

Project Code Review Report

Project Title: Personal Budget Manager Software

Team: Binary_Builders

Date: 01/11/2023

Overview

The provided code appears to be an implementation of the "Personal Budget Manager Software" project described in the proposal. The code aims to offer a simple budget management solution. Here, I'll review the code in comparison to the proposal.

Code vs. Proposal

Programming Language and OOP Principles:

- **Proposal:** The proposal mentions that the project will be developed using the C++ programming language and will apply Object-Oriented Programming (OOP) principles.
- **Code:** The code aligns with this by implementing classes and objects, encapsulation, and object-oriented design. It utilizes a class named `BudgetManager` for managing financial data.

User Authentication:

- **Proposal:** The proposal outlines user registration and authentication as one of the features.
- **Code:** The code indeed implements user authentication. Users are prompted to enter their username and password, and it verifies this against stored data in separate files.

Budget Tracking:

- **Proposal:** The proposal mentions features for income tracking, expense tracking, and budget setting and monitoring.

- Code: The code successfully implements these features. Users can add income, expenses, and set budgets. The balance is accurately updated based on these transactions.

Transaction History:

- Proposal: The proposal suggests that the software will allow users to view their transaction history.
- Code: The code provides a function, `viewTransactions()`, which displays the transaction history, including the type (income, expense, or budget), category, and amount.

Data Security and Privacy:

- Proposal: Data security and privacy are highlighted in the proposal.
- Code: The code stores user data in separate files, providing a basic level of data privacy. However, a more secure method could be implemented.

User Interface:

- Proposal: The proposal emphasizes the importance of a user-friendly and intuitive interface.
- Code: The code provides a text-based user interface (console-based) with clear menus and instructions, although it lacks a graphical interface.

Reports and Analytics:

- Proposal: The proposal mentions generating comprehensive reports and analytics for users.
- Code: The code currently lacks advanced reporting and analytics. It provides basic financial information but doesn't offer in-depth analysis.

Timeline:

- Proposal: The proposal presents a project timeline with phases.
- Code: The code does not reflect the project phases mentioned in the proposal. The code appears to combine user authentication and the main menu without clear separation.

Additional Comments

The code's user interface is text-based and limited to a console. To align with the proposal's emphasis on a user-friendly interface, a graphical user interface (GUI) could be considered.

While the code implements basic user authentication, it lacks features such as user account creation and a password recovery mechanism.

Data security could be improved by employing more robust data encryption and storage practices, as mentioned in the proposal.

The code does not provide personalized financial insights and recommendations, as suggested in the proposal. Implementing this feature could enhance the software's utility.

Conclusion

The code partially aligns with the proposal, providing core features like income tracking, expense tracking, and basic user authentication. However, there are areas where it deviates from the original proposal, such as the lack of user account creation, a graphical interface, and advanced reporting. To fully realize the project's potential, further development is needed to bring the code in line with the proposed features and objectives.