

Gab's Banking System - Project Overview and Code Structure

Project Overview:

The Gab's Banking System aims to provide a basic simulation of banking functionalities through an interactive terminal interface. It prioritizes user authentication and secure handling of user account data stored in a CSV file. The system is designed with modularity in mind, separating concerns such as user authentication, balance management, account management, and UI display into distinct modules and directories. This structure enhances code organization, maintainability, and potential for future expansion.

- **main.py**: This is the main entry point of the Gab's Banking System. It handles user authentication (login/signup) and then directs authenticated users to the main menu. It also includes error handling for various exceptions like ValueError, FileNotFoundError, CsvError, and generic Exception. The file also checks the database file and content using the `check_database_file` and `check_database_content` functions from `file_checker.py`.
- **handle_user_choice_prompt.py**: This file presents the user with a navigation menu after successful login. It allows users to choose options like withdraw, deposit, display balance, display user information, manage account, or log out. It calls functions from other modules based on the user's choice.
- **file_checker_functions**: This directory contains modules for handling file-checking related operations.
 - **check_database_content.py**: Checks the integrity and validity of the content of the banking database CSV file.
 - **check_database_file.py**: retrieves the user banking database CSV file and creates a new banking database CSV file if it doesn't exist.
- **balance_management_functions**: This directory contains modules for handling balance-related operations.
 - **deposit_to_user_balance.py**: Handles depositing funds into a user's account.
 - **withdraw_from_user_balance.py**: Handles withdrawing funds from a user's account.
- **authentication_handlers**: This directory contains modules for handling user authentication.
 - **handle_authentication.py**: Authenticates the user whether they want to create a new account or log in to an existing account.
 - **auth_handling_functions**: This subdirectory contains the functions used by `handle_authentication.py`.
 - **create_new_account.py**: Creates a new user account in the system.
 - **verify_user_account.py**: Verifies the user's credentials against the stored data.

- **account_management_handlers:** This directory contains modules for handling account management tasks.
 - **handle_account_management.py:** Handles the display of account management options like change username, change password, and delete account.
 - **account_handling_functions:** This subdirectory contains the functions used by handle_account_management.py.
 - **change_username.py:** Allows a user to change their username.
 - **change_user_password.py:** Allows a user to change their password.
 - **delete_user_account.py:** Allows a user to delete their account.
- **UI_display_to_users_functions:** This directory contains modules for displaying information to the user.
 - **display_account_management_options.py:** Displays the options available for account management.
 - **display_balance.py:** Displays the user's account balance.
 - **display_user_information.py:** Displays the user's information (full name, ID number).
 - **display_user_navigation.py:** Displays the main navigation menu for the user.
- **Essentials:** This directory contains essential modules used throughout the banking system.
 - **Custom_Errors.py:** Defines custom exception classes (CsvError) for error handling.
 - **is_float.py:** Contains a function to check if a string can be converted to a float.
 - **Terminal_colors.py:** Defines color codes for terminal output.
 - **User_options.py:** Defines the options for user input.
- **gab's_banking_system_users.csv:** CSV file that stores user data, including user IDs, usernames, balances, and passcodes.

- - - CONTRIBUTORS - - -

Gabriel B. Semogan: Project Leader, Quality Assurance & Bug Reporting. Significantly improved system stability by resolving critical bugs

Czar Mari L. Rosas: Developed user authentication module and file integrity checker module. Addressed key issues in data synchronization and user authentication.

Jedidiah Genes B. Rañola: Designed the user interface and created all command line interface components. Improved system stability by resolving critical bugs

Derek Josh M. Reyes: Initial project setup, developed core program model and design. Improved system stability by resolving critical bugs