



## **Homework # 7**

**13016215 Software Engineering Principle Laboratory**

**Software Engineering Program**

**Faculty of Engineering, KMITL**

By

65011277 Chanasorn Howattnakulphong

65011320 Kanokjan Singhsuwan

65011381 Napatr Sapprasert

## User Functional requirement:

1. The system shall allow users to create accounts with a unique username and password.
2. The system shall enable the users to define and customize habits they want to develop, including the name and the difficulty of the tasks.
3. The system shall allow users to log their daily progress for each habit, indicating successful completions or any deviations, along with keeping count of the number of times the task is completed.
4. The system shall provide users with visual representations of their habit progress, represented by virtual trees growing as the user completes their tasks, and withering as the user fails to complete their tasks.
5. The system shall enable users to set events on a calendar to save them.
6. The system shall enable users to customize custom notifications based on the events saved on the calendar and to notify the users accordingly.
7. The system shall give users access to insights and analytics about their habit performance, identifying trends or patterns.
8. The system shall allow users to be able to view historical data for their habits, facilitating self-reflection and improvement.

9. The system shall allow the user to share their progress online.
10. The system should support data backup and synchronization across multiple devices to ensure a seamless user experience.

#### User Non-Functional requirement:

1. The system shall have a user-friendly interface to facilitate easy navigation and habit tracking. Users should be able to fluently use the app within 20 minutes.
2. The system shall respond promptly to user interactions, minimizing load times and delays. The system should respond within 5 seconds of the user input.
3. The system shall have robust security, protecting data against unauthorized access.
4. The system shall provide the UI in the US English language.
5. The system shall be able to run on a PC with at least 8GB of ram
6. The system shall be able to run on different operating systems including Mac, Windows, and Linux.
7. The system must have a notification when the system has an error
8. The system shall provide a tutorial pop-up window.

9. The system shall provide dark mode to match the users' preferences.
10. The system shall be able to run with less than 5% crashes on typical uses.

### System Functional Requirements:

1. The system must authenticate users securely during the registration and login processes by checking the input information with the credentials in the database
2. The system shall display the habit-related data by retrieving data from the database
3. The system must manage and do error handling and send notifications to the admin when it occurs.
4. The system shall store and provide to create, retrieve, update, and delete operations on database data.
5. The system shall require re-enter of the password if it is incorrect, if there are more than 5 incorrect attempts, a notification will be sent to the admin.
6. The system shall provide a timestamp of each complete event or activity.
7. The system shall provide to delete LOGS records when log exceeds 12 months.

8. The system must be able to reset the daily tasks at midnight using local device time.

9. The system shall notify the user with pop-up notifications according to calendar events set, following local time.

10. The system shall update displayed data in real-time according to the changes in database

### System Non-Functional Requirements:

1. The system must be developed with an object-oriented approach.

2. The system must be developed using the ZODB non-sql database for an object-oriented approach.

3. The system must be developed using Python4, with Pyside6 QtWidgets libraries for an object-oriented approach.

4. The system must be developed by using 1920x1080 or 16:9 resolution.

5. The system must be version-controlled under git.

6. The system must be developed co-operatively using GitHub desktop, as it is a free, functional, and efficient GUI for Git.

7. The system must be available to users 24/7.

8. The system must be developed following the PEP8 Code formatting to maintain the code standard throughout the project.

9. The system database should roll back in case of data failure.

10. The system shall encrypt sensitive data such as passwords using public key encryption