Rachel Warner

Software Design Documentation

Personal Wellness Tracker

Contents

Introduction

The Personal Wellness Tracker will be an application that allows users to track and visualize their wellness in different aspects. The application will focus on visualizing financial and emotional well-being to aid the user in making better financial and emotional decisions.

This software's intended audience or users would be anyone interested in having an application that visualizes their financial and emotional wellness while being interested in data privacy as analysis, and storage are done locally.

Requirements Analysis:

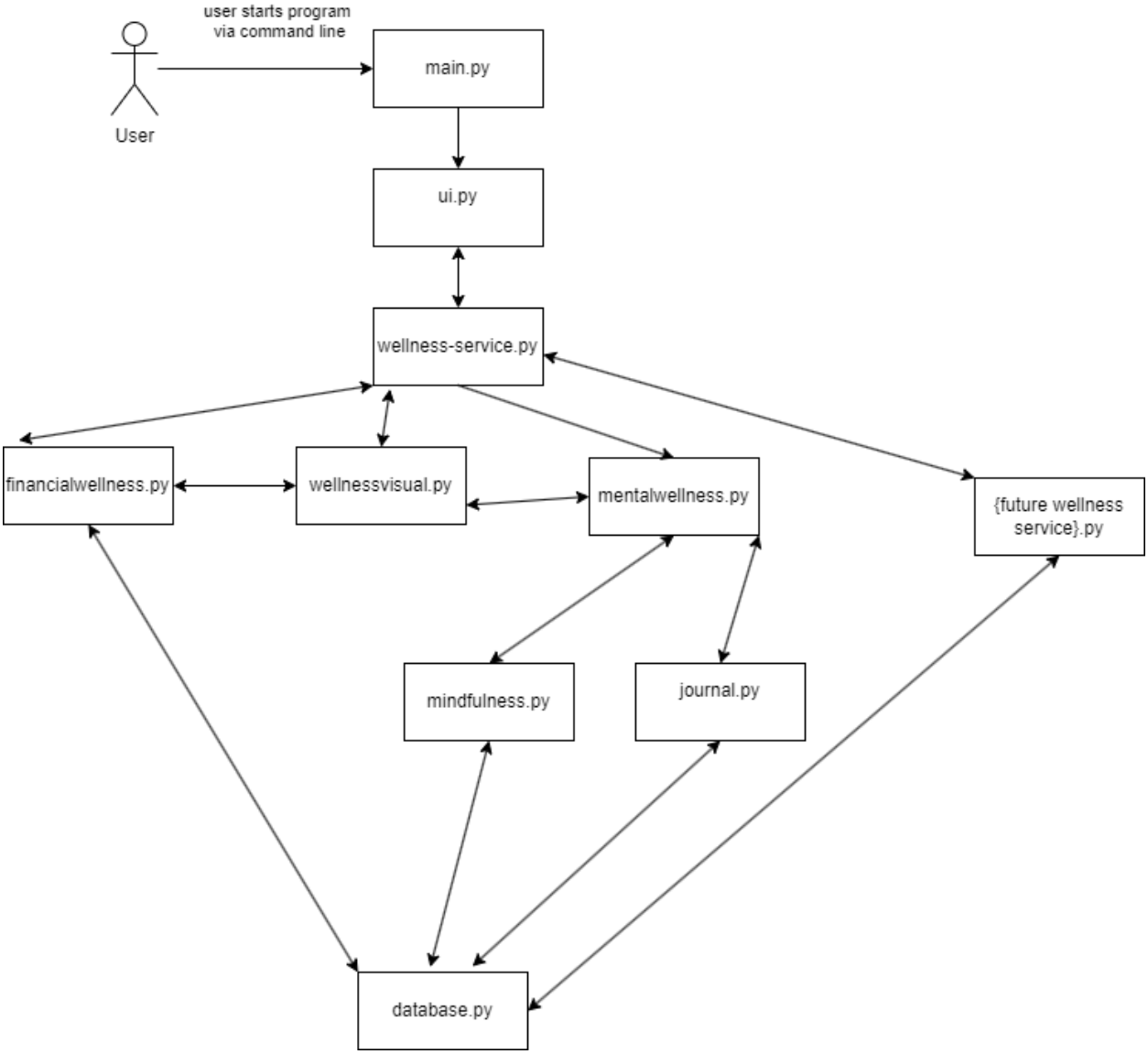
Necessary Libraries:

- Tkinter
- Matplotlib
- SqlAlchemy
- Sqlite

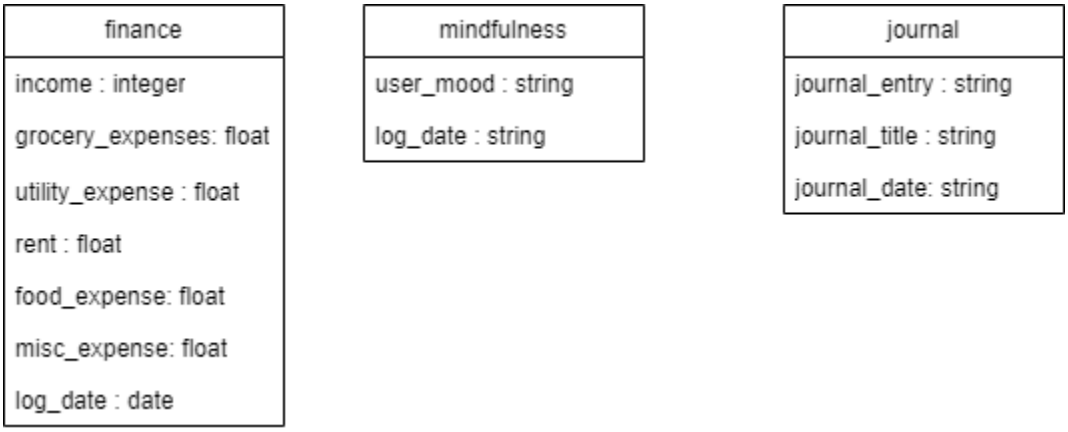
Requirements:

- The Personal Wellness Tracker shall allow the user to:
    - Track financial wellness through:
        - Displaying the user's financial analysis:
            - Spending vs income
            - Allow the user to select a date range to view
            - A pie chart providing a visual of the users' expenses
        - Allowing the user to:
            - Input salary and expenses
    - Track emotional wellness:
        - Displaying the user's wellness analysis:
            - Display the user's mood history via week or month.
            - Allow the user to select which month or week for emotional analysis
    - Journal:
        - Allow the user to:
            - Create, read, update, and delete journal entries

System Architecture



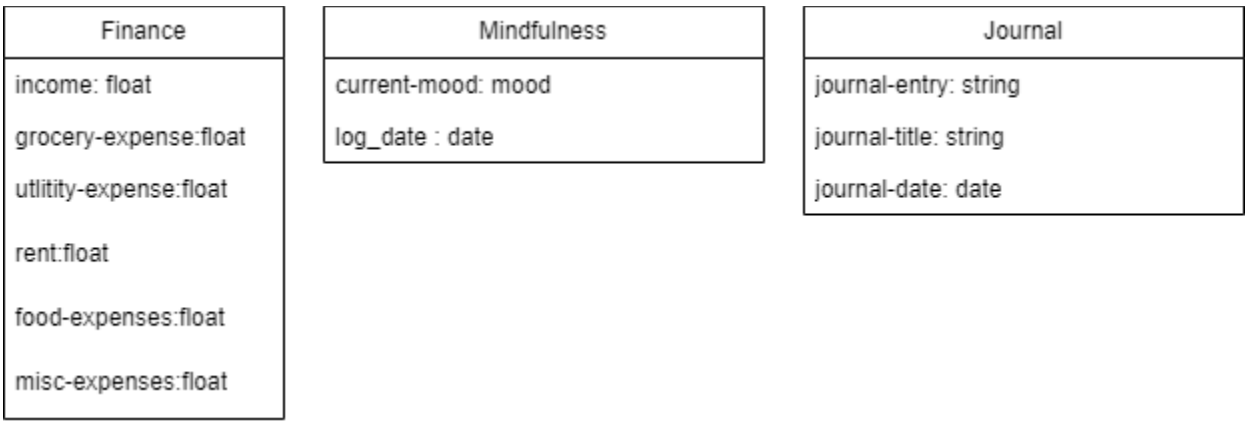
Database Design



The database is created and interacted with using SQLite and SQLAlchemy.

Data Design:

Three primary classes will be stored, retrieved, and manipulated.



Finance

The finance class holds all the finance information that will be stored, retrieved, and displayed to the user. These include common expenses like groceries and utilities and the user's salary. The user will also be allowed to adjust the amounts for updated reports and analysis.

Mindfulness

The mindfulness class holds data regarding the user's mood. The user will be able to change the mood for the day

Journal

The journal class holds the user's journal data it also holds a historical record of their journals. This

User Interface Design:

Tkinter:

Module/Component Design

Structure:

The following are the main components of the application

Main.py

- Serves as the entry point to the program

Ui.py

- UI.py may control two different UI displays.
    - The UI may consist of an interactive CLI or GUI via Tkinter.
- Contains the UI definitions for tkinter interactions with wellness.py
- Contains logic for displaying pie graphs for wellness information.

Wellness-service.py

- Serves as the middle-man between all of the wellness services
- The wellness-service module will be responsible for data validation before interacting with other wellness modules as well as for calling other modules when needed.

Finance.py

- The finance.py module serves as a module that fetches financial data from the database for displaying to the user
- A module fetching financial data from the database for displaying to the user and analysis.
- The finance.py module will allow users to:
    - Create, read, update, and delete financial data

Mindfulness.py

- The mindfulness.py module is a module that fetches historical mood data from the database to display to the user.
- The mindfulness.py module will allow users to:
    - Create, read, and update mood data

Journal.py

- The journal.py module is a module that will allow a user to retrieve historical journal data.
- The journal.py module will allow users to:
    - Create, read, update, and delete journal entries

Database.py

- The database.py module will handle the database logic for retrieving data, checking if the in-memory database exists, and creating the database.
- The database.py module will use SQLAlchemy as an Object Relational Mapper to assist with database interactions.

Testing Strategy:

Outline the testing methods and procedures to ensure the software meets the specified requirements

Include test cases, scenarios, and expected outcomes

Test case	Scenario	Expected Outcome	Pass/Fail
Welcome menu displays wellness options	Welcome menu displays: Financial and Mindfulness	Welcome menu displays after pin is entered	PASS
User is navigated to respected view from the welcome menu (a)	User selects financial	Financial Wellness menu/screen appears	PASS
User is navigated to respected view/menu from the welcome menu (b)	User selects Mental	Mindfulness wellness menu/screen appears	PASS
User at financial menu (a)	User at financial menu	Financial Menu Window displays	PASS
User at financial menu (b)	User enters information on financial menu	User can enter financial information (income, expenses)	PASS
User at financial menu (c)	User clicks the submit button	When user clicks submit input is validated	PASS
User at financial menu (d)	User clicks the submit button	The user is alerted that the changes were saved	PASS
User at financial menu (e)	User clicks the Back to Welcome button	User can navigate back to welcome menu	PASS
User at financial menu (f)	User at financial menu	By default, the pie chart shows all of the entries entered by the user	PASS
User at financial menu (h)	User enters new data and clicks the submit button	The pie chart displayed updates with new data inserted	PASS
User at financial menu (l)	User clicks the Log History button	The user is navigated to the Financial History Screen	PASS
User at Financial History Screen (a)	User at Financial History Screen	The user is able to see historical log of entries by date	PASS
User at Financial History Screen (b)	User clicks on a date on the Financial History Screen	The user is navigated to the update finance log screen	PASS
User at Update Finance Screen (c)	User at Update Finance Screen	The User is able to navigate to the Update Finance Screen	PASS

User at Update Finance Screen (d)	User clicks the submit button with complete data	The user is notified that the update was successful	PASS
User at Update Finance Screen(e)	User clicks the submit button with incomplete data	The user is notified that all fields require a value	PASS
User at Mindfulness menu (a)	User at mindfulness menu	Mindfulness menu window displays	PASS
User at mindfulness menu (b – 1)	User at mindfulness menu	Mindfulness menu displays a pie chart of the user's mood history	PASS
User at mindfulness menu (b – 2)	User at mindfulness menu	Mindfulness menu pie chart filterable by date	PASS
User at mindfulness menu (c)	User adding mood data	Mindfulness menu pie chart updates automatically with new data entry	PASS
User at mindfulness menu (d)	User should be able to navigate to welcome menu	After clicking a button, the user gets navigated back to the welcome menu	PASS
User at mindfulness menu (e)	User should be able to navigate to the Journal menu	After clicking a button the user gets navigated to the Journal menu	PASS
User at mindfulness menu (f)	User clicks on the Log History Button	The user is navigated to the Mindfulness Log History Screen	PASS
User at Mindfulness Log History Screen (a)	User at Mindfulness Log History Screen	The Log History Screen shows a history of mindfulness data by date	PASS
User at Mindfulness Log History Screen (b)	User clicks on date within Mindfulness Log History Screen	The user is navigated to the Update Mindfulness Screen	PASS
User at Update Mindfulness Menu	User at Update Mindfulness Menu	Update Mindfulness Menu appears	PASS
User at Update Mindfulness Menu	User updates the Mindfulness Value	The User is alerted it was successful/unsuccessful	PASS
User at Journal Display Menu	When at the journal menu, the user sees a chronological display of journal entries (newest first)	A chronological view of journal entries from the user is displayed	PASS
User at Journal Menu (a)	The user can add a journal entry with a Title	The user saves a journal entry and is alerted if successful	PASS

User at Journal Menu	The user adds a journal entry without a title	The date is set as the title of the entry	PASS
User at Journal Menu:	The user should be able to navigate back to the mindfulness menu	When the user clicks on a button, the user is navigated back to the mindfulness menu	PASS
User at Journal Menu	The user clicks on the Log History Button	The user is navigated to the Journal Log History Screen	PASS
User at Journal Log History Screen	The User is at the Journal Log History Screen	The user should see a history of the journals created by date	PASS
User at Journal Log History Screen	The User clicks on a log date	The user is navigated to the update journal screen	PASS
User at update journal screen	The user updates the journal	The journal update is saved	PASS

Implementation plan:

Milestone	Timeline	Complete
Create Local Database	July 1st – 7th	COMPLETE
Create Database Schema	July 1st – 7th	COMPLETE
Create finance.py Module	July 1st – 7th	COMPLETE
Create mindfulness.py module	July 1st – 7th	COMPLETE
Create journal.py module	July 1st – 7th	COMPLETE
Journal Data Persistence Complete	July 1st – 7th	COMPLETE
Finance Data Persistence Complete	July 8th – 14th	COMPLETE
Mindfulness Data Persistence Complete	July 8th – 14th	COMPLETE
Ui.py	July 8th – 14th	COMPLETE
Welcome page Created and displaying information	July 15th – 21st	COMPLETE
Wellness Service Page Created and displaying information	July 15th – 21st	COMPLETE
Financial Wellness Page Created and Displaying information	July 15th – 21st	COMPLETE

Mindfulness Page Created and displaying information	July 15th- 21st	COMPLETE
Journal Page Created and displaying information	July 15th – 21st	COMPLETE
Financial Log History Page Created and Displaying information	July 15th- 21st	COMEPLETE
Mindfulness Log History Page Created and Displaying information	July 15th – 21st	COMPLETE
Journal Log History Page Created and Displaying information	July 15th-21st	COMPLETE