** The system shall maintain consistent button styles, colors, and menu layouts across all pages
- **NFR-U-4** The system shall comply with WCAG 2.1 Level AA accessibility standards
- **NFR-U-5** The system shall display an interactive tutorial during the first login session
- **NFR-U-6** The system shall provide clear error messages that explain problems and suggest corrective actions
- **NFR-U-7** The system shall use intuitive icons and labels familiar to students for common actions



# A Context Diagram



**Description:** The Context Diagram illustrates the boundaries of the Student Productivity Platform and its interaction with external entities.

## B Use Case Diagram

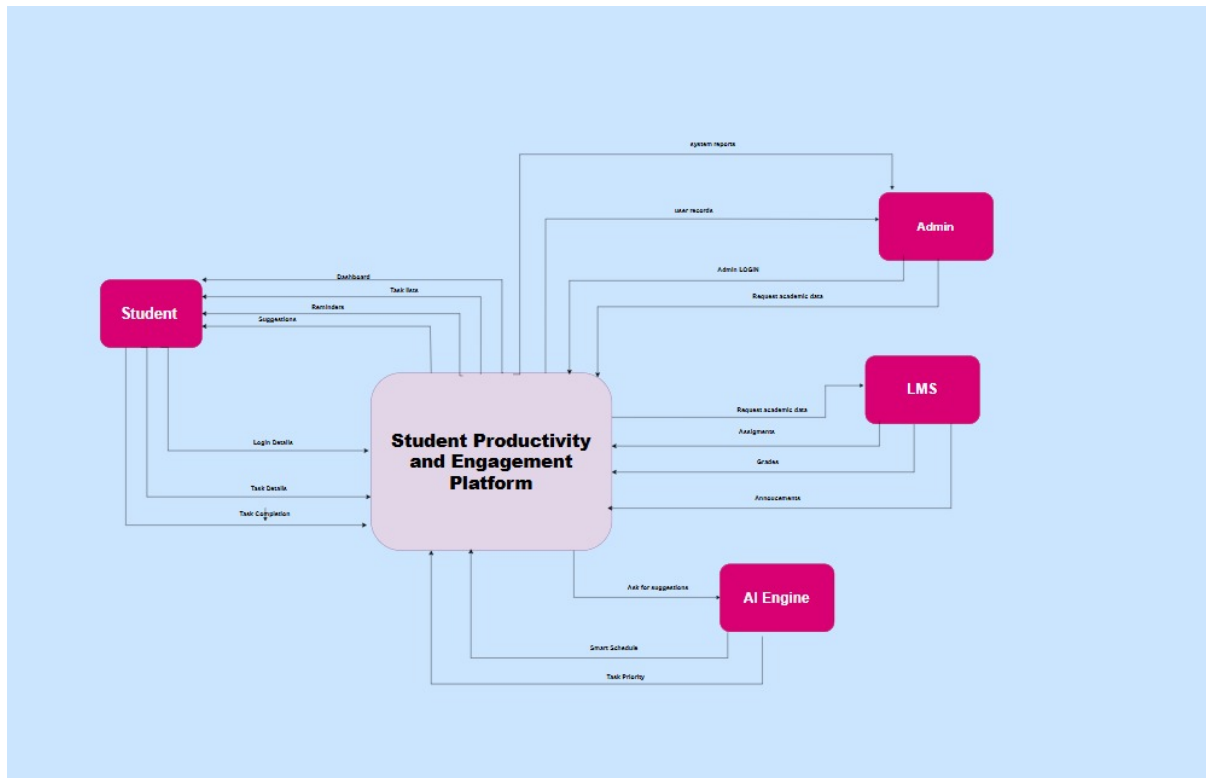


Figure 3: Use Case Diagram showing system actors and their interactions

## C Glossary

**Assignment** A task or homework item posted by an instructor in the university LMS

**Badge** A digital achievement award unlocked when students reach specific milestones

**Dashboard** The main landing page displaying aggregated information

**Gamification** The application of game-design elements to increase user engagement

**LMS** Learning Management System where instructors post course materials

**Streak** A count of consecutive days during which a student completes at least one task

**Workload Indicator** A color-coded visual representing current academic pressure level