

---

## *Conception Phase for Habit Tracking Application*

---

### **1. Introduction:**

As the lead developer of a habit tracking application, the goal is to build a system that allows users to create, track, and manage their habits efficiently. The application should allow users to add new habits, mark tasks as completed, and monitor their progress over time.

In this concept phase, the primary focus is on defining the essential components, user interaction flow, and data management of the application. The aim is to lay a solid technical foundation for the project.

---

### **2. Core Components:**

#### **a. Habit Class:**

- This is the central entity in the application. Each habit has attributes like:
  - **Name:** The name of the habit (e.g., "Exercise", "Reading").
  - **Periodicity:** How often the habit needs to be completed (daily, weekly, etc.).
  - **Current Streak:** The number of consecutive days/weeks the habit has been completed.
  - **Longest Streak:** The longest streak of successful completions.
  - **Created At:** Timestamp when the habit was created.
  - **Completed Dates:** List of dates when the habit was marked as completed.
  - **Is Broken:** A boolean indicating whether the streak has been broken.

#### **b. Habit Tracker Class:**

- Manages a collection of habits.
- Responsible for creating, deleting, and retrieving habits.
- Handles the logic for marking habits as complete and updating streaks.
- Stores habit data in an organized way (e.g., using a list or dictionary).
- Retrieves habits from storage (database or in-memory).

#### **c. Data Storage:**

- Habit data can be stored in a **database** (like SQLite for small applications or PostgreSQL for larger scale).
- For simplicity, in the initial prototype, we can use **JSON** files to store habit data persistently.

- The structure could involve saving each habit object as a JSON entry with its attributes (name, periodicity, current streak, etc.).
  - Data retrieval would involve reading from this file to load habits into memory.
- 

### 3. Features of the Habit Tracking Application:

#### 1. Add Habit

- **Description:** Users can add new habits by providing a name and setting the periodicity (either `daily` or `weekly`).
- **Implementation:**
  - Input fields for habit name and periodicity.
  - Once the habit is added, it is saved to the system (in-memory or database/JSON).

#### 2. Complete Habit

- **Description:** Users can mark habits as completed. The habit's streak will be updated accordingly, and the app ensures that the data remains consistent.
- **Implementation:**
  - On selecting a habit, users mark it as completed.
  - Updates the current streak and checks if the streak is broken.
  - Saves completion date.

#### 3. View Habits (Normal and Periodicity-Wise)

- **Description:** View a list of all habits or filter them by their periodicity (either `daily` or `weekly`).
- **Implementation:**
  - Display a list of habits with options to filter by periodicity.
  - Each habit displays relevant data (e.g., name, current streak, longest streak).

#### 4. Track Streaks

- **Description:** Track and monitor both the current and the longest streaks for each habit.
- **Implementation:**
  - Display current streak and longest streak for each habit.
  - Track streaks in the system, updating after each completion.

#### 5. Persistent Storage

- **Description:** Save all habit data (habits, streaks, completed dates) to a JSON file, ensuring persistence across app sessions.
- **Implementation:**
  - Habit data is stored in a JSON file (or database in future stages).
  - Data is read and written to the file when habits are added, marked as completed, or updated.

#### 6. Delete Habit

- **Description:** Users can delete a habit from the app, and it will be removed from the JSON file as well.
- **Implementation:**
  - Add a delete option in the habit view.
  - Once deleted, the habit data will be removed from the storage (JSON file or database).

---

## 4. Technical Considerations:

### a. Storage Solution:

- The initial version of the app can use in-memory storage for simplicity, but we should consider expanding to a more robust solution like SQLite for persistent storage, allowing the app to store user data between sessions.

### b. Communication between Components:

- The Habit Tracker is responsible for interacting with the data storage layer (whether it be in-memory or database).
  - The Habit class is independent and simply stores information about each habit. And provides methods for marking complete, creating a **Shallow Copy** of object and check for consistency of habits completion
  - Interaction between the user interface (UI) and the Habit Tracker will be handled through API calls, enabling the CLI to display habit data and allow users to add or mark habits as complete and etc
- 

## 5. User Interaction Flow:

### a. Create Habit:

- User logs in, selects "Create Habit," and inputs a habit name (e.g., "Exercise") and periodicity (e.g., "daily").
- Habit is stored in the system.

### b. Complete Habit:

- User logs in and sees a list of habits with their current streak.
- User marks a habit as completed. The system records the completion and updates the streaks accordingly.

### c. View Habit Progress:

- User selects a habit to view its details.
- The system displays the current streak, longest streak, and a list of completed dates.

## 6. High-Level Workflow Example:

### 1. Add Habit:

- User enters "Exercise" as a new habit and sets it to `daily/weekly`.
- Habit is added and saved in the JSON file.
- 2. **Complete Habit:**
  - User marks "Exercise" as completed on `2024-12-01`.
  - The current streak is updated, and the data is stored.
- 3. **View Habits:**
  - User views their habits, filters to see daily habits.
  - "Exercise" appears with a current streak of 1.
- 4. **Track Streaks:**
  - The system displays both the current streak (1 day) and the longest streak (1 day) for the habit.
- 5. **Persistent Storage:**
  - All the data is saved in the JSON file after every habit completion and update.
- 6. **Delete Habit:**
  - User chooses to delete "Exercise".
  - The habit is removed from the list, and the JSON file is updated accordingly.

## CLASS DIAGRAM

