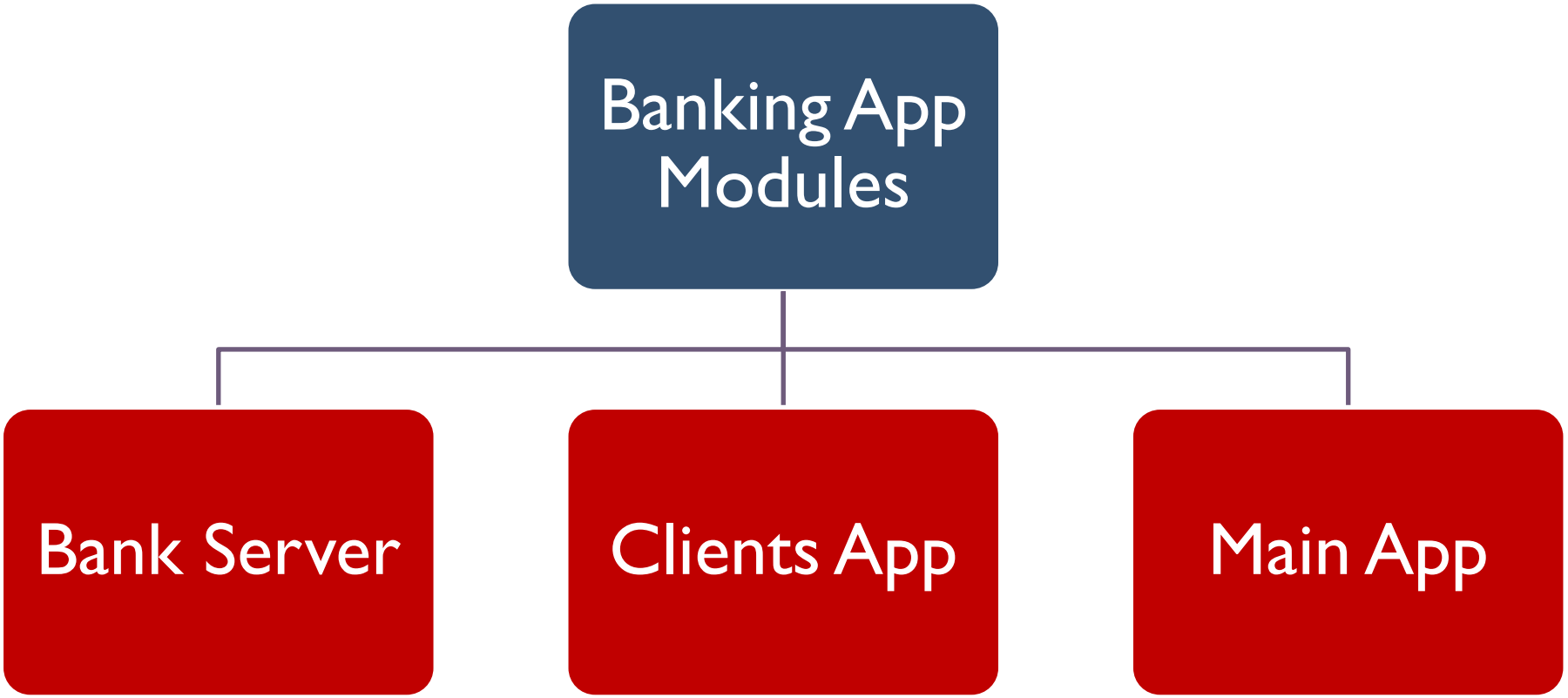
Ahmed Wael Essam

PYTHON FUNDAMENTALS PROJECT

Banking Management System App

Table of Contents:

Our Banking Management system app consists of 3 Modules which are:



We will discuss each module of these modules and go through each one of them in detail later in this document

First: Bank Server

Description

This Module contains the core logic of the application. It has the following classes:

- User: This class represents a bank user and has the following properties: **name**, **phone**, **balance**, **code**, **AccountNo**, **PaymentAdress**, and **Counter**.

- AmitBank: This class represents the bank and contains the following main methods:
  - create_user(): This method creates a new **User** object and adds it to the bank's list of users.
  - UsersCount(): This method returns the total number of users in the bank.
  - AccountLogin(): This method takes an account number and checks if it exists in the bank's list of users. If it does, the user is logged in.

Alongside other methods:

- Status (): This method prints the status of the user account (name, account number, balance)
- Transfer (): This method is used in transferring money between users in the bank as it takes the payment address of the recipient and the amount of money to be transferred

Methods

- Account Login (): inside this method, it verifies that there is an account associated with the entered by the user and then asks for his secret pin with 3 times entry trials in case of mistakes and then logs the user into his account

- Transfer (): inside this method, first it checks if the amount is available in the account and then searches for the given payment address of the receiving account, and if found. it asks for confirmation from the sender user by asking about his secret pin and validating it. Then it modifies the accounts of the sender and the receiver and their files in each user folder and increments the number of transactions to this user

- Status (): inside this method, it prints the user's name and account number, and current balance

Second: Clients App

Description

This Module is responsible for handling the user interface of the application, including input and output to the user. It contains several functions:

Numbering (): This function takes user input and returns an integer. It checks that the input is a valid integer and handles any exceptions.
UsersData (): This function takes a user object and saves it to a text file for persistent storage.
AccountMenu (): This function displays a menu for account holders to select different account operations.
Transactioncounter(): This function creates a text file containing the transaction history of the user

Methods

AccountMenu (): this function prints a list to the user when he logs into his bank account and takes input from a user according to his desired action from 4 options to deposit/withdraw/transfer/log out

- Deposit: takes the amount the user wants to deposit in his account and adds them to his current balance and updates his text file and adds a transaction in the transactions text file recording type of transaction and amount of money added to the balance and then logs the user into the previous menu to choose whether to perform another operation or logs out

- Withdraw: takes the amount the user wants to withdraw from his account and takes from his current balance and updates his text file and adds a transaction in the transactions text file recording type of transaction and amount of money taken from the balance and then logs the user into the previous menu to choose whether to perform another operation or logs out

- Transfer: asks for the receiver payment address and concatenates it to '@AmitBank' and then asks for the transferred amount and calls the Transfer function discussed in the *BankServer Module*

Finally: MainApp

Description

This Module is the entry point for the application. It imports the *ClientsApp* and *BankServer* modules and contains the following functions:

- **MainMenu ():** This function displays a menu for the user and returns an integer representing the user's choice.
- A while loop that runs the application and handles user input based on the user's choice from the menu.

Methods

While loop: it has an always true condition to keep the app running repeatedly, and inside this loop, it takes the user input to create a new user in the bank or log in to his current account or view the number of customers in the bank or to exit the app

- **1 (Create a new user):** it takes all the needed data to create a new user to match the User class in *BankServer* with some restrictions on some inputs such as the pin and minimum amount of money to open a bank account and provides each user with an account number and payment address for transferring and receiving money

- **2 (Login):** it asks the user for his account number and checks in the bank user's database for the given account number if found it continues to ask about his pin and sends his data to the AccountMenu function that was discussed earlier in *ClientsApp*

- **3(Display Users count):** it displays the number of all users in the bank to the terminal