ATM Class

This class is built to handle the creation of accounts, the storage of passwords, balance, transactions history and more tasks like the same

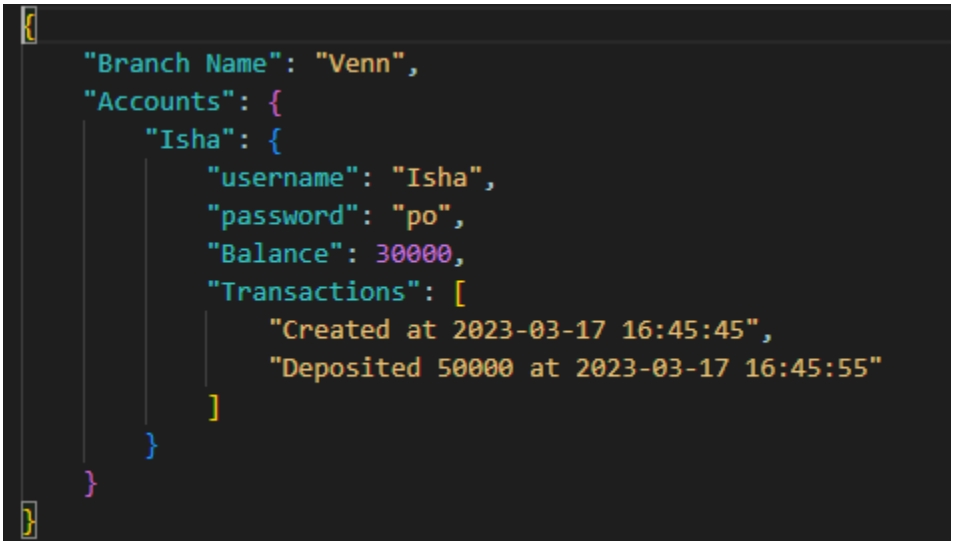
__init__(self,branch_name,json_path)

Arguments

- branch_name: Name of the branch / ATM
- json_path: The path to the json file that stores the all the records of all the users
  A json file is basically a nested dictionary that stores all the data.
  This is the structure of the json file.

Variables

- self.logged_in : indicates if any user is currently logged in
- self.logged_in_user : saves the username of the user currently logged in.



Function

- Will read and load the json file into the memory
- And will also initialize certain variables

create_acc(self,user_acc,password)

Arguments

- Creates a new account in the json

  The structure of each person's account is the following
  - Username - Stores the username as a string.
  - Password - Password, duh as a string
  - Balance - Current money in the account as INT
  - Transaction - Stores the details of all the transactions in a LIST

```
person = {"username":user_acc,
          "password":password,
          "Balance":0,
          "Transactions":[timestamp]}
```

Function

- Adds the person's account in the list of accounts stored in the main json file.
- Also logs the user in.

login(self,user_acc,password):

Arguments:

- user_acc : user name
- password : password of the user, used to log the user in

Function

- Logs the user in after authentication, sets the two log in variables
- Also returns different status codes
  - Logged in - For when the login was successful
  - Incorrect Password - For when the password was incorrect
  - Not a Member - For when the account doesn't exist.

current_balance(self)

Function:

- Returns the current balance of the logged in user as an INT 💰

deposit(self,amount)

Argument:

- amount: The amount of money you want to deposit (there's no limit, cuz I haven't studied economics, and I don't know the consequences of inflation) 💸

Function

- Increases the balance of the logged in user with the specified amount. 🏦

withdraw(self,amount)

Argument

- amount : The amount of money you want to withdraw 💸

Function

- Will withdraw the required money and subtract it from the balance if the amount is less than or equal to the balance

logout(self)

Function

- Logs the current user out

delete_acc(self,password)

Function

- Deletes the user after confirming the password.

commit_changes(self)

Function

- All the changes made the self.bank file will get erased when the program stops executing
- To ensure that the new changes are saved, they are written to the same json file from which all the data was initially read from.