Assignment 5

Making a Banking System using Python

Programming Languages SWE3006_42

- For this Assignment, you will be creating a Banking System using Python.

- You are required to be able to use json to store and retrieve informations during the interactions.
  - You can use a .json file with python to store and retrieve the usernames, passwords and the interactions. For more detail, you can visit https://codeinstitute.net/global/blog/working-with-json-in-python/.

- During the assignment, you will make two users to show the interaction between the two.

- All transactions and user details should be stored and retrieved from json file(s).

Initial screen

- When you first run the program, you will have three options {Register, Login,Exit}

- When you choose Exit, it will terminate the program.

Register

- You will input your username, login password, PIN password.
- When registration is successful, the user will be assigned a random 5 digit number as the bank account number.
- When registration is unsuccessful, output error message(s) and retry the registration process from the start.
- Error Messages:
  - Username: If there exists a user with the same username, print an error message.
  - Login password: It needs to be over 7 letters, contain at least one capital letter and contains at least one special character.(we simply define a special character as one of !@#$%^&*())
    - If the input password does not meet the requirements above, print an error message of what requirements were violated. (If all three requirements are violated, print all three)
  - PIN password: It is a 4 digit number password used for withdrawing, depositing and transfering. If the input PIN password is not a 4 digit number, print an error message.

Login

- You will input the username and the login password to login
- If the login details are invalid, print an error message.
- When login is successful, it will move to the main screen.

Main screen

- It will always display the username, current balance (new users have 100$) and the bank account number.

- If one of the five options {History, Withdraw, Deposit, Transfer, Logout} are chosen, it will switch to the corresponding screen. Note that Logout option makes you go back to the initial screen.

1.History

- In this screen, it will display all the transaction history of the user.
- All transactions will be listed in reverse time order (The most recent transaction will be listed first)
- All transactions will have the time information displayed. (When it happened)

```
from datetime import datetime
datetime.now().strftime('%Y-%m-%d %H:%M:%S')
```

- Transaction types and the displayed information:
  - Withdraw: Time, Withdrawed amount
  - Deposit: Time, Deposited amount
  - Transfer: Time, From, To , Transferred Amount
    - From: The username of the person that sent the money
    - To: The user name of the person that received the money

2.Withdraw

- It will always display the username, current balance and the bank account number.
- Withdraw the input amount of money from the balance.
- Input the right PIN password to successfully withdraw.
- Possible exceptions handling.
- Returns to Main Screen when successful.

3.Deposit

- It will always display the username, current balance and the bank account number.
- Deposit the input amount of money to the balance.
- Input the right PIN password to successfully deposit.
- Possible exceptions handling.
- Returns to Main Screen when successful.

4.Transfer

- It will always display the username, current balance and the bank account number.
- Input the bank account number and the wanted amount to transfer.
- Input the right PIN password to successfully transfer.
- Possible exceptions handling.
- Returns to Main Screen when successful.

Evaluation

- You are only allowed to use **Python** for this assignment.

- You should submit a zip file contains the python code(s),storage file(s) and a report in i-campus and name the zip file with {student_id}_Assignment5.zip (If any one of the three submissions are missing, you will be given 0 points for the assignment):

    - **Code**: {student_id)_banking.py file containing the python code for the game. We will be running this file to assess.
        - You are free to use more python files to code. If so, submit these files as well and include it in the report.

    - **Storage File**: The actual result of the json file(s) used for writing the report.

    - **Report**: You should submit 1 pdf report {student_id}_banking_report.pdf with:
        - 1.The explanation of your code.
        - 2. The architecture of how you stored and retrieved the information from the json file(s).
        - 3.The screen shots of the results from your coded program.

Report screen shots

- When simulating your program and taking screenshots, you should at least do the following:
  - Register two different users before moving onto the transaction pages
  - Do all three transaction types for each users before showing history
  - Show the exceptions handled for all pages (if there are).

- **Design**(10pt): If you make the interface neat enough and easy to understand, you will get full points.

- **Code**(80pt): All pages should work as intended and have the exceptions (if there are) handled. Your points will be deducted for each mishandled exceptions.
  - Initial Screen (5pt)
  - Register (15pt)
  - Login (5pt)
  - Main screen (5pt)
  - History (15pt)
  - Withdraw (10pt)
  - Deposit (10pt)
  - Transfer (15pt)

- **Report**(10pt)

Any type of plagiarism, code sharing and usage of chat models like Chat-GPT will result in your final grade being F.

QnAs

Please use the Discussion session in I-campus for any questions about the assignment.

Additional statement: We are sorry for not replying to the questions on the previous assignments. We will make sure to consider all the misconceptions and the questions on the assignment 3 and 4 when grading. We are sorry once again…