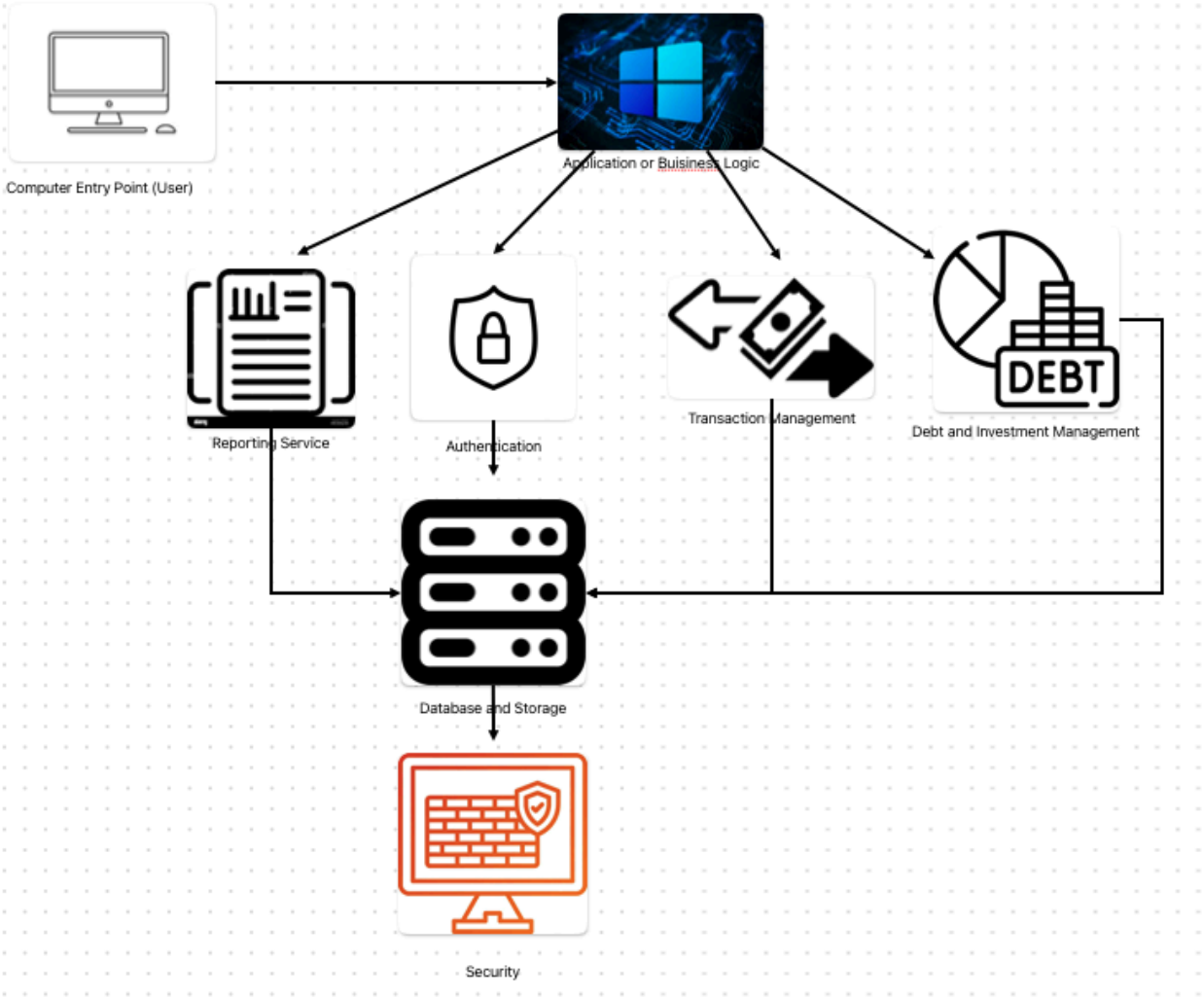
Software title: Personal Finance Assistant

Team members: Duc Nguyen, Adrian Balingit

Brief overview of the system: The Personal Finance Assistant is a comprehensive software designed to help individuals, especially the young, achieve their financial goals by helping them manage their finances effectively. It provides users with tools for budgeting including tracking income and spending to aid in financial management. Moreover, the software also offers tools for managing debt.

I. Architectural diagram of all major components



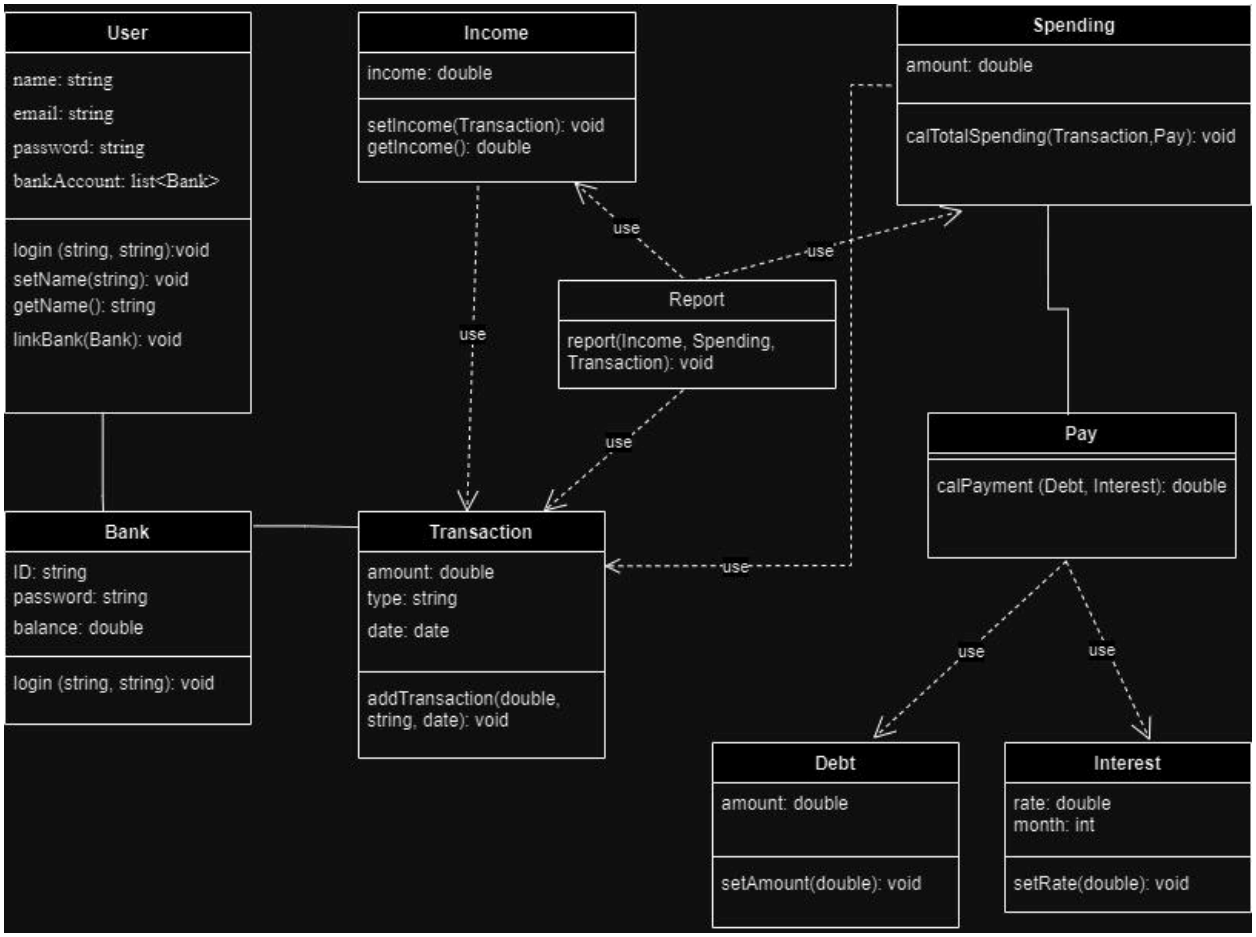
Description of SWA Diagram

1. User Interface (UI)
    - Will include all the graphical interface and user interaction mechanisms. This will include mobile applications, web pages, and the interactive tools for budgeting, financial planning, and spendings tracking.
    - Connects and communicates directly with the Application/Business Logic Layer to take in user input and display data.
2. Application/Business Logic
    - Manages user login, user session, and security

- Connects with Data Access Layer for data operations such as reporting services, authentication, transaction management, and Debt and investment management

3. Reporting Service
    - Generates detailed financial reports for the user that also includes insights.
    - Connects to the Database and Storage
4. Authentication
    - Manages the user login and security checks
    - Connects to the database and storage
5. Transaction management
    - Tracks, categorizes, and stores user transactions
    - Connects to the database and storage
6. Debt and Investment Management
    - Tracks debt and gives investment advice
7. Database and Storage
    - Consists of databases and storage systems responsible for persisting user data, transaction record, and financial plans.
    - Connects to the last layer of security
8. Security
    - All security measures for sensitive financial data that will include encryption, authentication protocols, and authorization checks.

II. UML Class Diagram: User Class, Bank Class, Transaction Class, Income Class, Debt Class, Interest Class, Pay Class, Spending Class, Report Class.



Description of classes:

User Class: Represents users of the software

Bank Class: Represents the bank linked by the user

Transaction Class: Represents all the transactions that are happening in the bank

Income Class: Represents the income based on transactions

Debt Class: Represents the amount of the user's debt

Interest Class: Represents the debt's interest rate and the number of months needed to pay

Pay Class: Represents the amount the user needs to pay based on the amount of debt and interest

Spending Class: Represents the total spending including transactions and debt pay

Report Class: Showing report of user spending and income.

Description of attributes:

1. User Class:

- name: a name chosen by the user
- email: email address of the user for communication and authentication
- password: password chosen by the user for authentication
- bankAccount: a list of linked bank accounts of the user

2. Bank Class:

- ID: the user's bank account ID for authentication to link the bank account
- password: user's bank password for authentication to link the bank account
- balance: the balance user has in their bank account

3. Transaction Class:

- amount: the amount of money in each transaction
- type: type of transaction
- date: the date the transaction occurred

4. Income Class:

- income: the amount of user income determined by the positive value transaction

5.. Spending Class:

- amount: the amount of user spending determined by the negative value transaction plus debt pay

6. Debt Class:

- amount: the amount of debt of the user

7. Interest Class:

- rate: the interest rate of the debt
- month: the amount of month needed to pay off the debt

8. Pay Class: no attributes

9. Report Class: no attributes

Description of operations

1. User Class:

a. login(string email, string password): void

- Description: Authenticates the user to establish a secure connection to access their account.
- Parameter: email and password (type: string for both) provided by the user to connect to their account.
- Return: void.

b. setName(string name): void

- Description: Set the name provided by the user for their account.
- Parameter: name (type: string) provided by the user.
- Return: void.

c. getName(): string

- Description: Get the user name to display it.
- Parameter: none.
- Return: the user name (type string)

d. linkBank(Bank account): void

- Description: Links the user's bank account to their personal finance assistant account to track their transactions within their bank account.
- Parameter: account (type: bank) the user's bank account linked to their personal finance assistant account.
- Return: void

2. Bank Class:

login(string email, string password): void

- Description: Authenticates the user to establish a secure connection to link their bank account to access their balance and transaction.
- Parameter: email and password (type: string for both) of the bank provided by the user to link to their account.
- Return: void

3. Transaction Class:

addTransaction(double amount, string type, date date): void

- Description: get transactions from the linked bank account and add them to the user's personal finance assistant account to track their income and spending.
- Parameter: basic information of the transaction including amount of money (double), type of transaction (string), and the date of transaction (date).
- Return: void.

4. Income Class:

a. setIncome(Transaction t): void

- Description: Get the positive value transactions of the linked bank account and set it as income.

- Parameter: t (type Transaction) the transactions from the linked bank account and only accept the positive amount.
- Return: void.

b. getIncome(): double

- Description: Get the income for further use.
- Parameter: none.
- Return: user income (type: double).

5. Spending Class:

calTotalSpending(Transaction t, Pay p): void

- Description: calculate the total spending including the debt payment and assign it to the amount
- Parameter: the transactions (type Transaction) from the linked bank account, only accept the negative amount and the debt payment (type Pay)
- Return: void

6. Debt Class:

setAmount(double amount): void

- Description: set the debt amount
- Parameter: amount (type double) amount of debt provided by the user
- Return: void

7. Interest Class:

setRate(double rate): void

- Description: set the debt interest rate
- Parameter: rate (type double) interest rate provided by the user

- Return: void

8. Pay Class:

calPayment(Debt d, Interest i): double

- Description: calculate the debt payment per month based on the amount, rate, and how many months to pay.
- Parameter: d (type Debt) amount of debt, i (type Interest) interest rate with the month
- Return: the payment per month (type double)

9. Report Class: no attributes

report(Income i, Spending s, Transaction t): void

- Description: Report all the information about user finance including how much they earn and spend each month, and also their detailed information in the transactions.
- Parameter: i (type Income) amount of income, s (type Spending) , and t (type Transaction)
- Return: void

Partitioning of tasks:

Feb 28: Software title and Brief overview by Duc Nguyen

March 7: UML diagram and description by Duc Nguyen

Team member responsibilities