

**North South University**

**Department of Electrical & Computer Engineering**

**Software Requirements Specification**

**for**  
**Intelligent Task Planner**

**Version 1.0**

Prepared by:

*Zuhayer Islam (223 2061 642)*

*Tanvir Ahmed (223 1047 642)*

*Faheema Shaheed Tamanna (223 2144 642)*

*Shefa Tabassum (223 2993 042)*

This document is part of an assignment for the course CSE327

Submitted to Dr. Mohammad Rezwanul Huq

Date:

*August 4, 2025*

**Table of Contents**

[1.1 Purpose 4](#_Toc205231054)

[1.2 Intended Audience 4](#_Toc205231055)

[1.3 Intended Use 4](#_Toc205231056)

[1.4 Product Scope 5](#_Toc205231057)

[1.5 Definitions and Acronyms 5](#_Toc205231058)

[2. Overall Description 6](#_Toc205231059)

[2.1 User Needs 6](#_Toc205231060)

[2.2 Assumptions and Dependencies 6](#_Toc205231061)

[3. System Features and Requirements 7](#_Toc205231062)

[3.1 Functional Requirements 7](#_Toc205231063)

[1. FR1: User Registration 8](#_Toc205231064)

[2. FR2: User Login 9](#_Toc205231066)

[3. FR3: Add Task 10](#_Toc205231068)

[4. FR4: Edit Task 11](#_Toc205231070)

[5. FR5: Delete Task 12](#_Toc205231072)

[6. FR6: View Tasks 12](#_Toc205231074)

[7. FR7: View Analytics 14](#_Toc205231075)

[8. FR8: Manage Profile 14](#_Toc205231076)

[9. FR9: View Schedule 15](#_Toc205231077)

[10. FR10: Add Pomodoro Timer 16](#_Toc205231078)

[11. FR11: User Logout 16](#_Toc205231079)

[3.2 Non Functional Requirements 17](#_Toc205231080)

[1. Usability 17](#_Toc205231081)

[3.3 Attributes of the Requirements 18](#_Toc205231082)

[3.4 External Interface Requirements 19](#_Toc205231083)

[1. User Interface 19](#_Toc205231084)

[2. Hardware Interfaces 22](#_Toc205231085)

[3. Software Interfaces 23](#_Toc205231086)

[4. Communication Interfaces 23](#_Toc205231087)

[4. Other Requirements 23](#_Toc205231088)

[4.1 Database Requirements 23](#_Toc205231089)

[4.2 Legal and Regulatory Requirements 23](#_Toc205231090)

[4.3 Internationalization and Localization 24](#_Toc205231091)

[4.4 Risk Management (FMEA Matrix) 24](#_Toc205231092)

[5. Appendices 25](#_Toc205231093)

[5.1 Use Case Diagram 25](#_Toc205231094)

[5.2 To Be Determined (TBD) List 26](#_Toc205231095)

[5.3 Contributions in the SRS 26](#_Toc205231096)

**1. INTRODUCTION**

## 1.1 Purpose

This document is a detailed software requirements specification (SRS) for the ‘Intelligent Task Planner’ web application named ‘Productivo’. Productivo is a Django-based web application designed to help university students manage their academic workload and daily tasks. It aims to provide smart scheduling recommendations based on user inputs such as task deadlines, priorities, and available time.

## 1.2 Intended Audience

This product is primarily designed for students, particularly university students, to help them manage their schedules effectively.

## 1.3 Intended Use

The system allows registered users to manage their tasks and schedules effectively. It will support priority-based task sorting, smart time slot recommendations according to the user’s schedule, reminder notifications, and a Pomodoro timer and visual analytics to help track time management efficiency. The system will also allow students to make manual changes to their schedules when necessary, while adapting intelligently over time.

## 1.4 Product Scope

This system will:

* Allow students to add, edit, delete, and view academic tasks.
* Provide AI-assisted or rule-based scheduling suggestions.
* Support task prioritization, time blocking, reminders, and Pomodoro timing.
* Display analytics on time usage and productivity trends.
* Allow manual adjustment of task schedules.
* Sync with Google Calendar.

## 1.5 Definitions and Acronyms

* Task: A scheduled activity for a given day. The user inputs details such as day, date, priority, task type, and duration.
* Pomodoro Timer: A 25-minute timer that users can use to take breaks between tasks to improve work efficiency.
* UI: User Interface
* API: Application Programming Interface
* FR: Functional Requirement
* NFR: Non Functional Requirement
* EARS: Easy approach to Requirements Syntax.
* Gherkin: A simple language used to write test scenarios easy for technical or non -technical stakeholders to understand.

# 2. Overall Description

## 2.1 User Needs

University students require a platform to efficiently manage their day-to-day

academic activities. Productivo aims to provide this solution by addressing several

common challenges:

* Students require a platform to plan their schedules effectively and monitor their daily progress.
* Students need the application to provide timely reminders so that they don’t miss deadlines.
* Students need timer options to provide structured breaks between study sessions to maintain focus and productivity.
* Students need to be able to view and categorize tasks based on type, such as assignments, exams, or personal tasks, for better organization and clarity.
* Students need suggestions to improve their schedules.

## 2.2 Assumptions and Dependencies

Assumptions:

* Users will access the system through modern web browsers (e.g., Chrome, Firefox).
* Users will have a stable internet connection.
* Users are expected to register and log in before accessing core features.
* Students will input accurate information related to their tasks (e.g., date, time, type, priority).
* The system will be primarily used by students.

Dependencies:

* The system is developed using Django, which requires a compatible Python environment.
* The application depends on a relational database (e.g., PostgreSQL, MySQL) to store user data and task information.
* Google Calendar API integration is required to enable calendar sync functionality.
* Reminder notifications may depend on a third-party library or email/SMS service (e.g., SMTP, Twilio, or Push services).
* The Pomodoro timer and real-time components may rely on JavaScript (with WebSockets or AJAX).

# 3. System Features and Requirements

## 3.1 Functional Requirements

Functional requirements describe the specific behavior and functions the system

must perform. It covers user requirements, provides explanations for each requirement, and outlines possible scenarios of system behavior in response to user interactions.

### FR1: User Registration

### EARS: The system shall allow a new user to register by providing their first name, last name, date of birth, a unique username, email address and password.

**Gherkin:**

**Scenario:** Successful Registration

Given the user has entered a unique username and valid credentials

When the user clicks on the register button.

Then the user will be redirected to the Login page

And an email shall be sent notifying the user about their successful registration.

**Scenario:** Unsuccessful Registration

Given the user has entered a non-unique username or invalid credentials

When the user clicks on the register button

Then the user will be notified that there is an error in the data they provided

### FR2: User Login

### EARS: The system shall allow a user to login by providing their username and password.

**Gherkin:**

**Scenario:** Successful Login

Given the user has entered the correct username and password

When the user clicks on the login button

Then the user will be redirected to the Dashboard Page

**Scenario:** Unsuccessful Login

Given the user has entered incorrect credentials

When the user clicks on the login button

Then the user will be notified that there is an error in the data they have provided and

the user will stay in the Login Page.

### FR3: Add Task

### EARS: The system shall allow a user to add a Task by providing Task details such as title, description, priority, category, deadline, and reminder.

**Gherkin:**

**Scenario:** Successful Addition of Task

Given the user has entered the task details (title, description, priority, category,

deadline, and reminder)

When the user clicks on the “Add Task” button

Then the user will be redirected to the Dashboard Page and the Task will appear on

Dashboard Page, Task Page, Schedule Page and Analytics Page.

**Scenario:** Task collides with pre-existing Tasks

Given the user has entered a date and timing that overlaps with a previously added

Task.

When the user clicks on the "Add Task" button

Then the system notifies the user about the overlapping task

And the user chooses to either confirm or reject adding the overlapping task

When the user confirms

Then the task is added

And the user is redirected to the Dashboard page

And the task appears on the Dashboard page, Task page, Schedule page, and Analytics

page

When the user rejects

Then the task is not added

And the user remains on the Add Task page to modify the task details

### FR4: Edit Task

### EARS: The system shall allow a user to edit a Task by providing Task details such as title, description, priority, category, deadline, and reminder.

**Gherkin:**

**Scenario:** Successful edit of Task

Given the user has modified the task details (title, description, priority, category,

deadline, and reminder.)

for editing

When the user clicks on the “Edit Task” button

Then the user will be redirected to the Dashboard Page and the updated Task will

appear on Dashboard Page, Task Page, Schedule Page and Analytics Page.

### FR5: Delete Task

### EARS: The system shall allow a user to delete a Task.

**Gherkin:**

**Scenario:** Successful Deletion of Task

Given the user chooses a Task for deletion

When the user clicks on the “Delete Task” button

Then the user will be redirected to the Dashboard Page and the Task will be removed

from the Dashboard Page, Task Page, Schedule Page and Analytics Page.

### FR6: View Tasks

**EARS:** The system shall allow a user to view the following:

* View All tasks
* View Today’s tasks
* View Upcoming tasks
* View Completed tasks

**Gherkin:**

**Scenario:** Successful view of Task

Given the user chooses a Task for viewing

When the user clicks on the “Tasks” button

Then the user will be redirected to the Task Page.

When the user selects the "All" view

Then the user sees all their scheduled tasks

When the user selects the "Today" view

Then the user sees tasks scheduled for today

When the user selects the "Upcoming" view

Then the user sees tasks scheduled for upcoming dates

When the user selects the "Completed" view

Then the user sees all completed tasks

### FR7: View Analytics

**EARS:** The system shall allow the user to view their weekly/monthly/quarterly

progress.

**Gherkin:**

**Scenario:** Successful view of the progress

Given the user chooses to view their task analytics

When the user clicks on the “Analytics” button

Then the user is redirected to the Analytics Page.

When the user selects a particular view (weekly/monthly/quarterly)

Then the user sees their progress through charts/graphs, Task completion rate,

productivity tips etc.

### FR8: Manage Profile

**EARS:** The user shall be able to edit their profile information and other preferences

for using the application.

**Gherkin:**

**Scenario:** Successful change in user settings

Given the user chooses to view their Settings

When the user clicks on the “Settings” button

Then the user is redirected to the Settings Page which offers different settings

(Profile/ Preference/ Notifications/ Account).

When the user clicks on some settings and modifies then clicks “save changes” button

Then the users settings are edited.

### FR9: View Schedule

**EARS:** The user shall be able to view their daily/ weekly/monthly schedule

**Gherkin:**

**Scenario:** Successful view of schedule

Given the user chooses to view their schedule

When the user clicks on the “Schedule” button

Then the user is redirected to the Schedule Page which offers different schedule

views (Daily/ Weekly/ Monthly).

### FR10: Add Pomodoro Timer

**EARS:** The user shall be able to add a Pomodoro Timer to allow breaks between an activity.

**Gherkin:**

**Scenario:** Successful addition of the timer

Given the user chooses to add ‘Pomodoro Timer’ from their Dashboard

When the user clicks on the “Add PD Timer” button

Then the user can view a 25-minute countdown timer on the screen

And after 25 minutes, a 5-minute break timer is automatically started

And an alarm rings to indicate the start of the break

And after 5 minutes, another alarm rings to prompt the user to resume their activity

And after 4 such cycles, a long break of 15–30 minutes is offered

### FR11: User Logout

**EARS:** The user shall be able to logout of the system and will be redirected to the homepage.

## 3.2 Non Functional Requirements

This section outlines the quality attributes the system must meet, including usability,

performance, scalability, security, maintainability, and portability.

### Usability

The system shall provide a clean user interface. New students should be able to use the application without any prior training and complete registration and login within 3 minutes.

1. **Performance**

The system shall respond to 95% of request in within 2 seconds.

1. **Scalability**

The system shall support a minimum of 1000 concurrent users, maintaining performance levels defined in the previous section (Performance).

1. **Security**

* The system shall use hashed passwords to protect the user passwords.
* The system shall not allow unauthorized users to access the Admin (API). Only authenticated users with appropriate roles shall have access to the admin site.

1. **Maintainability**

* The codebase shall follow Django development best practices
* At least 80% of the codebase shall be documented using inline comments.

1. **Portability**

The system shall be fully functional on the latest two versions of major web browsers, including Chrome, Firefox, Safari, and Microsoft Edge. It shall be responsive and usable on desktop or laptop screen resolutions ( 1024 px width).

## 3.3 Attributes of the Requirements

|  |  |  |
| --- | --- | --- |
| **Requirement Number** | **Priority** | **Status** |
| FR1-User Registration | High | Approved |
| FR2-User Login | High | Approved |
| FR3-Add Task | High | Approved |
| FR4-Edit Task | Medium | Approved |
| FR5-Delete Task | Medium | Approved |
| FR6-View Tasks | High | Approved |
| FR7-View Analytics | Low | Approved |
| FR8-Manage Profile | Low | Approved |
| FR9-View Schedule | Medium | Approved |
| FR10-Add Pomodoro Timer | Low | Approved |
| FR11-User Logout | High | Approved |
| NFR1-Usability | High | Approved |
| NFR2-Performance | High | Approved |
| NFR3-Scalability | Medium | Approved |
| NFR4-Security | High | Approved |
| NFR5-Maintainability | Medium | Approved |
| NFR6-Portability | Medium | Approved |

**Note:** All requirements are sourced from discussions with teammates based on the requirements of the CSE327 project, as of August 4th 2025.

## 3.4 External Interface Requirements

### 1. User Interface

The system provides a web-based user interface accessible through a browser on devices connected to the local network. The interface includes pages for user login, registration, dashboard, and profile management. It follows responsive design principles for usability on standard desktop/laptop screens.

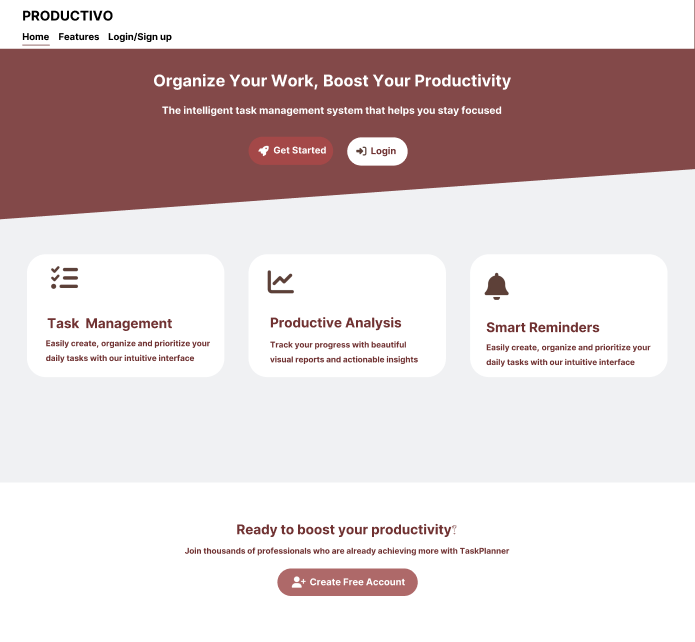


Figure 1: Home Page

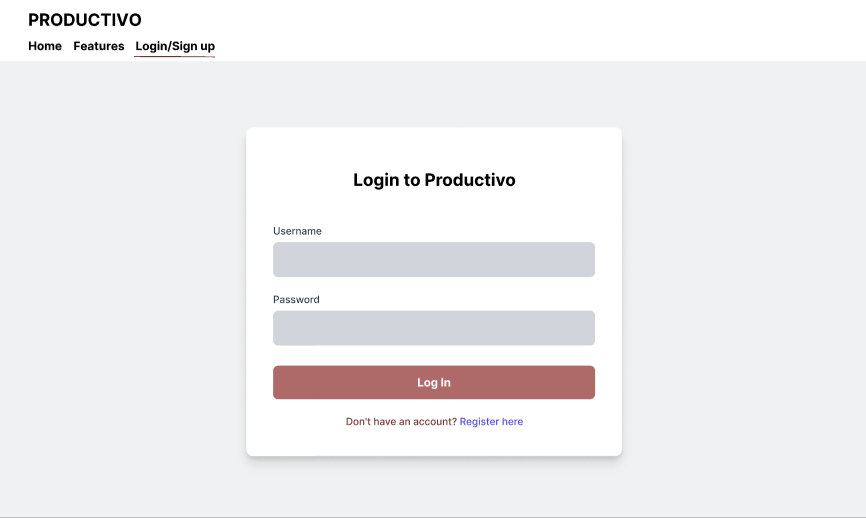


Figure 2: Login Page

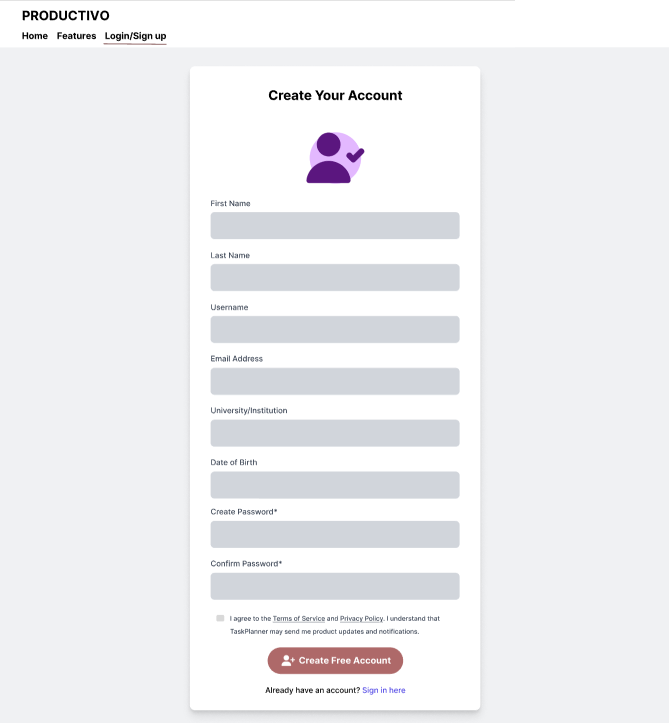


Figure 3: Registration Page

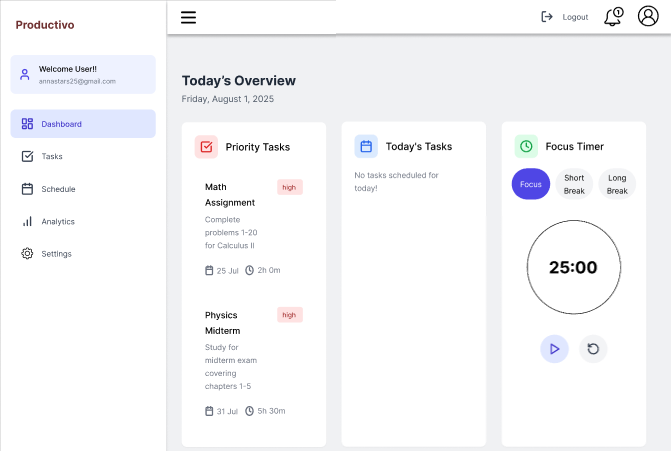


Figure 4: Dashboard Page

### 2. Hardware Interfaces

Not applicable since system does not interact with any external hardware components.

### 3. Software Interfaces

The system does not integrate with any external software systems or third-party APIs. It uses SQLite as the internal database backend, which is managed locally within the Django environment.

### 4. Communication Interfaces

The system is hosted locally using Django’s built-in development server and is accessible over the internal network. All communication between client browsers and the server occurs over HTTP.

# 4. Other Requirements

## 4.1 Database Requirements

The system uses SQLite as the backend database during development. It stores user credentials, course information, and other application data in a relational format. The Django Object Relational Mapping is used for database operations.

## 4.2 Legal and Regulatory Requirements

The project is currently developed for academic purpose and therefore does not include any specific legal or regulatory requirements. However, the project adheres to the ethical guidelines related to the production of responsible software.

## 4.3 Internationalization and Localization

Internationalization and Localization are not within the scope of this project. The application is developed to be used in English only.

## 4.4 Risk Management (FMEA Matrix)

Risk Management is not within the scope of this project.

# 5. Appendices

# 5.1 Use Case Diagram

Figure 5: Use Case Diagram

## 

## 

## 5.2 To Be Determined (TBD) List

1. Integration with Google Calendar to Sync Tasks.
2. Mechanism for providing task management suggestions (AI driven or rule based logic)

## 5.3 Contributions in the SRS

* *Tanvir Ahmed (223 1047 642)* – Created the Use Case Diagram
* *Faheema Shaheed Tamanna (223 2144 642)* – Designed the User Interface Mockups
* *Zuhayer Islam (223 2061 642)* –

Wrote several sections of the document, formatted the content, and integrated all components into a cohesive structure. Also ensured that all requirements were clearly understandable and conformed to the standards of a Software Requirements Specification (SRS).

* *Shefa Tabassum (223 2993 042)* –

Refined the Use Case Diagram after team discussion and analysis, contributed to the

functional requirements section and edited the document for clarity, consistency, and

overall quality.

Each team member contributed in their own way, bringing unique strengths to the development and completion of this Software Requirements Specification (SRS) document.