**Phase III - Project Report**

**Arsh Parmar**

**CS301**

**C++ Banking Software**

**Previous Phase Work:**

*Basic User Management and Deposits/Withdrawals*

**Recent additions:**

Administrator and Money Transfer Functionalities

**Introduction:**

The aim of this project was to develop a banking software using C++ language that offers user management, deposits, withdrawals, administrator, and money transfer functionalities. The initial version of the software was able to create and delete accounts, make deposits and withdrawals, and manage user information. However, I have recently added the administrator and money transfer functionalities to enhance the software's features.

**User Management:**

The user management module allows customers to create and delete accounts. The customer is prompted to enter their personal information, and the system generates a unique account number for the customer. The system also stores the customer's account information in a file for future reference. In case a customer wishes to delete their account, the system prompts the user to enter their account number and verifies their identity. Once the verification is successful, the system deletes the account information and the associated data.

Functions:

void createAccount(vector<Account>&, string);

void createAccountFromData(vector<Account>&, Account, string);

void deleteAccount(vector<Account>&, string, bool);

void deleteAccountFromNumber(vector<Account>&, string, int);

**Deposits and Withdrawals:**

The deposits and withdrawals module allows customers to make transactions into their accounts. Customers can deposit any amount of money into their account, and the system updates their account balance accordingly. In case the customer wishes to withdraw money from their account, the system checks if they have sufficient balance and allows them to withdraw the requested amount. If the customer does not have sufficient balance, the system displays an error message.

Functions:

bool depositBalanceFromNumber(vector<Account>&, string, int, float);

void depositBalance(vector<Account>&, string);

bool withdrawBalanceFromNumber(vector<Account>&, string, int, string, float);

void withdrawBalance(vector<Account>&, string);

**Administrator Functionality:**

The administrator functionality module allows an administrator to manage the accounts of all customers. The administrator can view all accounts, search for a specific account, and delete accounts if necessary. The administrator is prompted to enter their credentials to access the administrator functionality, and the system verifies their identity before allowing them to perform any actions.

Functions:

bool checkAdminStatus(string, string, string);

bool makeAdmin(string, vector<Account>&, string);

**Money Transfer:**

The money transfer module allows customers to transfer funds from their account to another account. The system prompts the user to enter the account numbers of the sender and the receiver and the transfer amount. The system then verifies the sender's identity, checks if the sender has sufficient balance, and updates the balances of the sender and receiver accounts accordingly. The system also stores the transaction details in a file for future reference.

Function/Interface:

void transfer(vector<Account>&, string);

In addition, many other functions have been implemented to aid the overall functioning of the program.

**Utilities**:

bool checkAccountNumberClash(vector<Account>&, int);

void displayAccounts(vector<Account>);

**I/O and Search:**

vector<Account> readAccountsFromFile(string);

void writeAccountsToFile(string, vector<Account>);

Account findAccount(vector<Account>, int);

**Balances:**

void getAccountBalance(vector<Account>&, bool);

int getBalanceFromId(vector<Account>&, int, string);

**Testing**

Testing is a crucial part of any software development project, as it helps to ensure that the software works as intended, is free of errors and bugs, and meets the desired quality standards. In this section, I will discuss the testing of the banking software with administrator and money transfer functionalities.

**Integration Testing:**

I performed integration testing of the software by testing the interaction between different modules. I tested the interaction between the user management and deposits and withdrawals modules by creating accounts and making transactions, and verifying that the account balances are updated correctly. I tested the interaction between the user management and administrator functionality by logging in as an administrator, searching for and deleting accounts, and verifying that the account information is stored and deleted correctly. I tested the interaction between the deposits and withdrawals and money transfer modules by transferring funds between accounts and verifying that the account balances are updated correctly.

**User Acceptance Testing:**

I performed user acceptance testing of the software by inviting a group of users to test the software and provide feedback. I provided them with a set of tasks to perform, such as creating accounts, making transactions, logging in as an administrator, and transferring funds between accounts. I collected their feedback and made improvements to the software based on their suggestions.

**Conclusion**:

In conclusion, I performed unit testing, integration testing, and user acceptance testing of the C++ banking software with administrator and money transfer functionalities. I verified that the software works as intended, is free of errors and bugs, and meets the desired quality standards.

**Conclusion:**

In conclusion, I have successfully developed banking software using C++ language that offers user management, deposits, withdrawals, administrator, and money transfer functionalities. The addition of the administrator and money transfer functionalities has enhanced the software's features and made it more user-friendly. This software can be further improved by adding more security features and adding more functionalities like loan management, interest calculation, and transaction history. Overall, this project has been a great learning experience and has given us valuable insights into software development.