Personal Finance Tracker Project Proposal

Team: Goofy Goldfishes
Team Members: Matt Bennett, Ally Herleth, Ruben Alvarez, Enoch Adegbola, and Jo Ann Kern
Due: August 31, 2025

Project Overview

The Goofy Goldfishes team proposes the development of a Personal Finance Tracker application which will henceforth be referred to as GillPay^{TM}. Designed with college students in mind. The tool will help users record income and expenses, categorize spending in ways that reflect a student's lifestyle, generate summaries, and visualize financial patterns in fun and engaging ways. By building a playful yet practical tracker, our project aims to improve financial awareness while keeping the process approachable and enjoyable.

Design and Application Features

The GillPay Personal Finance Tracker will be developed in a modular format (see Figure 1), allowing individual components to be designed, tested, and extended independently before full integration with the entire codebase/application.

The core functionality of the application will center on transaction recording, categorization, summaries, and data storage. Users will be able to record income and expense transactions by entering details such as amount, date, category, and optional roles. Categories will be tailored to the needs of college-aged students (e.g., food, rent, utilities, entertainment, transportation, textbooks, beverages, and other various and sundry campus fees. This should still permit our team to provide the flexibility to create and/or offer custom categories should we deem it necessary. Summaries will provide weekly and/or monthly views of total income, expenses, and net gains or losses. Data will be stored in a comma separated values table (CSV) format to ensure portability, simplicity, and ease of sharing.

As the project progresses, advanced features will add both reliability and engagement. Validation and exception handling will enforce proper data entry by checking for numeric values (non-string input), valid dates, and potential duplicate records, while striving to manage various file errors or empty datasets (null values) properly. Users will be presented with a drop-down-style category picker so as to avoid potential for improper input.

Visualization will be approached with playful but meaningful representations. Turtle graphics will be used to create engaging outputs such as a fish tank thermometer where it rises with positive net worth or sinks with negative. Additional concepts like “Fishing for Savings” with different-sized fish may also be explored to reinforce progress in a fun way that resonates with our proposed user base.

The interface will begin as a command-line menu to provide immediate feedback and usability and will later be expanded to include a simple tkinter-based graphical interface. This phased design will ensure that functionality is available early in development while leaving room for refinement and enhanced user experience in later phases.

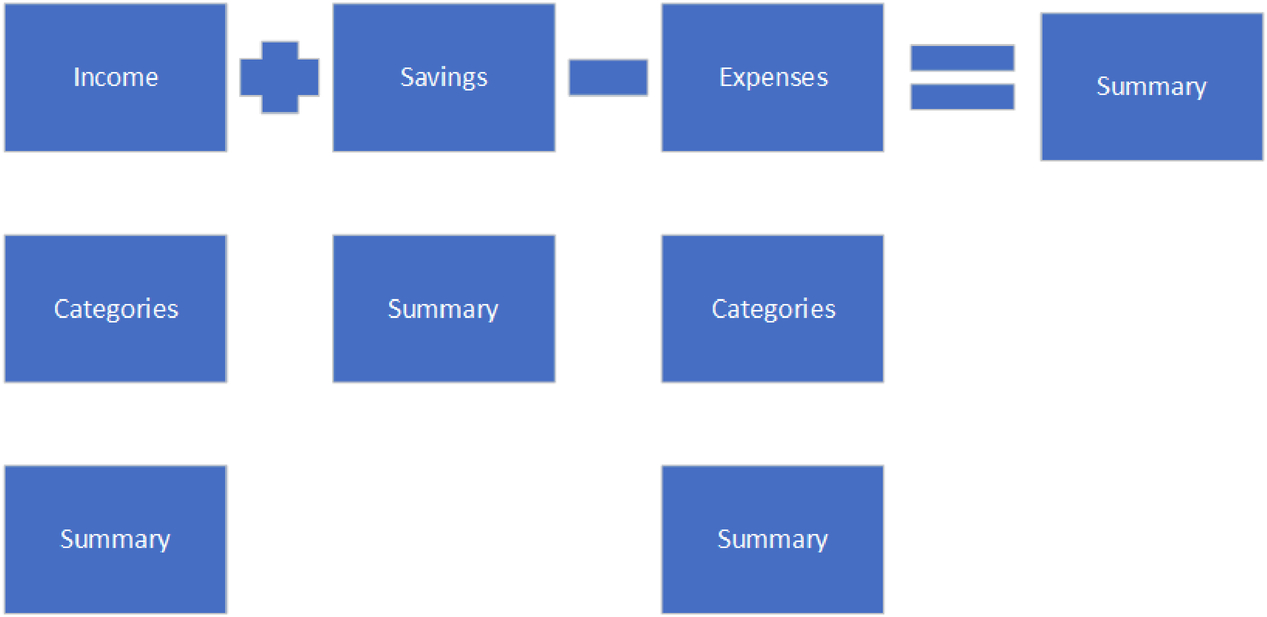

Figure 1 – Personal Finance Tracker Modules Overview

Timeline and Task Allocation

Planning and Initiation

The team met on 8.19.25 and 8.21.25 to develop a team charter (contract). The team also met on 8.26 and 8.28 to develop a project proposal, discuss requirements and design, and develop a project plan.

Team Alignment and Status Updates

Weekly meetings are scheduled throughout the project on Tuesday and Thursday evenings from 8-9 PM CST. The standing agreement is to finish deliverables one day (24 hours) before official deadlines. The team also agrees to review team progress weekly against the plan, review development status and test results, and to discuss modifications in an iterative approach to development.

Deliverables

Project deliverables will be developed in phases as noted below. Role assignments are listed with each phase.

Phase 1 – Project Proposal (Due Aug 31, 2025) - Est. 25 Hours

Team Leader: Matt

Documentation: Ally

Design and Plan Contributors: Ruben, Enoch, and Jo Ann

Key Milestones

- Aug 26: Finalize scope, features, roles and responsibilities (Team)
- Aug 28: Draft proposal, confirm libraries, and finalize approach (Team)
- Aug 30: Compile and edit (Documentors)
- Aug 31: Review and submit (Team Lead)

Phase 2 – Transaction Recording (Due Sept 14, 2025) - Est. 35 Hours

Expected Outcome: *A Python application that records transactions and saves them to a csv file. Documentation explaining the purpose and functional of the code (in code and external technical documentation).*

Team Leader: Enoch

Documentation: Jo Ann

Developers: Ruben and Matt

Testers: Ally

Key Milestones

- Sept 2: Team Meeting to coordinate tasks, integrate parts (Team Lead)
- Aug 29 - Sept 4: Develop pseudocode for peer review (Developers)
- Sept 4–9: Develop main module features including:
    - Input functionality and data validation rules (Developers)
    - Data storage and transaction recording (Developers)
    - Error handling and edge cases (Developers)
- Sept 9 & 11: Team Meeting to review progress, address issues/risks (Team Lead)
- Sept 11: Document inline code; create technical documentation; and update project plans (Documentors)
- Sept 11: Test with assertions in Python Code.(Testers)
- Sept 12: Adjust for any issues and retest (Developers)
- Sept 13: Final review and submission (Team Lead and Team)

Phase 3 – Reporting & Exception Handling (Due Sept 28, 2025) - Est. 35 Hours

Expected Outcome: *A Python application that creates multiple reports and handles exceptions including data validation errors, input errors, and edge errors. Documentation explaining the purpose and functional of the code (in code and external technical documentation).*

Team Leader: Ally

Documentation/Testing: Jo Ann and Enoch

Developers: Ruben and Matt

Key Milestones

- Sept 9 - Sept 11: Develop pseudocode for peer review (Developers)
- Sept 16 & 18: Team Meeting to review progress, address risks (Team Lead)
- Sept 16–23: Develop modules for reporting including summaries, validation, and exception handling (Developers)
- Sept 23 & 25: Team Meeting to review progress, address issues (Team Lead)
- Sept 24-25: Test assertions with Python Code. User test reports and CLI features (Testers)
- Sept 24-25: Document Code, Update Technical/User Documentation, Update Project Plan (Documentors)
- Sept 26: Adjust for any issues and retest (Developers)
- Sept 27: Final review and submission (Team Lead and Team)

Phase 4 – User Interface & Integration (Due Oct 12, 2025) - Est. 38 Hours

Expected Outcome: *A Python application that records transactions and saves them to a csv file. Documentation explaining the purpose and functional of the code (in code and external technical documentation).*

Team Leader: Jo Ann

Documentation: Matt

Developers: Ruben and Enoch

Testers: Ally

Key Milestones

- Sept 14 - Sept 18 - Develop pseudocode for peer review (Developers).

- Sept 30 & Oct 2: Team Meeting to review progress, address risks (Team Lead)
- Sept 30–Oct 7: Build Tkinter interface and Turtle visualizations (Developers)
- Oct 8: Integrate all modules (Developers)
- Oct 9 & 11: Team Meeting to review progress, draft project report (Team Lead)
- Oct 9: Test with assertions in Python and user testing (Testers)
- Oct 9: Document code, update technical documentation, create user documentation, update project plan (Documentors)
- Oct 10: Adjust for any issues and retest (Developers)
- Oct 11: Complete Project Report (Team Lead)
- Oct 11: Final review and submission (Team Lead)

Data Validation and Exception Handling

Data validation will be built into the code for specific validation rules. Testing will be conducted using Assertions in Python code to validate functions. Additional testing will ensure user experience and modular flow.

Validation Rules:

- Amounts must be numeric and non-negative.
- Dates must follow the YYYY-MM-DD format.
- Categories must exist or be created, with duplicates avoided.
- Type must be either income or expense

Exception Handling:

- File errors (missing, corrupted, incorrect data format, or locked CSV).
- Empty data sets (display a “no data available” message).
- Invalid inputs (prompt users to re-enter data that did not meet data validation rules for numeric, text, currency, dates, and categories).
- Edge Cases (Zero-income periods, empty categories, extremely large amounts over $9,999.99).
- Duplicate entries (warn and confirm before saving).

External Libraries

Proposed external modules we plan to reference:

- **CSV (standard library):** Core data storage in early phases, chosen for portability.
- **pandas:** Efficient handling and summarization of transaction data, plus preparation for visualizations.
- **Tkinter (standard library):** Simple GUI for user-friendly interaction in Phase 4.
- **Turtle (standard library):** Required for creative, visual engagement with financial data.

> “The pandastable library provides a table widget for Tkinter with plotting and data manipulation functionality. It uses the pandas DataFrame class to store table data. Pandas is an open source Python library providing high-performance data structures and data analysis tools. Tkinter is the standard GUI toolkit for python. It is intended for the following uses:

> - for python/tkinter GUI developers who want to include a table in their application that can store and process large amounts of data
> - for non-programmers who are not familiar with Python or the pandas API and want to use the included DataExplore application to manipulate/view their data
> - it may also be useful for data analysts and programmers who want to get an initial interactive look at their tabular data without coding" (Farrell, 2014)

Risk Management

The team will maintain a risks and issues log and a decisions log for tracking key aspects of the development effort. Risks will be identified, discussed in team meetings, and mitigated or accepted.

Key risks identified to date:

- **Inconsistent coding styles** → *Mitigation: use PEP8 and shared Git repo.*
- **Standards**→ *Mitigation: use ISTM standards and defined naming conventions*
- **Time commitments** → *Mitigation: stick to meeting schedule & contract deadlines.*
- **Deliverable Quality:** Ensuring high quality code→ *Mitigation: use pseudocode, conduct peer reviews, use assertions for testing, and user validation*

Team Responsibilities and Assumptions

- **Experience Level:** Each member has basic Python knowledge
- **Time Commitment:** Each member can commit 5–10 hrs/week.

Flexibility Clause

To preserve agility, the team reserves the right to adjust technical decisions as the project progresses. For example, CSV storage may be replaced by SQLite if performance, usability, or scope require a change. Any such adjustment will be approved by the team, documented in the Project Report, and reflected in updated design notes and timelines.

Generative AI Use Disclosure

The team used generative AI tools to brainstorm features, outline validation and exception handling strategies, and draft preliminary text. All AI-generated content was reviewed, edited, and refined to reflect the team's understanding and decisions. Final content is team-owned and tailored to our project direction.

Conclusion

The **GillPay: A Personal Finance Tracker for Students** is being developed with college students as the intended user group. The categories and functionality is aligned with the needs of college-students (typically age 18-24). Individuals outside of this group can use the functionality and customize categories. The proposal outlines the intended scope, features, design, and development plan that will allow our team to build a functional and creative finance tracking application for managing student finances. By combining modular design, practical functionality, and creative Turtle-based visualizations, our phased and collaborative approach will allow the Goofy Goldfishes to build a project that is both technically sound and engaging.

References

Farrell, D. (2014). *Introduction*. Introduction - pandastable documentation.

https://pandastable.readthedocs.io/en/latest/description.html