**Python Mini Project Report**

**Name: Aashray Lal Nakarmi**

**USN: 22BTRSN060**

**Sem / Sec: 2/A**

**Github Link:** <https://github.com/aashrayLalnakarmi/Bank-Management-System>

**Blogger:** [**https://aashraylalnakarmipython.blogspot.com/2023/07/bank-management-system-python-mini.html**](https://aashraylalnakarmipython.blogspot.com/2023/07/bank-management-system-python-mini.html)

**BANK MANAGEMENT SYSTEM**

**Abstract**

This is a Python program designed to provide a simplified solution for managing customer accounts in a bank. It allows bank employees to perform various operations, such as creating new accounts, depositing and withdrawing funds, viewing account details, modifying account information, and closing accounts. The program utilizes class-based structures and data persistence through pickling to store and retrieve account data efficiently which has been taught to us throughout the 5 modules. The program is menu-driven, offering an intuitive interface for bank employees to navigate through different functionalities easily. Error handling and input validation ensure a smooth user experience and prevent any unintended errors during account operations.

The project aims to provide a user-friendly and effective solution for bank employees to manage customer accounts efficiently, making it a valuable tool for bank management and administration. In summary, the Bank Management System program serves as an accessible and functional tool for managing customer accounts, showcasing the effective utilization of Python programming techniques for data management and user interaction.

1. **Problem Statement**

The system should be able to handle the creation of new accounts, deposit and withdraw amounts, display account details, list all account holders, close an account, and modify account information. In order to create such a system I have identified what problems we must address, this allows for easier implementation of the program later on.

**Features:**

Create Account: The system should allow bank employees to create a new customer account. They should be able to enter the account number, account holder's name, account type (either 'C' for current or 'S' for savings), and the initial deposit amount.

Deposit Amount: Bank employees should be able to deposit a specified amount into a customer's account. They will need to provide the account number and the amount to be deposited.

Withdraw Amount: Bank employees should be able to withdraw a specified amount from a customer's account. They will need to provide the account number and the amount to be withdrawn. However, the system should ensure that the account has sufficient balance before processing the withdrawal.

Balance Enquiry: Bank employees should be able to check the balance of a specific customer account. They will need to provide the account number, and the system should display the account balance.

Display All Account Holders: The system should provide an option to display details of all the account holders. This will allow bank employees to see a list of all customers along with their account numbers, names, account types, and current balances.

Close an Account: Bank employees should have the ability to close an account. They will need to provide the account number, and the system should remove that account from the records.

Modify an Account: Bank employees should be able to modify the details of a customer's account. They will need to provide the account number and then be able to update the account holder's name, account type, and balance.

1. **Methodology**

This Bank Management System program is designed to simulate a basic banking system that allows bank employees to perform various operations related to customer accounts. For the implementation of this program I have used the problems stated in the problem statement and given the implementation details for the same.

**Class Definition:**

The program starts by defining a Python class named Account, which represents a customer account. This class includes various methods to perform operations like account creation, deposit, withdrawal, balance enquiry, etc.

The Account class has attributes such as accNo (account number), name (account holder's name), deposit (account balance), and type (account type - either 'C' for current or 'S' for savings).

**User Interface and Menu:**

The main functionality of the program is driven by a menu-driven interface. The user is presented with a list of options to choose from using the input() function.

**Account Operations:**

Create Account: The method createAccount() is used to create a new customer account. It prompts the user to enter the account details (account number, account holder's name, account type, and initial deposit) and then creates a new account instance.

Deposit and Withdraw Amount: The methods depositAmount() and withdrawAmount() allow the bank employee to deposit and withdraw funds from an existing account. The appropriate amount is added to or subtracted from the account balance.

Balance Enquiry: The method displaySp(num) allows the bank employee to check the balance of a specific customer account by providing the account number.

Display All Account Holders: The method displayAll() displays details of all the account holders, including account number, name, account type, and balance.

Close an Account: The method deleteAccount(num) allows the bank employee to close a customer account by providing the account number. The corresponding account is removed from the records.

Modify an Account: The method modifyAccount(num) allows the bank employee to modify the account details (name, account type, and balance) of a specific customer account by providing the account number.

**Data Storage and Persistence:**

The program uses pickling to store customer account data in a file named "accounts.data". Pickling is basically a method for serializing Python objects, making it convenient for data persistence.

The methods writeAccountsFile(account) and writeAccount() handle the writing and updating of account data into the "accounts.data" file.

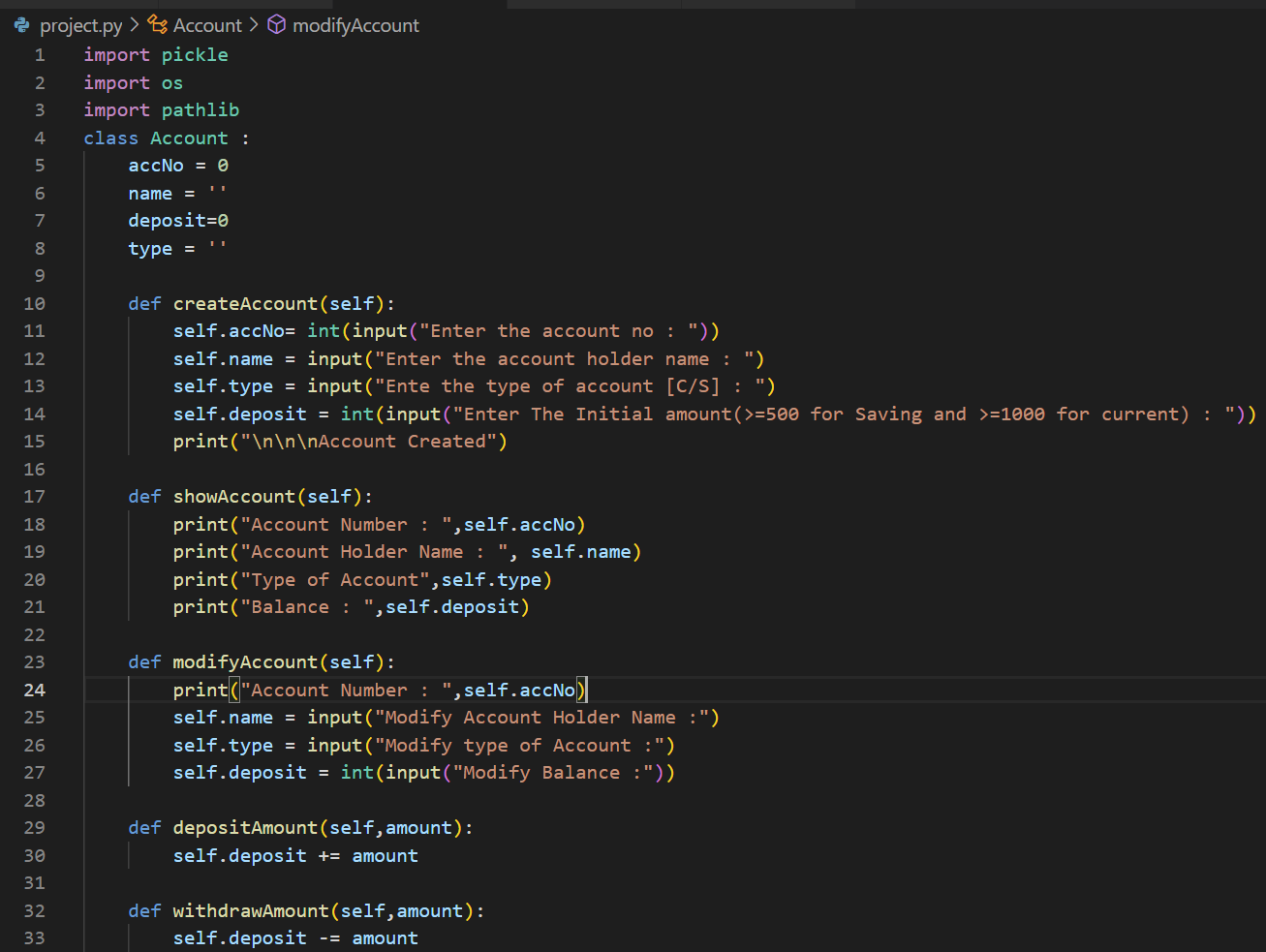
**Error Handling and Input Validation:**

The program performs input validation to ensure that the user provides valid inputs for account-related operations. The system checks for the existence of the "accounts.data" file before reading or writing data to it.

**User-Friendly Interface:**

The program provides a user-friendly interface with clear prompts and messages to guide the bank employee through various operations.

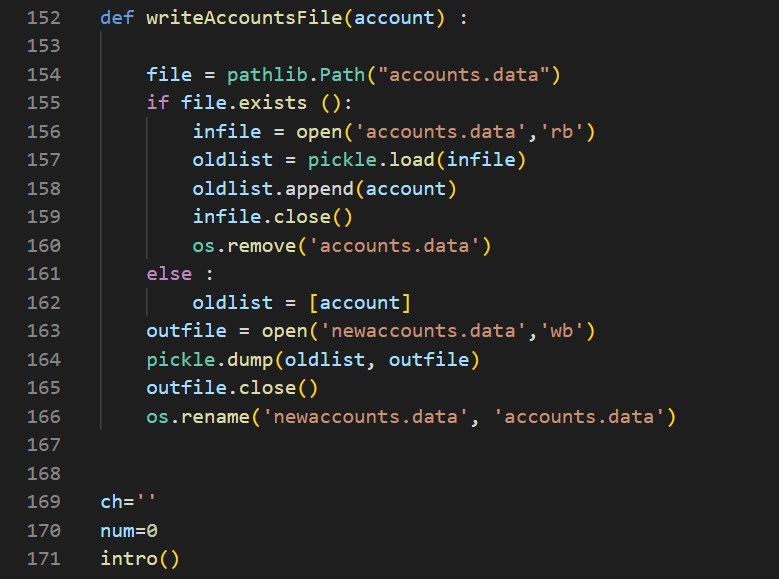
1. **Coding and Results (Snapshot)**

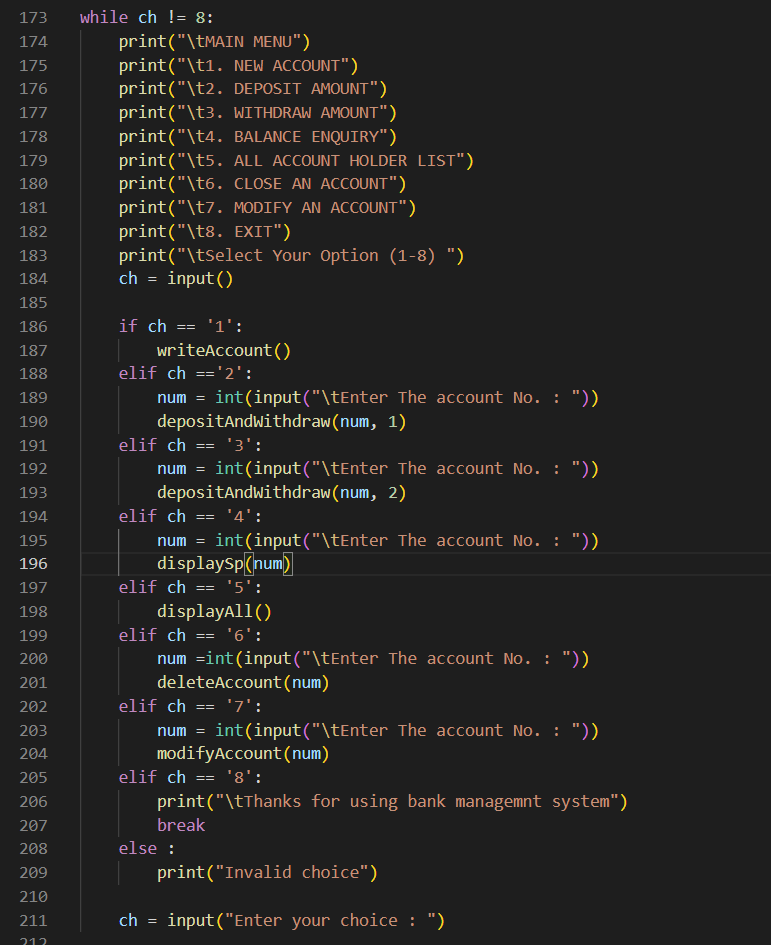
****

****

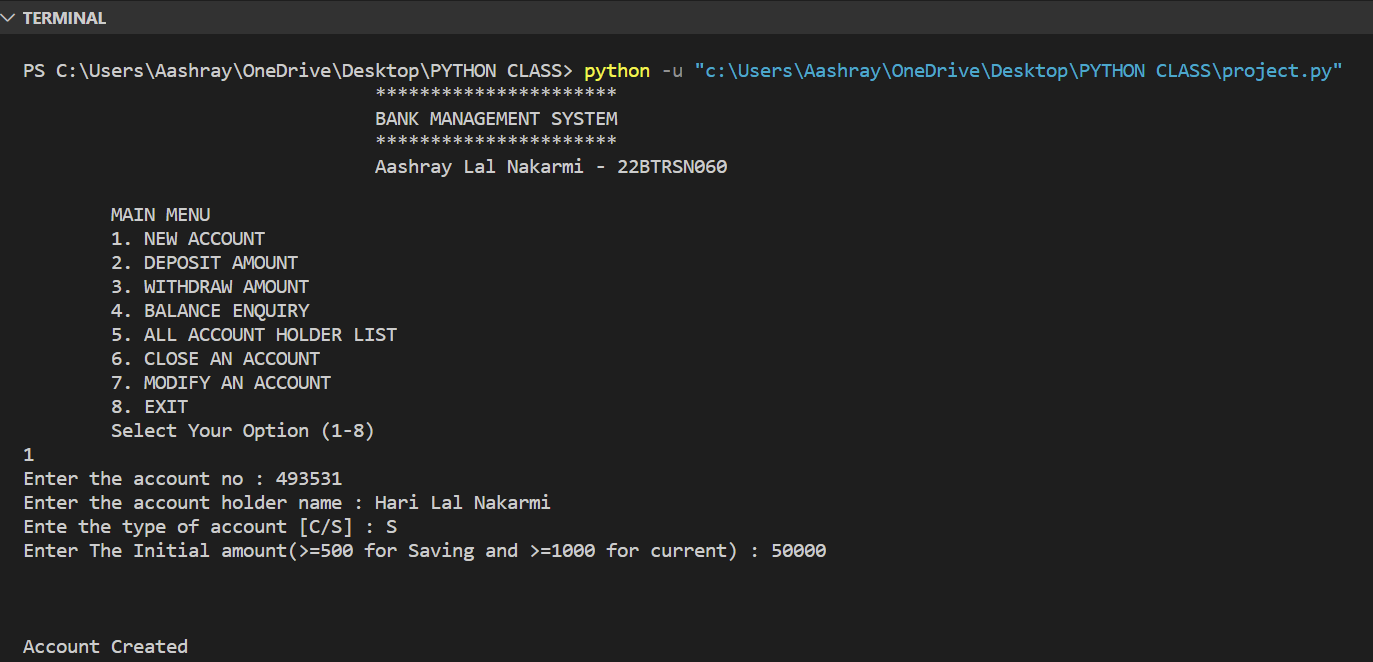
****

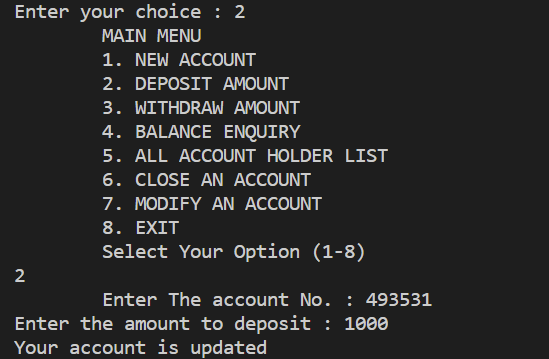
****

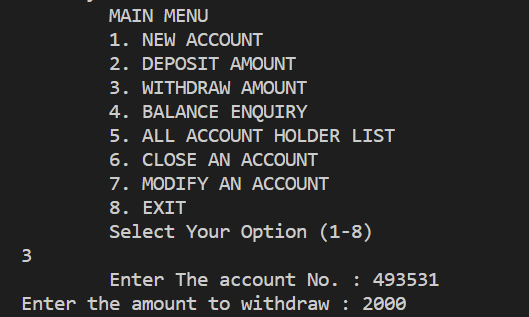
****

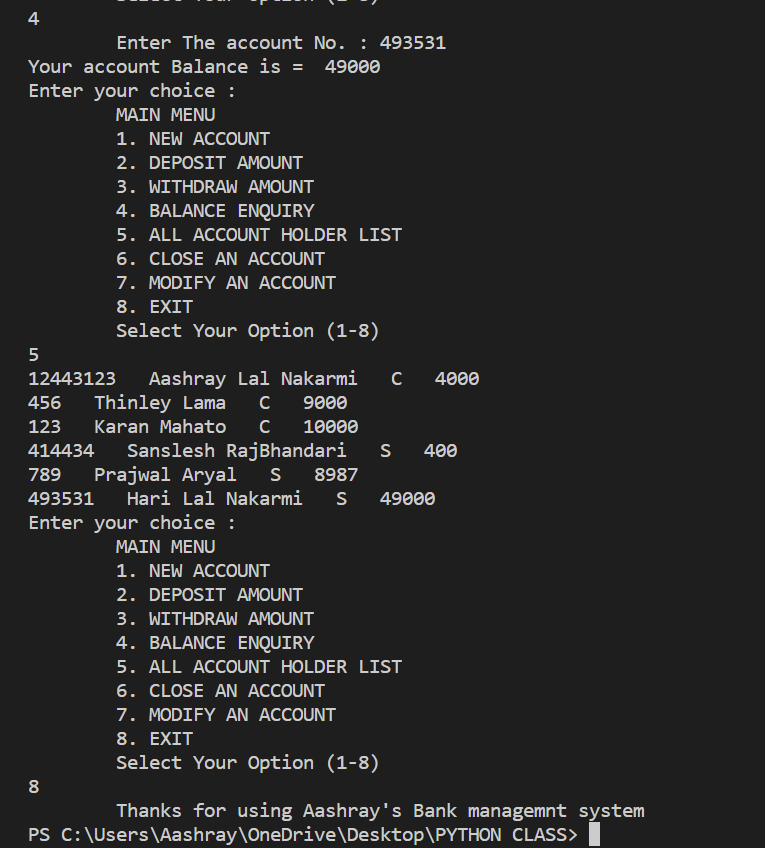
****

**OUTPUT:**

****

****

****

****

1. **Conclusion**

Overall, the Bank Management System program successfully fulfills the problem statement requirements and provides a practical solution for managing customer accounts in a bank. By following the guidelines and instructions provided in the report, bank employees can efficiently utilize this system to perform their daily tasks and operations.

To further enhance the system, potential improvements could be considered, such as adding more advanced features like interest calculations, transaction history tracking, and security measures like password protection for account access.

In conclusion, the Bank Management System offers a solid foundation for managing customer accounts and demonstrates the effectiveness of using Python programming with class-based structures and pickling for data persistence. With continuous improvement and development, this system can evolve into a comprehensive and sophisticated banking management solution.

1. **References**

* "A Bank Management System in Python" by Akash Kumar and Gaurav Mishra (2018). This paper presents a bank management system implemented in Python. The system allows users to create new accounts, deposit and withdraw money, and check their balance.
* "Design and Implementation of a Bank Management System" by Swati Gupta and Anju Singh (2019). This paper presents the design and implementation of a bank management system. The system is implemented using the Java programming language.
* "A Secure Bank Management System" by Ankita Jain and Ankita Sharma (2020). This paper presents a secure bank management system. The system uses various security measures to protect user data.