**Project on My Habit Track**

**Conception phase**

Habit tracking is a valuable practice for individuals seeking to cultivate positive behaviors and routines in their daily lives. This project presents a habit tracking application implemented using the tkinter library in Python, offering a user-friendly graphical interface. The application allows users to seamlessly add and remove habits, providing a dynamic and interactive tool for habit management. This project offers a foundation for developing more sophisticated habit-tracking solutions and encourages users to cultivate a more intentional and mindful lifestyle. The Habit Tracking application developed in Python using the Tkinter library serves as a user-friendly graphical interface for habit monitoring and recording. Users can input daily habits, and the application records each completion, displaying a chronological list within the interface. The program employs Tkinter widgets, including Entry, Button, and Text, to create an interactive experience. Through straightforward design and functionality, this application provides a simple yet effective tool for users to track and visualize their habits over time.

**INTRODUCTION**

The habit tracking software is made to assist users in efficiently recording and managing their habits, enabling them to make healthy adjustments and realise their individual goals. In today's fast-paced society, it is simple to become mired in daily responsibilities and forget our ideal behaviours. Our habits play a significant influence in determining how our lives turn out, whether it's through regular exercise, increased reading, or mindfulness exercises. The habit tracking app's goal is to give users a quick and easy way to keep track of and improve their habits. Users can obtain important insights into their behaviour, spot patterns, and make wise decisions to create routines that are healthier and more productive by keeping a record of their progress.

# Overview of the Solution

1. **Work Habits class**: In this class we are adding the habit and deleting the habit
2. **Clocked Status class**: Here we creted update\_clockedhabit\_panel method. This method updates the clocked habit information panel when the user selects a new session. The function observes the selected\_session paremeter and if it has changed, it updates the info to the newly selected session. The method waits 0.5 seconds and calls itself again, repeating the process previously describe
3. **Data class**: Here we active streak and selecting the days
4. **General class**: Finding the highest and lowest observed streaks in user\_habits
5. **Clocked Session class:** Listing all fields to display for any specific clocked sesission.
6. **Database class:** Here we load the json file and upadate data
7. **Habit class:** In this class we control the stopwatch, starts the stopwatch stops the stopwatch and loads the wok\_habits class in new window. The work \_habit class manages the cloacked habits defined by the user and performing submission of new habit, checking status of habits, checking status of clocked habits.
8. **Status:** In this class we created update general panel method. This method updates the general information panel when the user selects a new frequency.

# 1.2 User Interaction Flow

The Habit Tracker application provides a user-friendly interface for creating new habits, completing tasks, and tracking progress. The overall user experience and interaction with the application can be described as follows:

**Launching the Application**

* The user opens the application by running the main.py file.
* Then main window of the application appears, we have to login then providing options for creating new habits or viewing existing habits.
* Displays the habits pending for every specific day of the week the user is using the app. Clicking on the “Done” button will register one completion cycle for the habit. Each specific habit needs to be completed as many times in a day as the pre-defined daily goal. Once the user has hit the daily goal for any specific habit, the “Done” button for that habit will change to “Completed” and will be disabled. To visualize the data and insights gathered click the “H STAST!” button.

1. Creating a New Habit:

If the user choose to create a new habit:

* Type the new habit and frequency from drop down(e.g., daily, weekly, monthly) along with we have to select the days and actions like moring, evening etc.. from initial window and click on Add button to create the habit.
* After creating new habit you got successfully added your habit popup message.

1. Viewing Habits:

If the user selects the " Manage My Habbits" button:

* This panel displays the ALL habits defined by the user. The user can check all parameters for every single active habit. The parameter displayed are: Time, Habit Name, Daily Goal, and Frequency. Each habit listed has a “Delete” button. If clicked, the user will permanently delete all data associated with the selected habit.

1. Set Habit:

* In this page we can add the habit, frequency and also delete the habit. We can check the added habit in Manage My Habit page.

1. Clocked Habits Panel:

* This panel is used to track habits within a timed session. The user can user the “Set Habit” button to define new clocked habits. To start a session, click on the “Start Session” button. Ness will push a notification every time the stopwatch hits the targeted frequency for each active habit. To user can then select to either skip the habit or mark it as completed. To end the session, click the “End Session” button. To visualize the data gathered from the last 10 sessions, click the “C STAST!” button Habit Satus:

1. Habit Status:

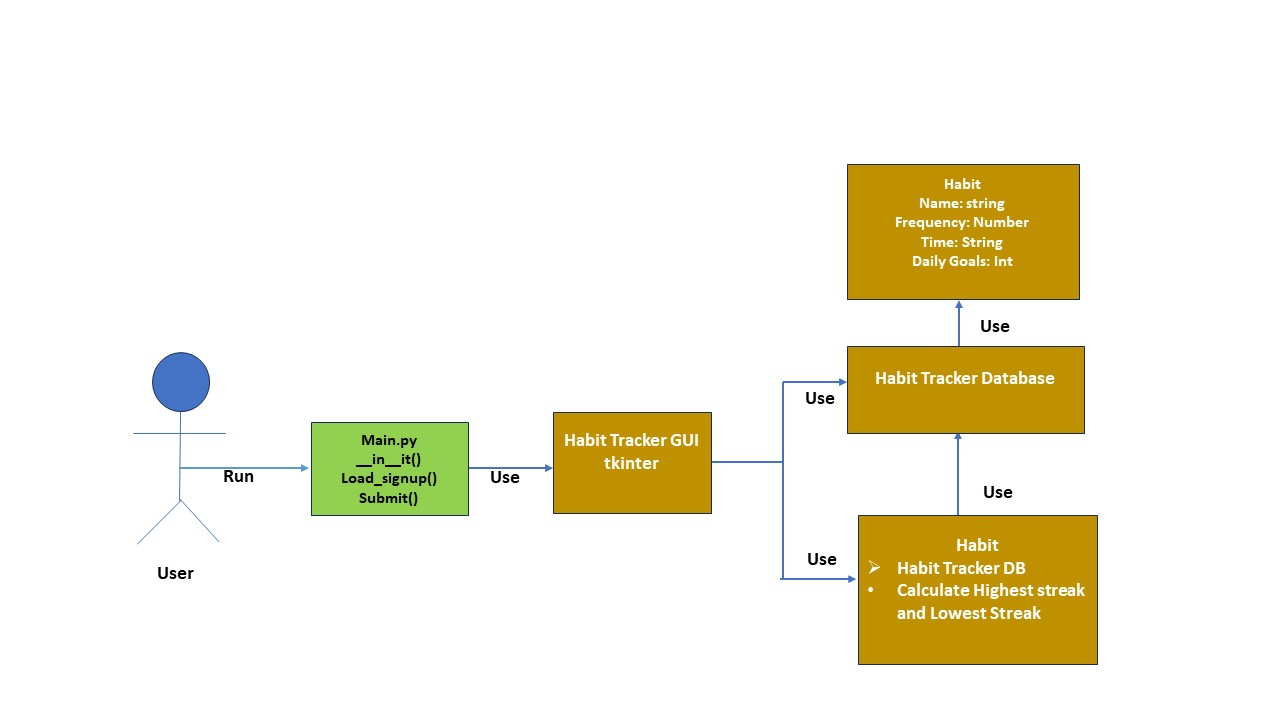
* Displaying Highest streak habit and lowest streak habit result. Here we comparing the streak of two habits.

1. Clocked Habit Satus:

* Here we find the Session Data means displaying habit name, frequency, target goal, achieved goal.

# Component Interaction and Communication

The interaction between the different components of the Habit Tracker application is illustrated using a component diagram. Figure 1 showcases the high-level components and their relationships, depicting how they communicate and exchange data. The communication and data flow of habit tracker is listed below:

 Figure 1.1: Component Diagram of Habit Tracker

1. The user opens the application by running the main.py file.
2. Main.py having methods like \_\_init\_\_(), Load\_signup(), submit() and also importing the packages here.
3. Here we use python library called tkinter used for creating graphical user interfaces (GUIs). It provides a set of tools and widgets for building desktop applications with graphical elements such as windows, buttons, text boxes, labels, and more.
4. Habit tracker database storing name of the habit and frequency, time etc….with the help of this database we access the result (Highest steak and lowest streak).

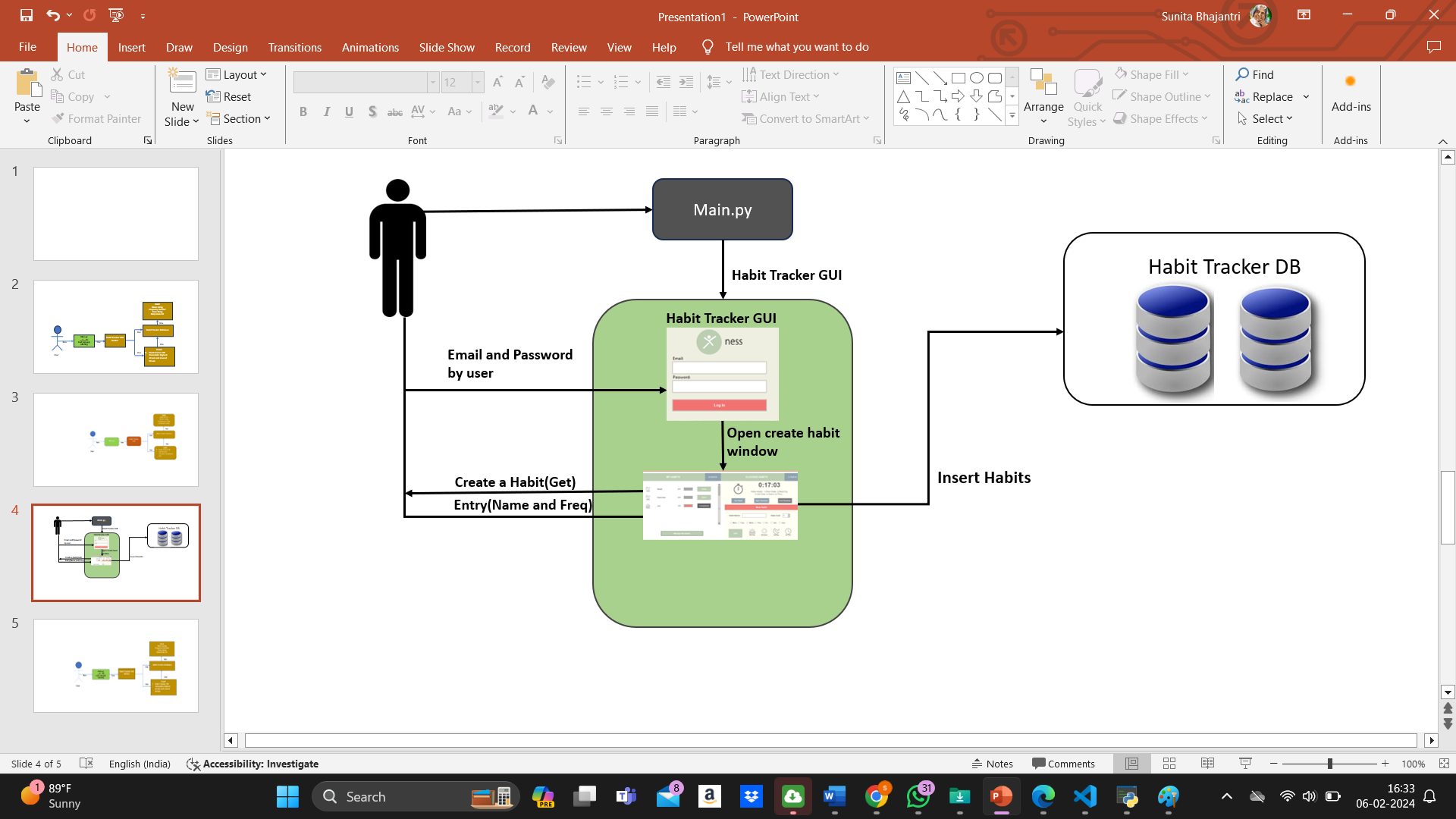


Figure 1.2: Component interaction for creating a new habit

1. After running the main.py file the application open with login page. Here we providing user credential like email and password and enters into the main page.
2. The user can add the new habit by providing habit anme, frequency, time and days and click on the add button. The newly created habit is goes and stored inside the database
3. The user can select the "Manage my Habits" button to view all created habits. “Habit” class communicates with “Database” to retrieve all habit data.
4. “Habit class” utilizes “Database” to fetch habit records, calculate completion rate and compare completion rates with other habits.
5. In this project we use clock why because to get the alert. Example when we set frequency of Drinking water as 1min after completion of 1 min it will give the pop like you should Drink Water now.
6. And also we use two buttons like Habit status for checking status of the habit and clock status for checking status of the clock.

# CONCLUSION

# Users can chart their behaviours and track their advancement over time using the Habit Tracker program in a straightforward and efficient manner. Users can easily download and start the application, create new habits, inspect current habits, and analyse habit analytics. In this application, we have created the application for Habit management. Users can effortlessly download and initiate the application, establish new routines, inspect current habits, and analyze habit status.

A user-friendly graphical user interface (GUI) developed with Tkinter, a robust database system, and the ability to calculate habit analytics such as completion rate, shortest streak, and longest streak are crucial features of the Habit Tracker application. These functionalities enable users to gain deeper insights into their habits, empowering them to make informed decisions aimed at improving their daily routines.The application's functionality is well-supported by the selection of Python as the programming language, database as json file, and Tkinter as the GUI library. The application's functionality is effectively bolstered by the choice of Python as the programming language, utilizing a JSON file as the database, and employing Tkinter as the GUI framework.