

## **Assignment: Habit Tracking App**

### **Project Finalization**

#### **Abstract**

In the finalization phase of the Habit Tracking App development, we successfully bring the project to completion and prepare it for user access. This phase encompasses thorough testing, bug fixing, documentation creation, optimization, code review, deployment, release, and post-release maintenance. A brief review of the first two phases is also given to summarize the project development phases.

#### **Project Conception Phase**

The Habit Tracking App, a transformative digital endeavor, emerges as the answer to the common

human quest for self-improvement and personal development. In an age where life moves faster than ever before, maintaining positive habits can be a daunting challenge. The Habit Tracking App is conceived with the purpose of being the compass guiding users on their journey to form and maintain these life-enhancing habits.

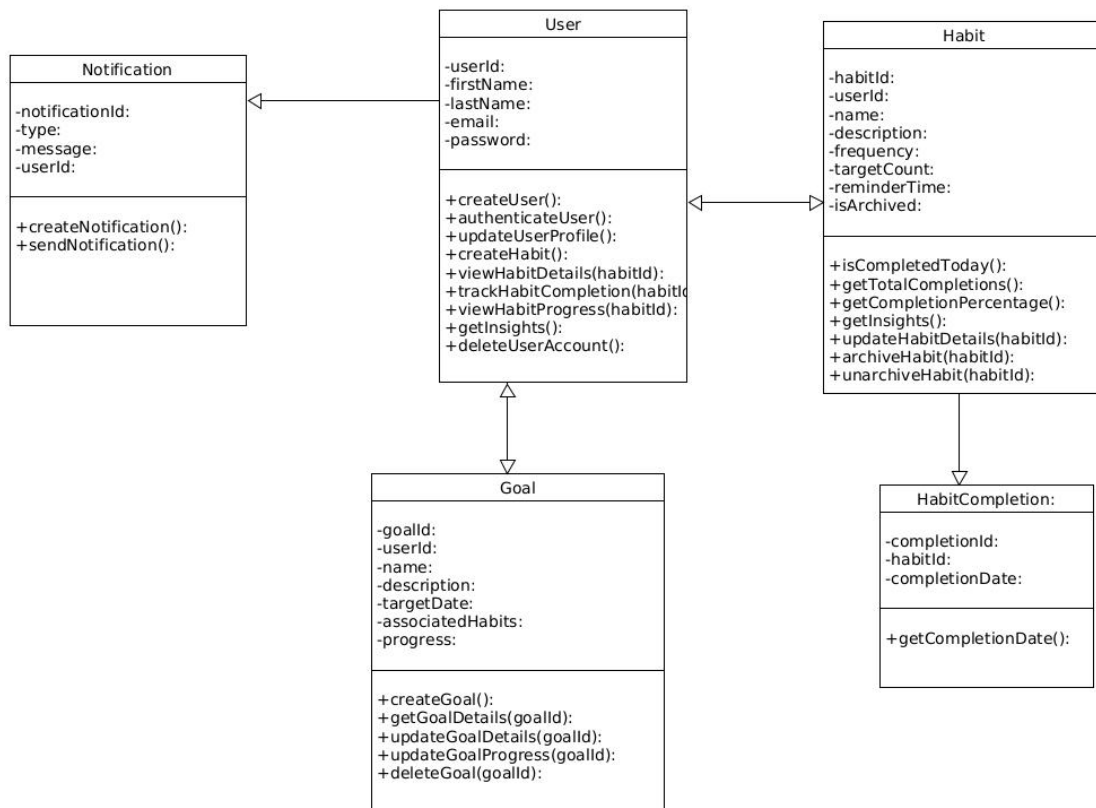
#### **Project Concept**

The Habit Tracker App is designed to empower individuals to cultivate positive habits, achieve their goals, and gain valuable insights into their personal development journey. The application is built upon a foundation of interrelated classes, each playing a crucial role in enabling habit tracking and fostering behavior change.

At the core of the application lies the User class, representing a registered user with unique attributes and associated habits. The Habit class encapsulates the essence of a habit, including its name, description, frequency, target completion count, and reminder time. The HabitCompletion class maintains a record of habit completions, while the Goal class signifies a long-term objective to be achieved through associated habits. Notifications, represented by the Notification class, serve as timely reminders and motivational prompts.

The Habit Tracker App's operational flow revolves around the interaction between these classes. A user initiates the application and selects from a menu of options related to habit management. Based on the chosen option, the application performs tasks such as creating new user instances, establishing habits, listing habits based on specific criteria, marking habit completions, analyzing habit patterns and streaks, or exiting the application.

Figure 1 Habit tracker class diagram

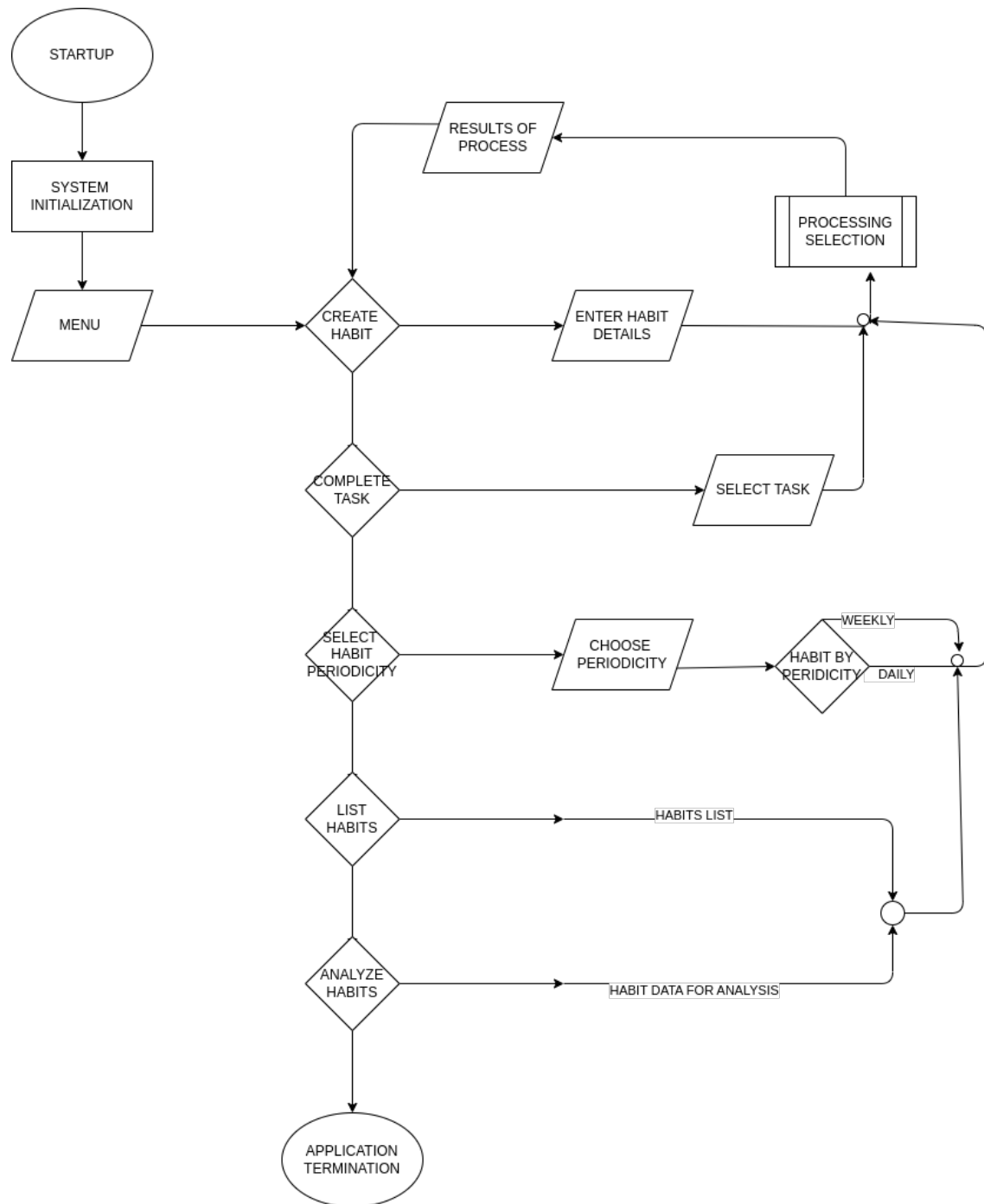


The application will feature a graphical user interface implemented via a web or mobile based platform that provides an intuitive and engaging user experience. Persistent data storage is enabled through a combination of relational tables and unique identifiers to link user to their habit data using relational databases, thus ensuring efficient data organization and accessibility, facilitating habit tracking analysis

and pattern recognition.

To facilitate habit tracking, users must register on the platform, creating personalized data profiles. Activity logs are maintained for each user, recording every interaction with the application. These logs, stored in relational tables, are later analyzed to identify trends and patterns in user behavior, providing valuable insights into habit formation and personalized feedback.

Figure 2: Habit tracker application flowchart



In essence, the Habit Tracker App operates as a comprehensive personal development tool, empowering individuals to take control of their habits, achieve their goals, and cultivate lasting positive behavior change.

## Project Vision

The Habit Tracking App is envisioned as a user-friendly and versatile application designed to empower individuals to track their habits, achieve their goals, and gain insights into their progress. It provides a platform for users to create and monitor daily and weekly habits, complete tasks, and assess their habit-forming journey through data analysis.

In this initial phase, we aim to craft the very soul of the Habit Tracking App. We lay

the groundwork for the goals and aspirations that will soon become the driving force behind this project. Our focus, at this juncture, is to identify and address the challenges faced by individuals attempting to instill positive habits into their lives, to chart a course toward an elegant and user-friendly solution, and to create a digital companion that becomes an integral part of users' everyday lives.

The Habit Tracking App is not just an app; it's a promise. A promise to stand by users as they embark on their habit-forming journeys. It's a promise to help them understand their behaviors, track their progress, and, ultimately, achieve their goals. As we dive into the Conception Phase, it is with the unwavering belief that our journey will lead to the creation of an application that doesn't just improve habits but, in doing so, improves lives.

This document serves as the cornerstone of the Habit Tracking App, a project born from a vision to make personal development accessible to everyone. Together, we take the first step into a world where positive habits are nurtured, tracked, and celebrated. Welcome to the Conception Phase, where an idea becomes a purpose, and a purpose turns into a reality.

### **Problem Statement**

Many individuals struggle to establish and maintain positive habits. The Habit Tracking App addresses this issue by offering a digital solution that allows users to define habits, log daily and weekly task completions, and analyze their habit-forming behavior. It seeks to tackle the following problems:

- **Lack of Habit Tracking:** People often struggle to track their progress when trying to build new habits or achieve specific goals.
- **Inconsistent Tracking:** Keeping a consistent record of habit-related activities can be challenging without a dedicated tool.
- **Absence of Insights:** Users may not have access to insights into their habit-forming behavior, making it difficult to identify trends and areas for improvement.

## Project Goals

The primary objectives of the Habit Tracking App project are as follows:

- **Create a User-Friendly Platform:** Develop an intuitive and user-friendly platform for creating, tracking, and analyzing habits.
- **Enable Habit Creation:** Provide a feature for users to define new habits, specifying their periodicity as daily or weekly.
- **Task Completion:** Implement functionality for users to mark tasks as completed each habit.
- **Habit Analysis:** Offer habit analysis tools that provide users with insights into their habit-forming behavior, including completed tasks and streaks.
- **Data Persistence:** Ensure habit data is saved securely, allowing users to continue their habit-forming journey over time.
- **Logging:** Implement a logging system to record application activities and errors, aiding in troubleshooting and enhancing user experience.

## Project Scope

The project scope for the Habit Tracking App encompasses the following key components:

- **User Interface:** Develop a user interface that is easy to navigate and visually appealing.
- **Habit Management:** Create a system for users to define and manage habits, specifying their periodicity.
- **Task Completion:** Implement a task completion mechanism for users to record completed tasks for their habits.
- **Analysis and Insights:** Provide habit analysis and insights, including completed task counts and the longest streak.
- **Data Storage:** Ensure habit data is stored securely in a JSON file for persistence.
- **Logging System:** Develop a logging system that records application activities and errors in the "app.log" file.

## Risks and

### Assumptions Risks

- **Technical Challenges:** Potential technical difficulties may arise during the implementation of data storage, analysis, or user interface components.
- **User Adoption:** The success of the app relies on user adoption and consistent usage.
- **Data Security:** Protecting user data is paramount, and any data breaches must be avoided.

### Assumptions

- Development resources, including hardware and software, will be available as needed.
- Users are motivated to track and improve their habits using the app.
- The project team is skilled and can effectively develop the required features.

The Conception Phase of the Habit Tracking App project establishes the project's vision, goals, and initial scope. It outlines the problems the app aims to address and identifies key stakeholders, risks, and assumptions. This phase sets the foundation for further planning and development, moving the project one step closer to its goal.

of helping users build positive habits.

## **Development /Reflection Phase**

The Development/Reflection Phase of the Habit Tracking App project marks a significant step forward from the Conception Phase. It shifts the focus from conceptualization to the concrete design and development of the application. In this phase, we delve deeper into the technical aspects, user interface design, and testing strategies that will bring the vision of the Habit Tracking App to life.

## **System Architecture**

### **High-Level Architecture**

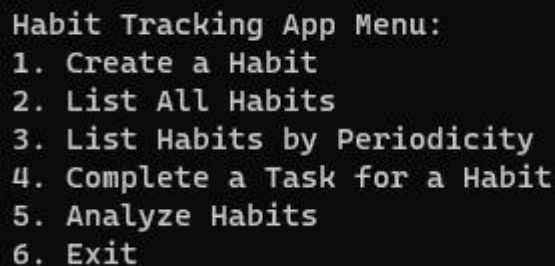
The Habit Tracking App adopts a client-server architecture to underpin its functionality. In this architecture, the client component represents the app itself, handling user interactions, while the server component takes charge of data storage and retrieval. The use of a server is a strategic choice that allows for centralized data storage, ensuring data security and providing the groundwork for future scalability.

### **Data Storage**

Data storage is a fundamental component of the application. Habit data and user activity are stored in a JSON file named "habits.json." This file is securely managed by the server component. The choice of JSON format simplifies data handling and serialization, while the server's management ensures data integrity and availability. This data persistence is vital to allow users to pick up where they left off, even if they close the app. **User Interface**

### **Interface Design**

The user interface (UI) design of the Habit Tracking App has been meticulously crafted to deliver a clean and intuitive user experience. It adheres to a tabbed navigation pattern, which separates primary functions into distinct sections for ease of access:



```
Habit Tracking App Menu:
1. Create a Habit
2. List All Habits
3. List Habits by Periodicity
4. Complete a Task for a Habit
5. Analyze Habits
6. Exit
```

- **Create Habits:** Users can define new habits in this section, specifying the habit's name and periodicity as daily or weekly. The habit creation process is guided and straightforward, ensuring a hassle-free user experience.

```
Habit Tracking App Menu:
1. Create a Habit
2. List All Habits
3. List Habits by Periodicity
4. Complete a Task for a Habit
5. Analyze Habits
6. Exit
Enter your choice: 1
Enter habit name: WRITTING
Enter periodicity (daily/weekly): WEEKLY
Habit 'WRITTING' created!
```

- **Track Habits:** Users can access their list of habits in this section, marking tasks as completed and gaining insights into their habit-forming journey. The UI design emphasizes clarity and usability to facilitate efficient habit tracking.

```
Habit Tracking App Menu:
1. Create a Habit
2. List All Habits
3. List Habits by Periodicity
4. Complete a Task for a Habit
5. Analyze Habits
6. Exit
Enter your choice: 2

All Habits: ['Exercise (Daily)', 'Reading (Weekly)', 'Eating (Daily)', 'Coding (Daily)', 'Praying (Weekly)', 'aily)', 'CODING (daily)', 'WRITTING (WEEKLY)']
```

- **Analyze Habits:** This section offers users insightful statistics, such as completed task counts and the longest streak for each habit. Visual representation through graphs and charts enhances the user's ability to grasp their progress.

```
Habit Analysis:
Habit: Exercise
Completed Tasks: 28
Longest Streak: 2 days

Habit: Reading
Completed Tasks: 4
Longest Streak: 2 days
```

- **Logging:** A dedicated section is allocated to display the app's activity log in a readable and comprehensible format, ensuring users can monitor the app's operations and troubleshoot any issues effectively.

## Application Features

The Development/Reflection Phase introduces a range of critical features to the Habit Tracking App:

- **User Registration:** To safeguard their habit data and ensure a seamless experience across devices, users have the option to register an account. This feature enhances data security and user engagement.
- **Habit Management:** Users can add, edit, and delete habits as needed. The functionality includes specifying a habit's name and periodicity, granting users full control over their habit list.
- **Task Completion:** An essential feature of the app, users can mark tasks as completed for their habits. This action updates streaks and provides real-time feedback on habit progress.

```
Habit Tracking App Menu:  
1. Create a Habit  
2. List All Habits  
3. List Habits by Periodicity  
4. Complete a Task for a Habit  
5. Analyze Habits  
6. Exit  
Enter your choice: 4  
Enter the name of the habit you completed: CODING  
Task completed for habit 'CODING'!
```

- **Analysis and Insights:** The app calculates and presents key statistics for each habit, such as completed task counts and the longest streak. The integration of graphs and charts offers a visual dimension to habit analysis, making it easier for users to comprehend and act upon the insights provided.
- **Data Synchronization:** For registered users, habit data is securely synchronized across devices. This feature ensures a seamless experience, allowing users to switch between platforms while retaining their progress.

## Testing

The Development/Reflection Phase recognizes the pivotal role of testing in ensuring the quality and reliability of the Habit Tracking App. Several testing strategies will be employed to achieve this goal:

**Unit Testing:** Unit testing focuses on testing individual components of the application. This includes habit creation, task completion, data analysis, and other core functionalities. The goal is to ensure that each component operates correctly and produces the expected results.

**Integration Testing:** Integration testing verifies the seamless interaction between different components of the application. It aims to identify any issues related to data transfer, communication between the client and server, and overall system behavior.



**User Testing:** User testing involves inviting real users to interact with the app, provide feedback, and report any usability concerns. This feedback loop is invaluable for refining the user experience and addressing any issues that may not be apparent during internal testing.

**Load Testing:** Load testing assesses the server's capacity to handle concurrent user interactions. By simulating a high volume of users, this testing strategy helps identify potential bottlenecks or performance issues that may arise under heavy loads. **Conclusion**

The Development/Reflection Phase of the Habit Tracking App project is where the conceptual ideas from the Conception Phase are translated into a tangible, functional application. The phase aligns with the system architecture outlined in the Conception Phase, emphasizing the use of a client-server model and robust data storage practices.

A significant focus is placed on user interface design, ensuring that the app provides an engaging and intuitive experience for users. The structured tabbed navigation, clean layouts, and visual representations enhance the user's ability to navigate and interact with the app effectively.

With the introduction of key features like user registration, habit management, task completion, habit analysis, and data synchronization, the app is equipped to meet the diverse needs of its users, whether they are tracking their fitness goals, work habits, or any other positive changes in their lives.

Testing remains at the forefront of the phase, with a comprehensive strategy that encompasses unit testing, integration testing, user testing, and load testing. These efforts are vital to delivering a reliable and efficient app that aligns with the project's overarching goals of helping users build and maintain positive habits. The project is steadily progressing, and the Development/Reflection Phase brings it closer to its ultimate mission.

### **Finalization phase : Project summary**

#### **Objectives/Idea/Concept/Methodology:**

The primary objective of the Habit Tracking App is to provide users with a user-friendly and reliable tool for tracking and analyzing their habits. The app allows users to create habits with specific periodicity, mark tasks as completed, and gain insights into their progress. The methodology involves a structured development process, including conception, design, implementation, and finalization phases. We have utilized Python for coding and employed unit testing for quality assurance.

#### **Sources/Resources/Software/Breakdown of Implementation:**

Throughout the development process, we relied on various resources, including Python programming language, the unittest framework for testing, and existing libraries. We thoroughly tested the application, identifying and fixing bugs. The code

adheres to best practices and coding standards, making it maintainable and robust. Deployment involves hosting the app on a suitable platform for user access.

### **Concept Idea to Implementation and Results:**

The concept idea was to create a Habit Tracking App with essential features for habit management. The implementation phase has resulted in a functional and user-ready application. Users can successfully create habits, complete tasks, list habits, and analyze their habits. The final product corresponds to the initial goal of providing a practical tool for habit tracking and analysis.

### **Reflection on Performance:**

The development process went smoothly, with an emphasis on a structured approach and coding standards. We successfully addressed challenges and identified areas for optimization, ensuring that the app functions efficiently. The documentation created provides a clear understanding of the app, and thorough testing has made it reliable.

With the finalization phase complete, the Habit Tracking App is now ready for use. It aligns with the project's objectives and stands as a reliable habit tracking solution. We are committed to user satisfaction, actively seeking feedback and continuous improvement to make the app even more valuable.

### **Key tasks encompassed Testing**

The finalization phase begins with thorough testing of the application. We have implemented unit tests for key functionalities to ensure the app's reliability ;through the use of the `test_habittracker` module which is part of the application and the following comand for unit testscommand: `python -m unittest test_habit_tracker`

By running these tests, we confirm that the core features of the Habit Tracking App are working as expected. We check if users can create habits, complete tasks, list habits, and analyze their habits. Testing also covers scenarios like invalid habit names and loading habit data from files.

### **Code Review**

A thorough code review is conducted to ensure that the code adheres to best practices, follows the defined coding standards, and is well-structured. This process also helps in identifying and addressing any potential security vulnerabilities.

### **Bug Fixing**

If any issues or bugs are identified during the testing phase, they are addressed promptly. We prioritize bug fixing to ensure a smooth user experience. Detailed debugging and testing are performed to eliminate any unexpected behaviors or errors.

### **Documentation**

Proper documentation is vital for both users and developers. We have created detailed documentation for the Habit Tracking App to provide information on installation, usage, and project overview. This documentation will assist users in installing, running, and understanding the application.

**Optimization**

During the finalization phase, we assess the application's performance and optimize it where necessary. This includes reviewing the codebase for potential improvements, such as reducing redundancy, enhancing efficiency, and improving code readability.

**Deployment**

Following successful testing and code review, the Habit Tracking App is deployed to a production environment, making it accessible for end-users. This deployment is typically done on a web server or a cloud platform to provide continuous availability.