Software Requirements Specification

for

<Task Management System>

Version <0.1>

Prepared by

Group Name: D11

Wu Chun Hei	**1155194804**	**1155194804@link.cuhk.edu.hk**
Yang Yufeng	**1155194222**	**1155194222@link.cuhk.edu.hk**
Li Chak Man	**1155194613**	**1155194613@link.cuhk.edu.hk**
<name>	**<student #>**	**<e-mail>**
<name>	**<student #>**	**<e-mail>**

Instructor: Dr. LAM Tak Kei

Course: CSCI 3100

Department: Computer Science

Date: 10/2/2025

CONTENTS **II**

REVISIONS **II**

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Yang Yufeng, Li Chak Man, Wu Chun Hei	Initial Draft	10/2/2025

1 Introduction

1.1 Document Purpose

This document outlines the requirements for TickPulse, a web-based task management system designed to help individuals organize, track and collaborate on tasks efficiently. The system will provide features such as task creation, prioritization, deadline reminders, showing to-do tasks on calendar, and customization of working period, rest period.

1.2 Product Scope

The system will:

- Allow users to create, edit and delete tasks with details like title, description of events, priority, and deadlines.
- Support task categorization using grouping of events and projects.
- Send notifications for deadlines, updates and reminders.
- Visualize to-do tasks on the calendar.
- Support focus mode, customization of working time and rest time of users' preference.

Out of scope:

- Mobile application development is currently not supported.

1.3 Intended Audience and Document Overview

This document is intended for:

- Client: To get familiar with the functionality of the projects
- Instructors: To evaluate the projects.

1.4 Definitions, Acronyms and Abbreviations

- UI: user interface
- TMS: Task Management System

1.5 Document Conventions

Formatting Conventions: this document follows the IEEE formatting requirements. Text uses Arial font size 11 throughout the document, single-spaced, with 1-inch margins. Section titles are in bold and content of each part are generally in point form for clear representation.

1.6 References and Acknowledgments

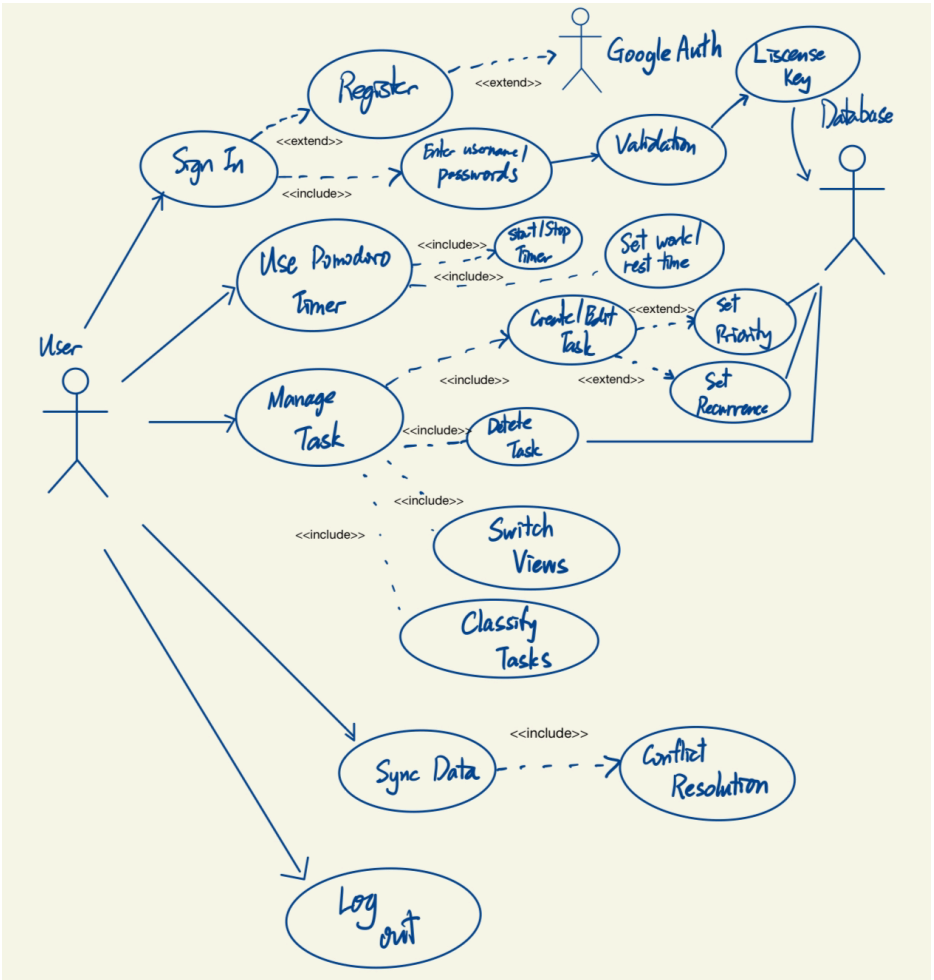
CSCI3100 course material

2 Overall Description

2.1 Product Overview

This application - TickPulse is a new, self-contained productivity management system designed to address the need for secure, cross-device task synchronization, especially for remote workers requiring cross-device synchronization feature. This product combines offline-first functionality with device-bound licensing to ensure data integrity across multiple devices. It integrates with Firebase for authentication and leverages a modern stack (React PWA + Python + MySQL) to maintain real-time synchronization while maintaining local cache resilience. The system operates within a modular ecosystem, interfacing with calendar services and academic platforms through standardized APIs, but doesn't require dependencies on external task management ecosystems.

Use Case Diagram:



2.2 Product Functionality

- User system:
    - Register/Login (Login through Google/ email authentication)
    - Individual User system
- Pomodoro Technique Timer
    - Set focus and rest times
        - Minimum 25-min focus blocks of Pomodoro Technique
- Tasks list
    - Select priority of tasks (low-green/ medium-yellow/ high-red)
    - At most 3 levels tasks
    - Classification of tasks (study, work, personal, health etc.)
    - Shown in different ways: Board, Calendar and List
    - Able to set repeat tasks that repeats under a certain cycle (daily/ weekly/ monthly)
    - Data will be synchronize during internet connection

2.3 Design and Implementation Constraints

- For technical constraint, about language support, i18next(https://www.i18next.com/) will be used to provide English, Simplified Chinese, Traditional Chinese options, it must load translations async via CDN.
- About the architecture, COMET long-polling and event-driven hybrid architecture with socket.io (version 4.7+ with EIO=4 transport) (https://socket.io/docs/v4/client-api/) will be used.
- Indexed DB(https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API) will be used to synchronize the local data to the cloud/server (local operation is prioritized).
- If data is being updated in more than one device, Last-Write-Win policy will be used in order to resolve conflicts. It must be in ISO 8601 UTC format.
- Firebase would be used for user authentication.
- For technologies being used, React 18 will be used for frontend development. MySQL adn Firebase will be used for database management. Python will be used for backend development. Git Flow will be used to maintain the branch.

2.4 Assumptions and Dependencies

- User devices should support modern browsers, e.g. Chrome 90+.
- Firebase free-tier quotas have to be sufficient for testing during development.
- The device clock error should be as minimum as possible ($\leq$ 5 mins), better follow the NTP protocol.
- Users won't tamper with local storage.
- Optimal end user screen resolution will be 1920X1080 (16:9).
- User devices should have $\leq$ 200ms latency to Firebase servers.
- The maximum concurrent tasks per user will be 500.
- Only able to show the last 30 days of the list history.

3 High-level Software Architecture

Client-server model is adapted for this application.

Component	Role	In our application
Client	Initiate request, provide UI and handle user interaction	Users interact with our application in web browsers
Server	Process request, Manage the request logic	Server validate request, execute logic, retrieve data from database
Database	Store and retrieve persistent data	MySQL database for storage of user information

4 Functional Requirements

4.1 Functional Requirements

The following table summarizes the functional requirements for the TMS.The following table summarizes the functional requirements for the TMS.

Req. No	**Title**	**Description**
R1	User Management	
R1.1	User Sign up	Allow new user to create account
R1.2	User Login	Users must log in using valid credentials before having access to the TMS
R1.3	User Logout	User can securely log out when finished
R1.4	Google Register/Login	User can use google account to register and login
R2	User Operation	
R2.1	Task Management	Users can manage the task list such as add new tasks, delete existing tasks, and arrange the task list.
R2.2	Timer setting	Set focus and rest time
R3	User Interface	A clear user interface with menu and button
R4	Task List	
R4.1	Visualize Task List	Show the task list in different way such as List, Board and Calendar
R4.2	Repeat task - Habits	Able to set the repeat tasks that repeat under a certain cycle(daily, weekly, monthly)
R4.3	Notification	Send a message to user email when the task starting time or ending time is returned
R5	Licence Management	User need to enter valid licence key before accessing to the system

4.2 Actors

An actor is anything that exchanges data with the system. An actor can be a user, external hardware, or another system. The following are identified actors on the system.

4.2.1 End User

The end user with valid credentials can access the system and use the basic features of the system.

4.3 Use Case Model

A use case defines a set of use-case instances, where each instance is a sequence of actions a system performs that yields an observable result of value to a particular actor. Following are the identified use cases.

4.3.1 Use Case #1: Sign up

Initiator – End user

Purpose - This use case allows the end user to get the permission to access the TMS.

Requirements Traceability – R1.1

Flow of Events

1. Basic Flow

   This use case begins when the actor tries to sign up on the TMS web site through entering his/her personal information such as Email, user ID and Password. The system will store the information and create an account for the end user.

2. Exceptions

   E1:Invalid information
   An invalid email is entered. The user may either re-enter email or terminate the use case.

4.3.2 Use Case #2: Login

Initiator – End user

Purpose - This use case enables actors to login to TMS in order to access the application and determine their access right

Requirements Traceability – R1.2

Preconditions - The end user successfully signs up before trying to login.

Flow of Events

1. Basic Flow

   This use case begins when the actor attempts to login the TMS web site through entering his/her User ID, Password and licence key. The system verifies that the password is valid and allows the actor to login the system.

2. Alternative Flow

   This use case begins when the actor attempts to login with Google if the actor's account is linked with his/her Google account.

3. Exceptions

   E1:Invalid Login Information
   An invalid User ID , Password or licence key is entered. The user may either re-enter the User ID and/or Password, or terminate the use case.

4.3.3 Use Case #3: Task Management

Initiator – End user

Purpose - This use case allows users to create and manage a list of tasks including addition, deletion and modification.

Requirements Traceability – R2.1

Preconditions - The end user successfully login the TMS.

Flow of Events

1. Basic Flow

   The end user can add/delete/update the task in the TMS

   If the activity selected is Add, the A-1: Add task alternative flow is performed.

   If the activity selected is Delete, the A-2: Delete task alternative flow is performed.

   If the activity selected is Update, the A-3: Update task alternative flow is performed.

2. Alternative Flow

   A-1: Create Task

The system displays the task creation screen. The user enters and saves the task information such as task name, priority of tasks, starting time and ending time[E-1], then the use case can restart or terminate.

A-2: Delete Task
The user deletes the specific task, then the use case can restart or terminate

A-3: Update Task
The system displays the task screen containing information for the specific task. The user updates and saves the information [E-1], then the use case can restart or terminate.

3. Exceptions

E-1: Invalid Time The ending time is before the starting time. The user may re-enter the information, restart, or terminate the use case.

4.3.4 Use Case #4: Timer Setting

Initiator – End user

Purpose - This use case allows users to set “Focus Time” and “Rest Time”. This provides clear visual effects to remind users to focus on doing the tasks while maintaining the balance between working and taking rest.

Requirements Traceability – R2.2

Preconditions - The end user successfully login the TMS.

Flow of Events

1. Basic Flow

The end user can set up the timer with “Focus Time” and “Rest Time”. The system will ring when the timer runs out of time.

5 Other Non-functional Requirements

5.1 Performance Requirements

P1: The system shall load the UI within 3 seconds
P2: Task reminder should be sent 5 minutes before the scheduled time.

5.2 Safety and Security Requirements

- All user data shall be encrypted in transit (https).
- Passwords shall be hashed and securely stored in the database.

5.3 Software Quality Attributes

5.3.1 Adaptability

The system should be able to support major decent browsers, including Firefox and Chrome.

5.3.2 Reliability

The system should have at least 95% uptime so as to ensure minimal downtimes for users.