**Project Overview**

(Anthony and Owyn)

This Bank\_Mangement\_System project is meant to be a simple Banking system. It has simple functionalities like creating an accounts, depositing funds, withdrawing funds, Transferring funds from one account to another, and view transactions history.

**Class Descriptions**

BankAccount (Abstract and implements TransactionsLogger)

* Attributes:
  + Int AccountNumber
  + String HolderName
  + Double Balance
  + String accountType
  + ArrayList<String> transactionLogs
* Methods
  + deposit(double amount): Adds to the balance
  + withdraw(double amount): Abstract
  + getBalance(): Returns balance
  + displayAccountInfo(): Displays all of account Info
  + logTransaction(String message): Add message to arraylist
  + printLogTransaction(): Prints transaction history
  + generateInterest(): Prints interest gain (if possible)
  + isAccountNum(): check if account number matches

SavingsAccount (Extends BankAccount)

* Attributes:
  + Extends BankAccount
* Methods:
  + @Override withdraw(double amount): Withdraws from balance only if $100 would remain in the account
  + @Override generateInterest(): Adds 1% interest

CheckingsAccount (Extends BankAccount)

* Attributes:
  + Extends BankAccount
* Methods:
  + @Override withdraw(double amount): Withdraws from balance, Adds overdraft fee if needed

TransactionsLogger (Interface)

* Attributes:
  + None
* Methods:
  + logTransactions(string message):

Bank

* Attributes:
  + ArrayList<BankAccount> bankAccounts
* Methods:
  + createAccount(String name, double balance, Int type): creates the corresponding account
  + findBankAccount(Int accountNumber): Finds account
  + deposit(int accountNumber, double amount): Finds account then deposits amount
  + withdraw(int accountNumber, double amount): Finds account then withdraws amount
  + transferFunds(int FromAccount, int ToAccount, double amount): Finds boths accounts then withdraws from the first and deposits into the second
  + viewTransactionHistory(int accountNumber): Finds account then prints prevoius transaction
  + displayAllAccounts(): Prints all accounts info
  + generateInterest(int accountNumber): Finds account then add interest

**Implementation Details**

The central class in this project is the Bank class. Inside the Bank class is an ArrayList of all the BankAccount classes for the Bank. Each of the methods within the Bank class then interacts with a BankAccount class or creates a new one in the ArrayList. The BankAccount class is an Abstract class used as a baseline for our two types of bank accounts: SavingsAccount and CheckingsAccount. Both use most of the same methods from the BankAccount class but they also override the abstract withdraw method to function as their account type should. BankAccount also implements the TransactionsLogger interface and uses the logTransactions method.

**Test Cases**

|  |  |
| --- | --- |
| **Action** | **Expected Output** |
| Create Account | "Account created successfully." |
| Deposit Money | "Deposit successful." |
| Withdraw Money (Savings) | "Insufficient funds!" |
| Withdraw Money (Checking) | "Overdraft used." |
| Transfer Funds | "Transfer successful." |
| View Transaction History | Displays past transactions. |

**Challenges and Solutions**

The biggest issue we faced with this project was unclear instructions on the class/method structure. Initially in the assignment the Bank class was supposed to implement the TransactionLogger interface but the BankAccount class was still supposed to have a logTransaction method. This created some confusion on how we were supposed to log transactions. We fixed this problem by ignoring the instructions and building a structure that made sense to us (BankAccount implementing TransactionsLogger) and then later the instructions were updated to match the structure we had already created.