

Bank System Documentation

Introduction

The Bank System is a simple banking application designed to manage customers, accounts, and transactions. It provides functionality for customer management, account handling, and transaction tracking.

Classes

1. **Customer**: Represents a bank customer with attributes such as customer ID, name, age, login, and password.
2. **SavingsAccount**: Represents a savings account with attributes including account ID, customer ID, balance, and monthly withdrawal limit.
3. **CheckingAccount**: Represents a checking account with attributes including account ID, customer ID, and balance.
4. **pretty_print**: A utility function to print formatted messages.

BankSystem Class

Attributes:

- **customers**: A dictionary containing customer information with login as keys and corresponding Customer objects as values.
- **accounts**: A dictionary containing account information with account ID as keys and corresponding Account objects as values.
- **transactions**: A dictionary containing transaction information with account ID as keys and lists of transactions as values.
- **current_user**: The ID of the currently logged-in user.
- **admin_password**: The password for admin access.

Methods:

- **__init__**(): Initializes the BankSystem with empty dictionaries for customers, accounts, and transactions. Sets the current_user to None and defines the admin password.
- **login(login: str, password: str)**: Logs in a customer or admin based on provided credentials.
- **admin_login(password: str)**: Logs in the admin with the provided password.
- **logout()**: Logs out the currently logged-in user.
- **load_data()**: Loads data from files into the BankSystem's dictionaries.
- **save_data()**: Saves data from the BankSystem's dictionaries into files.
- **create_customer(name: str, age: int, login: str, password: str)**: Creates a new customer and adds them to the system.
- **create_account(customer_id: str, account_type: str)**: Creates a new account for an existing customer.
- **view_transactions(account_id: str)**: Displays the transaction history for a given account.
- **perform_operations(account_id: str)**: Allows the user to perform operations such as deposit, withdrawal, and transfer.

- **check_balance(account_id: str):** Checks and displays the balance of a specified account.
- **delete_account(account_id: str):** Deletes the specified account from the system.
- **start():** Starts the BankSystem application, allowing users to login, create customers, create accounts, view transactions, perform operations, check balance, and logout.
- **menu():** Displays the menu for logged-in users, allowing them to perform various operations.

Usage

To use the Bank System, simply run the script **__main.py__**. Follow the on-screen prompts to perform various banking operations such as login, creating customers, creating accounts, viewing transactions, performing operations, checking balance, and logout.

Password for admin = “admin”