Habit Tracker – Finalization Phase

The Habit Tracker is an application that an individual can use to track behaviours that they would like to turn into habits. The application was written in Python and uses an SQLite3 database to store habit data on a user's computer. Features of the app include creating new daily or weekly habits, marking a habit as complete, and listing habits. The app also allows users to see their longest streak for a habit and the longest streak of all their habits. To use the app, a user downloads the directory containing the app files onto their computer and opens and executes the app's functionality using a shell scripting application such as Terminal or Command Prompt.

The making of the app was a smooth process overall and a process that increased the knowledge of the developers. Given that the development team has a foundational understanding of Python, the goal was to create a bare-bones habit tracker that performs all the functionality required by the customer and not focus on creating a beautiful graphical user interface. In the design phase, the first challenge was that the team had not used object-oriented programming and functional programming in Python in the past. Since the customer required that the app be created using object-oriented and functional programming, the team had to learn about how to implement object-oriented and functional programming in Python. ChatGPT was also used to get ideas for how to design the application with these requirements in mind. The ideas formed from ChatGPT became the basis for the original design of the app. Designing the basic building blocks of the app turned out to be relatively simple. However, fine-tuning the design to meet all the customer's requirements was an iterative process, and testing the application revealed design flaws that had to be resolved. Two issues found when testing the app had to do with the display of information. When a user wanted to see a list of all their habits, the predefined and user-created habits were not separated. Additionally, when the app displayed the longest streak of all habits, it would specify the longest streak but not the habit associated with it, which was information the team thought the user would want to know. Another issue encountered with the initial design of the app was that it was not designed to be modular and maintainable. The original design had the entire application code in a single file, but this was changed so that the code was separated into two files: one that creates the habit instances and one that focuses on executing the functionality of the app.

The development phase had more challenges than the design phase. One of the challenges was that the application has quite an interconnected design, which meant the team sometimes had to go through a long trail of interconnections between classes and functions to debug problems. Another challenge was getting the correct logic for calculating streaks. The logic proved to be more complex than expected, and initially, the longest streak was not calculated for consecutive days or weeks. Coming up with the logic to see if a completion took place in the same week or a different week took some time. Besides interconnections in the design and calculating streaks correctly, the team lacked expertise with object-oriented programming (OOP)in Python, which resulted in the developers having to learn and research how to implement OOP using Python. Additionally, the team felt that using functional programming to calculate streak statistics was not the simplest or most intuitive way to calculate the statistics. However, since using functional programming for the statistics was a requirement, the team needed to learn about lambda(), map(), filter(), and max() functions. Apart from a lack of knowledge of OOP and functional programming, it was the team's first time using SQLite3. They had to learn how to create an SQLite3 database and how to visualize the data in the database for testing purposes. SQLite3 turned out to be easier to use than expected, and the ability to use SQL made it easy to store and retrieve data.

The app was tested in two different ways. The first method was using the app like a user would and verifying whether the results being produced were accurate by checking the data in the database. The second method was creating a unit test suite to test the Habit and HabitTracker classes of the app. The unit tests proved useful and revealed that the app was not calculating streaks correctly. The fact that a database was being used to store and retrieve a user's data caused some issues when creating the unit tests. The timestamps for completion test data were the same, which caused incorrect streak calculations, and data that was already in the database was being used when getting streak statistics. Solutions to these issues were to manually insert completions with different timestamps and to use an in-memory database to store test data that would get deleted every time the unit test program was executed.

Overall, the habit tracker app is a success. It meets all the customer's requirements, and all the functionality works correctly. It is also fairly easy to use and install and does not require much technical knowledge. The design of the app also makes it easy to add new functionality to the app if required in the future. The features of the app that the team is most proud of are the ability to calculate the longest streak for a specified habit and the ability to calculate the longest streak for all habits. The streak information can motivate users to continue their efforts of forming a new healthy habit and also keeps them accountable, which is very beneficial for users and is central to the app's goal of helping users form new habits to improve their lives.

Github project link: https://github.com/stevening1/habit_tracker