1. Project Overview (1 Paragraph)

## Team:

Group Members: Calvin White, Pratham Snehi

Group Number: 12

## Systems purpose:

The main goal of this project is to create a banking system where it keeps a list of accounts both checking and saving as well as a transaction history of account creation, deposits, withdraws and transfers.

## Key functionalities:

Creating bank accounts that are either checking or savings. Withdrawing, depositing, or transferring money between accounts. On checking accounts there is a overdraft fee that if you withdraw past it, it charges your account. For savings account you need to start with at least 100$ to create the account, savings accounts also can gain monthly interest. A transaction logger keeping track of anything the account does(creation, withdraw , deposit, transfer, overdraft).

2. Class Descriptions

Abstract Class: BankAccount  
Attributes:

accountNumber (Unique identifier)

holderName (Customer’s name)

balance (Current balance)

accountType (Savings or Checking)

Methods:

deposit(double amount)

withdraw(double amount)

getBalance()

displayAccountInfo()

recordTransaction(String message)

Class: SavingsAccount (Extends BankAccount)

Unique Attributes:  
All of them are inherited from BankAccount  
Unique Methods:  
chargeOverdraft()

Class: CheckingAccount (Extends BankAccount)

Unique Attributes:  
All of them are inherited from BankAccount  
Unique Methods:

applyInterest()

Interface: TransactionLogger

Attributes:

None  
Abstract Methods:

logTransaction(String message)

Class: Bank

Attributes:

BankAccountArray (an arraylist of all the bank accounts that are created)  
Methods:

createAccount()  
authenticateUser(String username, String password)  
deposit(int accountNumber, double amount)  
withdraw(int accountNumber, double amount)  
 transferFunds(int fromAccount, int toAccount, double amount)  
viewTransactionHistory(int accountNumberdisplayAllAccounts()

Class: Main

Attributes:

None

Methods:

main()

3. Implementation Details (Short Summary)

When you run the main you enter a loop where you can select choices using a switch case. You can create accounts, deposit/withdraw/transfer money within them, view history, get account info, or exit the program. Within the choices, for example create account, it will prompt you for your name, type of account, and starting deposit. Using what you input it will first check to make sure all the data you used works for your account type (for savings there is a $100 minimum) and will either tell you the issue or create your account and tell you the account number.

4. Test Cases (Table Format)

|  |  |
| --- | --- |
| Action | Output |
| Create Account | [Type of Account] created successfully with a balance of [Initial Deposit] |
| Deposit Money | Deposit Successful. New Balance is [New Balance] |
| Unsuccessful Withdraw Money (Saving) | Insufficient funds! |
| Overdraft Withdraw Money (Checking) | Successful with overdraft fee of $10 |
| Transfer Funds | Withdrawn [Amount], new amount remaining: [new Balance]. Deposited [Amount], remaining: [new Balance] |

5. Challenges & Solutions (Brief Explanation)

One of the challenges we faced was choosing a system for the account number, at first we used a UUID which would work great for a real bank but with the way we had our methods using the account numbers it would have been to hard to demonstrate, so instead we landed on a static count of accounts made and just added one for each new account, so every account would have a new account number.

Another challenge we faced was having the user select what option to do, originally, we had it so you typed out the full option and we found that tedious. Instead, we opted for a number system where you only had to select the number for the option you wanted.

We ran into the issue that if you input letters instead of numbers on the menu it would crash the program, then we remembered exception handling and implemented try catches on all of the inputs.

6. Conclusion & Future Enhancements (Optional)

As of now you must create new accounts every time you run the file, so in the future we would like to have it save the accounts and all the other information like transaction history and balance into a file or database.