Software Requirements Specification for TYMATE v1.0
Prepared by: Jessica Mae Lanuzo
Date: December 13, 2025

Version: 1.0

1. Introduction

This section provides an overview of the SRS document and the TYMATE product.

1.1 Purpose

TYMATE is designed to help students, especially those balancing school and part-time work, to organize tasks and manage their time intentionally. This document enumerates the functional and non-functional requirements of TYMATE. It shall serve as a clear reference for the development team and stakeholders to stay aligned on what the system is expected to do, how it should do them, and under what limitations it shall operate within.

The SRS also serves as the basis for system validation, acceptance testing, and future maintenance.

1.2 Scope

TYMATE is a desktop and web-based productivity application designed for **working students and regular students** in an academic setting.

- **For Working Students:** Tymate shall allow working students to track their school and job-related tasks, be aware of each task's completion duration or available time for completion, and receive context-aware smart tips in choosing what to prioritize.
- **For Regular Students:** The app will provide these students the tools to be able to keep track of their school activities, as well as personal tasks, by adding, categorizing, and marking each of them depending on their status of completion.

The scope of the app is limited to individual user task management. It will **not** include calendar integration, multi-user collaboration, or third-party account linking, but these features may be implemented in future versions.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification
- **UI**: User Interface
- **Flet**: Python-based UI framework built on Flutter
- **Time Budget**: User-defined limit of daily free hours

1.4 References

- ISO/IEC/IEEE 29148:2018 - Systems and software engineering — Life cycle processes — Requirements engineering.
- National Institute of Standards and Technology (NIST). Advanced *Encryption Standard (AES) Specification (FIPS PUB 197)*

1.5 Overview

This document is organized into three main sections. Section 1 provides an introduction. Section 2 gives an overall description of the product, its users, and its operational constraints. Section 3 details the specific functional and non-functional requirements.

2. Overall Description

This section describes the general factors that affect the product and its requirements.

2.1 Product Perspective

TYMATE will run as a standalone app in desktop, as well as desktop and mobile browsers, and will be built using Flet. Its subsystems will include:

- **User Interface (UI) Layer**: Provides views for task management, time budgeting, and (in future) analytics visualization.
- **Local Storage Layer**: Handles encrypted local data storage to keep user information secure.

As it will be built using Flet, the app will run across desktop and browsers with consistent layout and behavior.

2.2 Product Functions

The major functions of Tymate include:

- Secure user onboarding and local data encryption
- Task creation, editing, and deletion
- Categorization of activities
- Time balance tracking
- Visualization of completed tasks and trends

2.3 User Characteristics

- **Working Student (Primary)**: A student who balances school and part-time work. This user is technologically literate and values efficient time management. They use Tymate to organize academic and work tasks, track how time is spent, and receive productivity suggestions to prevent burnout and missed deadlines.
- **Regular Student (Secondary)**: A student who focuses mainly on academic tasks. They use Tymate to manage assignments, monitor deadlines, and visualize time usage patterns.
- **Future Expansion Users (Optional)**: In future versions, educators or mentors may use the system to view analytics or help students in planning their workloads. However, the current version focuses solely on individual use.

2.4 Constraints

- The system must run on desktop browsers (Chrome, Edge, Firefox) and mobile browsers (Safari, Chrome Mobile).
- The application must be developed using **Flet**.
- All locally stored sensitive data must be encrypted using industry-standard encryption methods.
- User passwords must be hashed using bcrypt (industry standard for authentication).
- File uploads shall be limited to 5 MB per file.

2.5 Assumptions and Dependencies

- Web users are expected to have intermittent internet access.
- The system assumes a single-user environment for each instance.
- No third-party calendar or account synchronization is required.

3. Specific Requirements

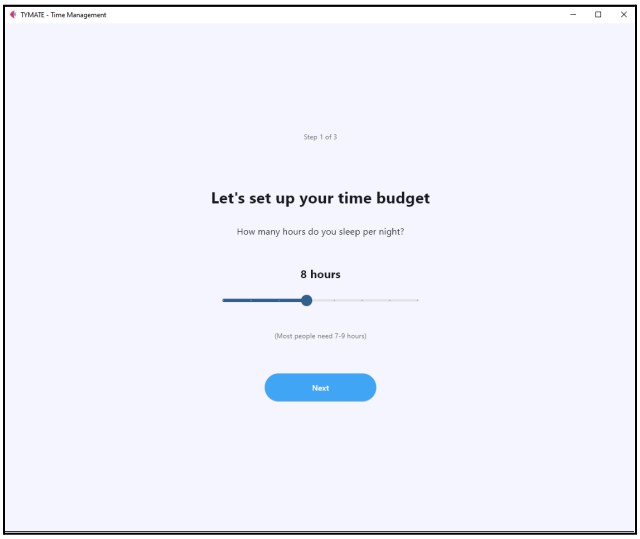
This section contains the detailed requirements necessary to build the system.

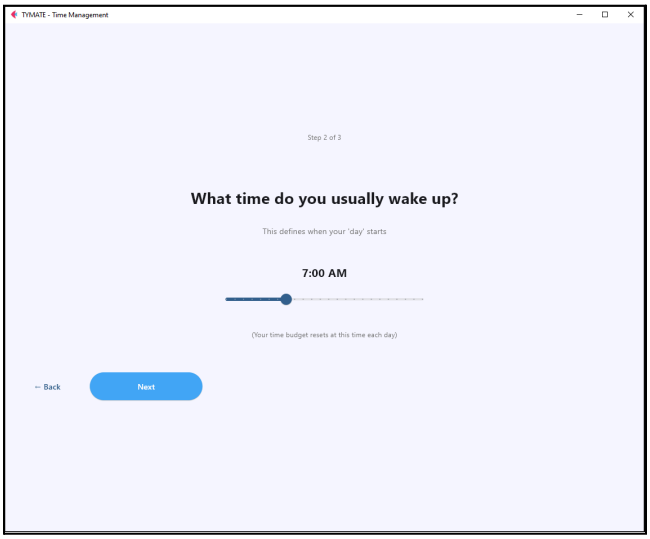
3.1 Functional Requirements

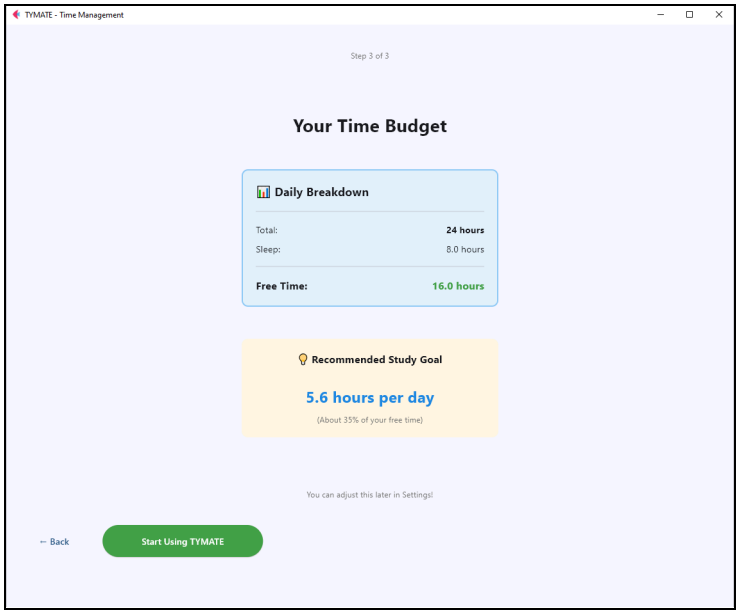
This details *what* the system should do.

FR-001: Time Budgeting

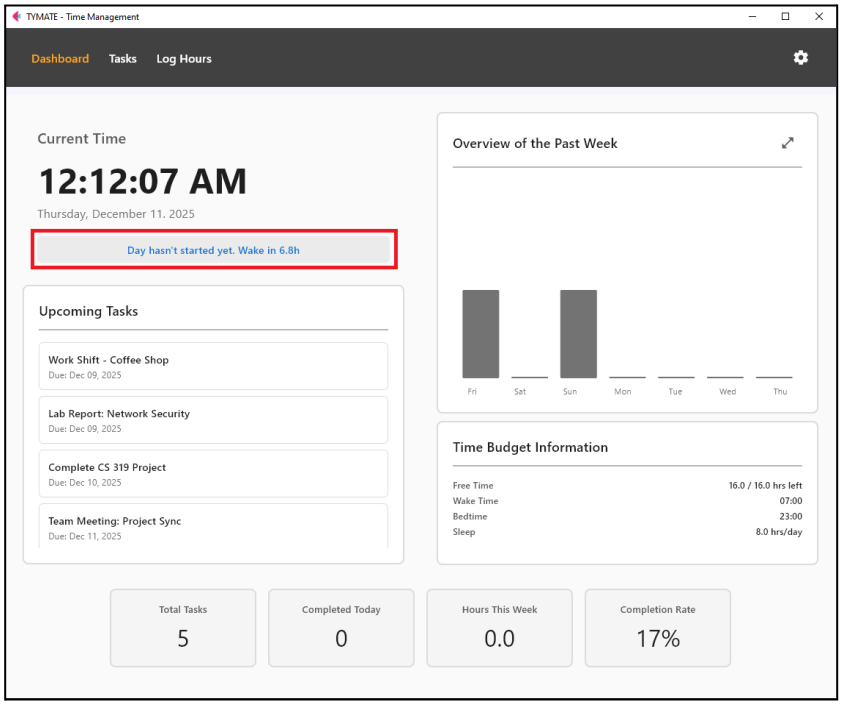
- **Description:** The system shall set a daily time budget using user input during onboarding. This will be used by the system to calculate and display user's remaining time for the day







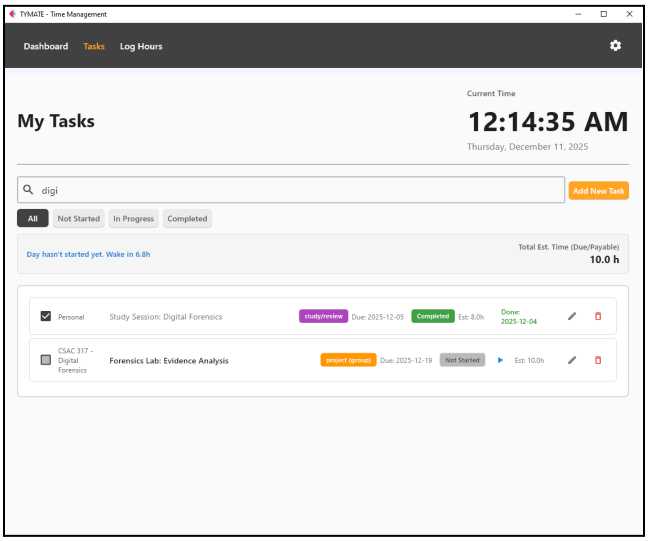
TYMATE's Onboarding/Setup Wizard and summary.

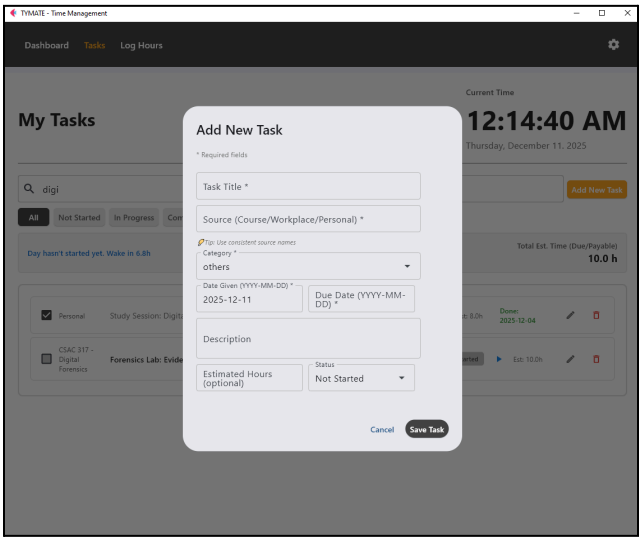


Tymate Dashboard with user's calculated remaining time (budget) for the day.

FR-002: Create Tasks

- **Description:** The system shall allow users to create tasks with a title, source, category, date given and due. It also allows for storing optional information such as task description and estimated time to accomplish.

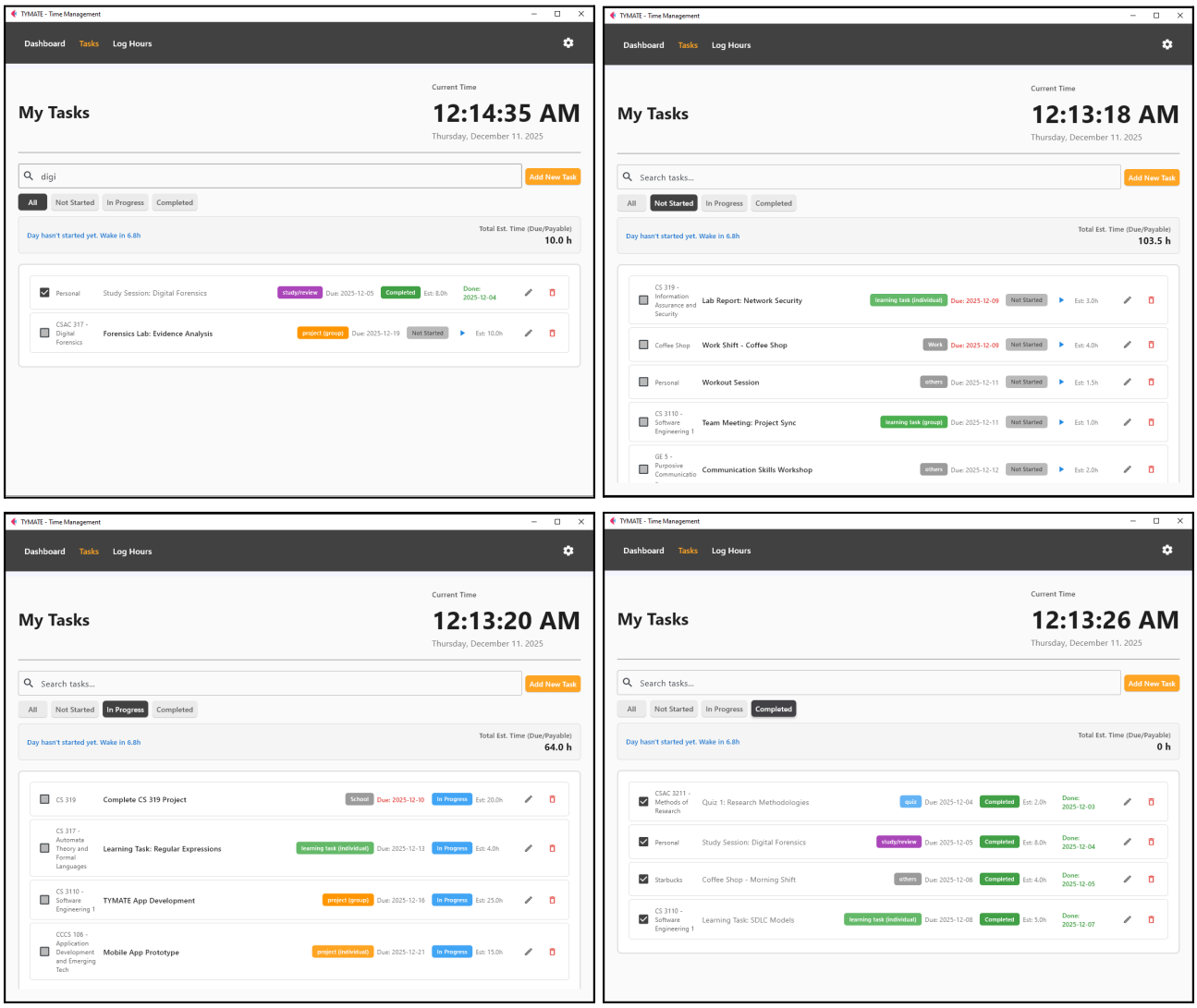




Tymate's Tasks Page and Add New Task Dialog.

FR-003: View Activities

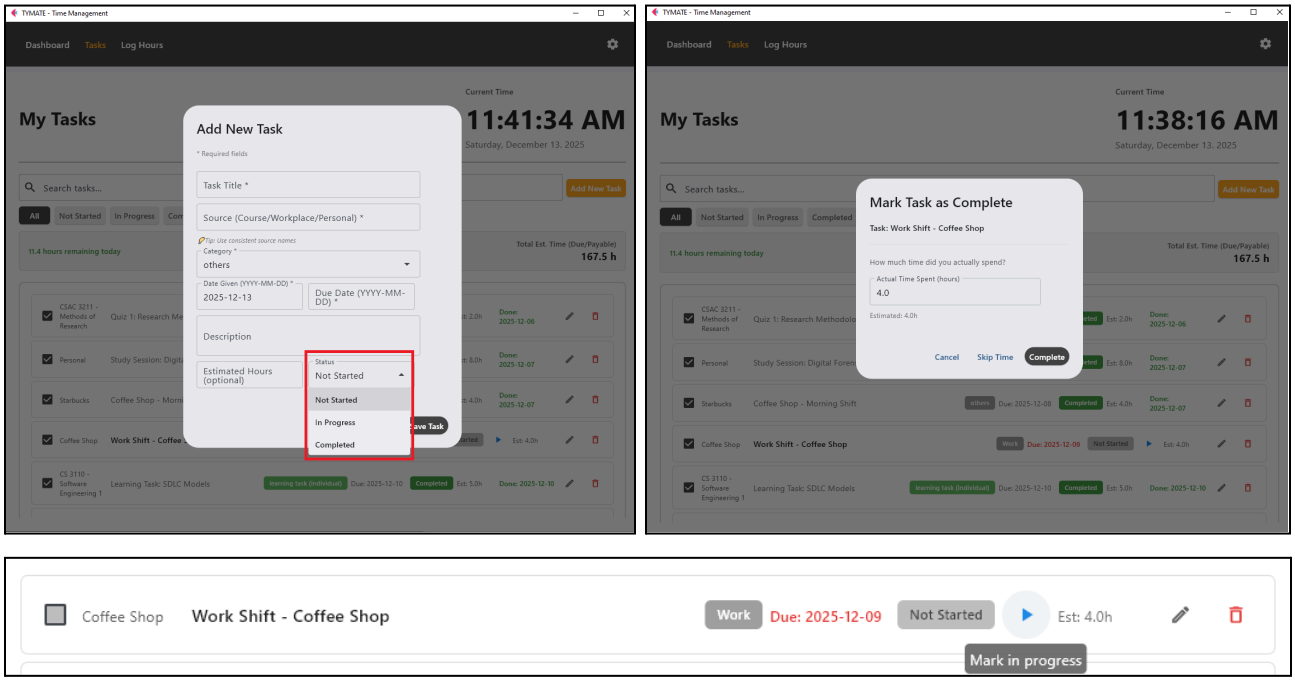
- **Description:** The system shall display the user's tasks in the Tasks Page of the app. This task list should be filterable by completion status to help users review ongoing and completed activities.



Tymate's Tasks Page with tasks filtered by completion status.

FR-004: Mark Progress

- **Description:** The system shall allow users to mark each task as Completed, In Progress, or Not Started. These progress indicators will help track the completion status of the user's activities.

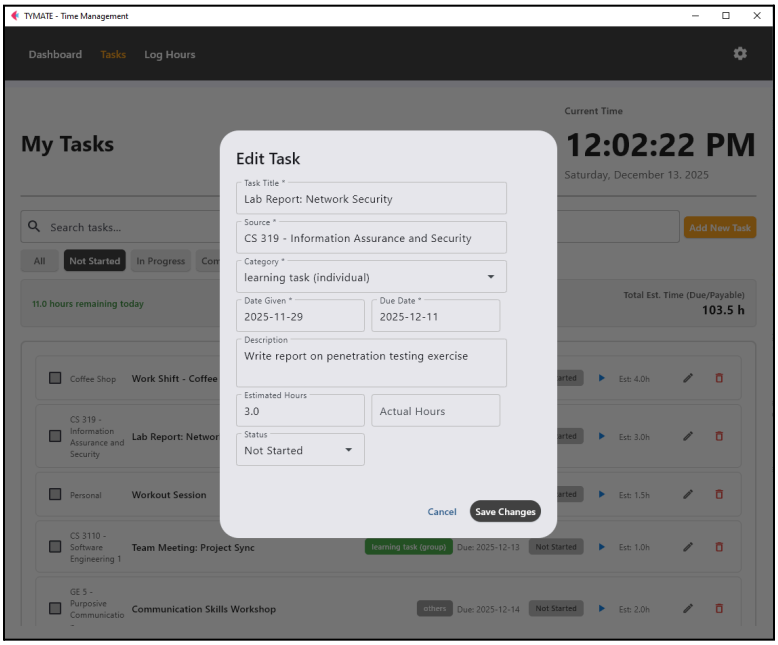


Ways of changing/assigning completion status to tasks during and after task creation.

FR-005: Edit or Delete Tasks

- **Description:** The system shall allow users to modify or delete existing task details. Upon completion, it will also allow users to input actual time spent to accomplish tasks and edit the date of completion.

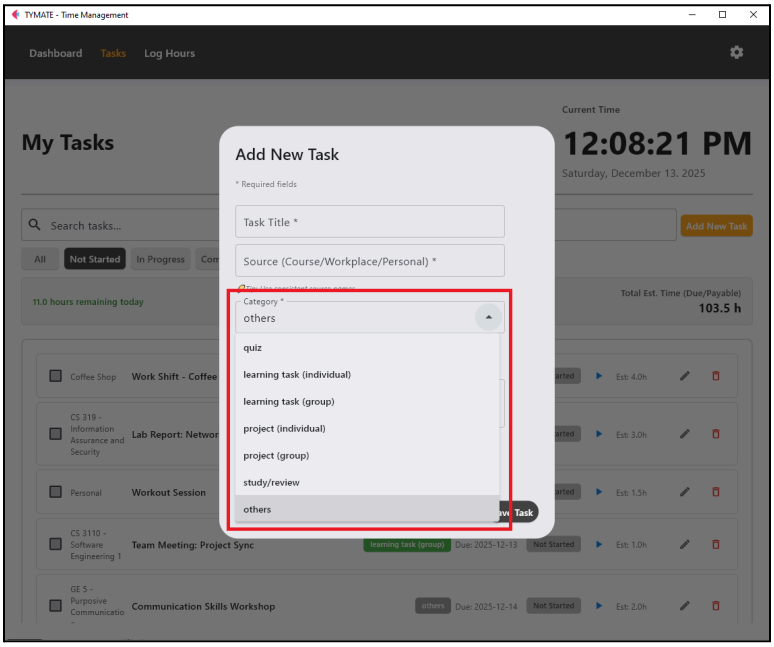
Edit task button in task item in the Tasks Page.



Tymate's Edit Task dialog.

FR-006: Categorize Tasks

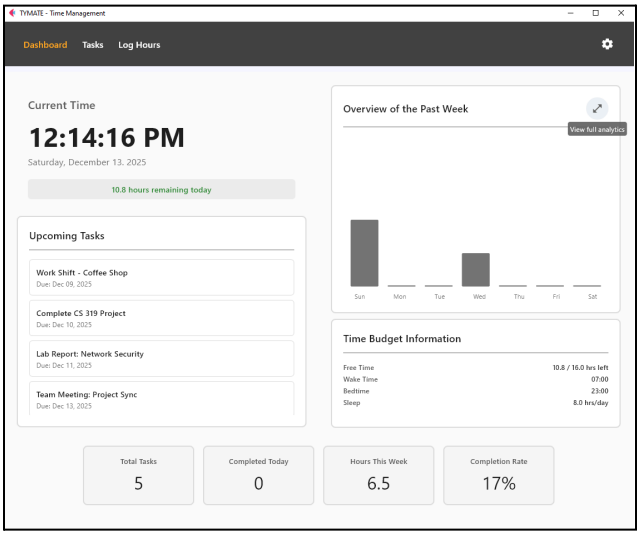
- **Description:** The system shall allow users to assign categories such as "quiz", "learning task", "project", and "others". The creation and modification of categories shall not be allowed.

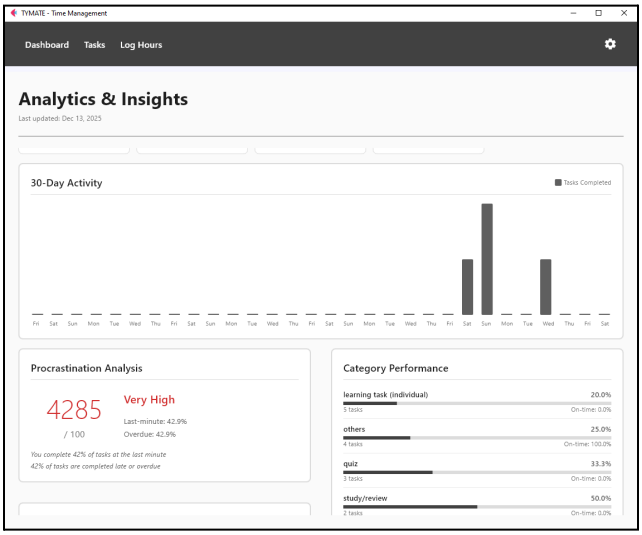


Tymate's Add New Task dialog with the category dropdown options.

FR-007: Time Balance Dashboard

- **Description:** The system shall present a dashboard summarizing total time spent, remaining budget, completed tasks, and trends in productivity. Charts and graphs will provide visual insight into how users allocate their time.





FR-008: Data Persistence

- **Description:** The system shall store all user data in an encrypted, persistent format to prevent data loss and maintain user data integrity.

3.2 Non-Functional Requirements

This details *how* the system should perform its functions.

NFR-001: Performance

- Description: The system shall respond to user input within 2 seconds and complete operations within 5 seconds on a stable 4G connection.

NFR-002: Usability

- Description: The user interface shall be intuitive. A new user shall be able to create and log tasks without requiring a tutorial.

NFR-003: Security

- Description: All user data shall be encrypted at rest and in transit. No sensitive credentials shall be stored in plaintext.

NFR-004: Reliability

- Description: The system shall maintain 99.5% uptime for core functions.

NFR-005: Portability

- Description: The system shall run on major desktop browsers (Chrome, Edge, Firefox) and mobile browsers (Safari, Chrome Mobile) with consistent layout and functionality.

NFR-006: Maintainability

- Description: The codebase shall follow a modular structure, separating UI, object models, and managers.

NFR-007: Scalability

- Description: The architecture shall support future extensions such as calendar integration or collaboration features without major redesign.

NFR-008: Documentation Compliance

- Description: The project shall include formal documentation such as user guides and developer READMEs compliant with the course standards.