**Final Report: Console-Based ATM System in C++**

**1. Introduction**

This project involves the development of a console-based ATM system written in C++ as a simulation of basic banking operations. The system is designed to manage user accounts, provide customer and admin functionality, and store data persistently using file handling techniques. The goal is to provide a robust, user-friendly ATM simulation suitable for integration with a GUI in the future.

**2. Methodology**

**2.1 Tools and Languages**

* Programming Language: C++
* Compiler: gcc & g++
* Platform: Windows

**2.2 Project Structure**

* Object-Oriented Programming (OOP) was used to model the ATM system.
* Classes include:
  + Account: Base class for account management
  + Transaction: Handles deposits, withdrawals, and transfers
  + ATM: Core class managing user interaction and overall system flow
* Users are stored in text files (accounts.txt) instead of arrays or vectors.

**2.3 File-Based Storage Approach**

* Each user account is serialized and stored in a file.
* Upon login or admin access, the system reads the file, loads data into objects, and updates the file upon any change.
* This allows persistent data management without dynamic memory structures.

**2.4 Program Flow**

* Splash screen (ASCII or text animation)
* Main Menu:
  + Customer Login
  + Admin Login
* Based on the role:
  + Admin can add/delete accounts
  + Customer can log in to Savings or Checking account

**3. How-To Guide**

**3.1 Running the Program**

1. Open the project in any C++ IDE.
2. Compile and run the program.
3. Follow on-screen instructions in the terminal.

**3.2 Admin Mode**

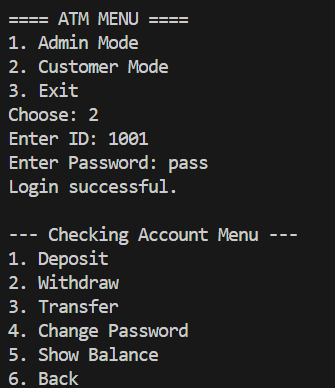
* Use predefined admin credentials (e.g., ID: admin, Pass: admin123).
* Options:
  + Add User: Enter details to create a new account.
  + Delete User: Provide user ID to remove from file.

**3.3 Customer Mode**

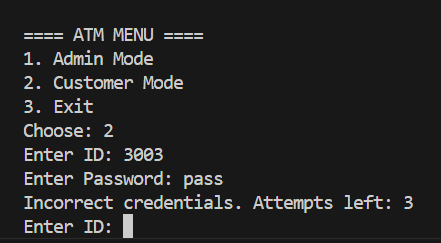
* Enter user ID and password.
* Choose account type (Savings or Checking).
* Available options:
  + **Checking**: Deposit, Withdraw, Transfer, Change Password
  + **Savings**: Deposit, Check Balance

**4. Results**

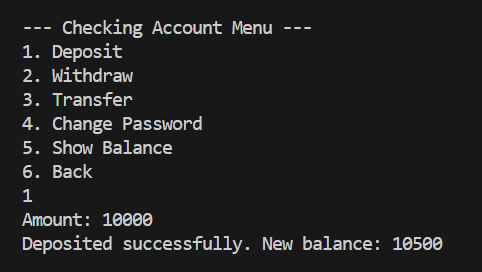
**4.1 Successful Customer Login**



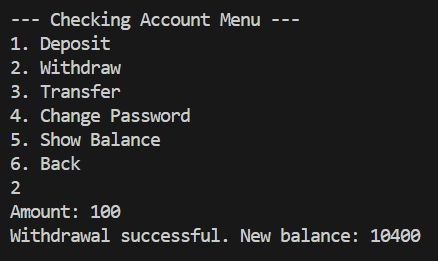
**4.2 Invalid Login**



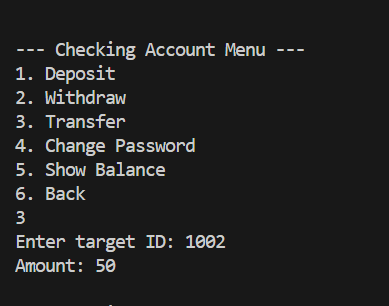
**4.3 Deposit Transaction**



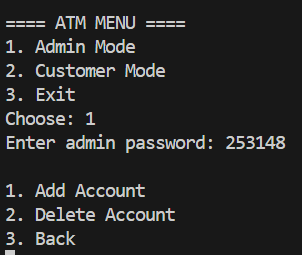
**4.4 Withdrawal Transaction**



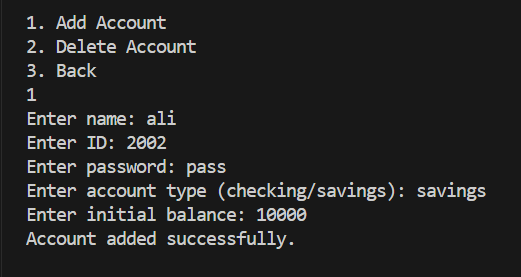
**4.5 Transfer Funds**



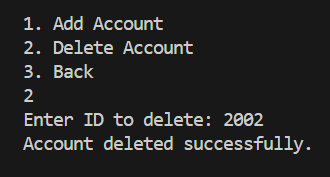
**4.7 Admin Login**



**4.7 Admin Adding New Account**



**4.8 Admin Deleting Account**



**5. Conclusion**

This console-based ATM system in C++ successfully implements file-based user account management and simulates banking features for both customers and administrators. It is modular, extendable, and sets the foundation for future integration into a graphical user interface (GUI) using C++ Builder.

**6. References**

* Cplusplus.com. (n.d.). File I/O. <https://cplusplus.com/doc/tutorial/files/>
* GeeksforGeeks. (n.d.). Object Oriented Programming in C++. <https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/>
* Stack Overflow. Community Q&A on C++ file handling and best practices.