

Deliverables

(Refer to the last page for test accounts to run the program!)

1. Minimum Requirements:

Customer -

- Login using a unique Account Number, their First Name and Last Name. If any of the three are typed in wrong, the customer will not be able to login. Note that First Name and Last Name starts with a capital letter, followed by lowercase letters.
- Customer then is able to select a specific account they want to perform an action on, either *Savings* or *Chequing*.
- For each of the accounts (*Savings* or *Chequing*), customer is able to choose to deposit, withdraw or transfer between the accounts.
- Each option has a warning message that makes sure that the customer does not attempt to deposit, withdraw or transfer when their amounts are insufficient.
- When withdrawing from the *chequing* account and the account balance will drop lower than \$1000, a warning message will appear to give the customer an option to proceed with the transaction despite the fee of \$2, or not execute the action.

Manager -

- Login procedure is the same as Customer.
- Manager will have the option to visit a particular customer's account, or to print information of the bank.
- Bank information option will list customer's account number, first and last name, total number of customers within the bank, and the total funds within the bank.
- To visit a customer's account, the manager will need the account number, first and last name of the customer to modify their details.
- Manager will be able to open/close accounts within the customer's accounts.

Maintenance -

- Login same as above
- Maintenance personnel will be able to turn on or off trace. Traces will print customer execution trace including the date of access into a text file named "trace.txt"

2. Enhancements:

Data Persistence -

- All user login information and bank account information is logged in three separate text files: "customer.txt" (contains customer login information and accounts), "managers.txt", and "maintenance.txt". All modifications such as deposit or withdraw will also affect the text files.

Customer Deletion -

- Manager when logged in has the option to delete a customer off the database if the customer decides to leave the bank. Changes will be made in the customer.txt file, and only the manager can perform this action.

Additional Savings and Chequing Accounts -

- Customer once logged in is able to select if they would like to access a certain set of account at startup. Different amounts are stored in the accounts for multiple purposes. ex)

```
***** Jieni's Bank! *****
Welcome Jieni!

Which account would you like to access today?

Account 1:
Balance in Savings: 2123.12
Balance in Chequing: 6543.00

Account 2:
Balance in Savings: 353.00
Balance in Chequing: 21312.00

Please type in the number that you would like to access.

*****
|
```

Creating new Customers -

- Manager can create a new customer once logged in. Manager will have to enter details of the customer such as their First name, Last name and amounts to deposit into the chequing and savings accounts. Note that manager does not get to choose the account number since it has to be unique and will be generated by the system.
- The created customer can be accessed immediately, and their information will be logged in "customer.txt"

3. System Requirements:

- Application is developed on the Mac, and is recommended to run program on a Mac. To run the program, go to Terminal > then navigate to the directory > then type in ./BankingSystem to start the program.

4. Scenarios:

- Opening a new account (Manager):
 - Manager met up with a new customer and gets their details to enter into the bank system.
 - Manager login into the system
 - Customer provided the following information:
 - Fname - "Jaisen"
 - Lname - "Bynoe"
 - Savings - 200
 - Chequing - 4000

(Picture ordered from left to right)

```

***** Welcome to Jieni's Bank! *****

Choose an option below to begin banking:

1. Customer Login
2. Manager Login
3. System Maintenance
4. Exit

*****
2

***** Jieni's Bank! *****

Choose an option below:

1. Login

*****
1

***** Jieni's Bank! *****

Please enter the account number:
1131
Please enter the first name:
Vanessa
Please enter the last name:
Hou

***** Jieni's Bank! *****

***** Jieni's Bank! *****
Welcome Vanessa
Enter the First name of the customer: Jaisen
Enter the Last name of the customer: Bynoe
Enter an amount to deposit to Savings: 200
Enter an amount to deposit to Chequing: 4000

***** New customer is now added *****
3

***** Jieni's Bank! *****
Welcome Vanessa
1. Access a customer account
2. View all accounts within Bank
3. Create new Customer Account
4. Exit

*****
3

***** Jieni's Bank! *****
New customer is now added
*****

customer.txt
1000,Jieni,Hou,1121,2412,121,3523.21
1001,Alex,Tan,442,100.12,5212,23
1002,Liam,Sue,1,23.32,3512,55
1031,Paul,Tan,1212,3412.83
1042,Adam,Freeman,1421,123
1213,Jerry,Lim,2222,1111.11
1545,Jaisen,Bynoe,4000.00,200.00

```

- Opening logs (Maintenance):
 - Maintenance personnel logs into system
 - Turns on the trace by selecting option
 - Logs out of the system
 - Maintenance guy checks logs in trace.txt

```

***** Welcome to Jieni's Bank! *****
Choose an option below to begin banking:
1. Customer Login
2. Manager Login
3. System Maintenance
4. Exit
*****
3
***** Jieni's Bank! *****
Choose an option below:
1. Login
*****
1
***** Jieni's Bank! *****
Please enter your account number:
1111
Please enter your first name:
Mark
Please enter your last name:
Hou
*****
***** Jieni's Bank! *****
Welcome Mark!
What would you like to do today?
1. Turn On Trace
2. Turn Off Trace
3. Logout
*****
1
Trace is now on. You can logout now.

```

on Trace.txt:

Tue Oct 13 00:10:50 2015

1000

Jieni

Hou is now logged in 1

Customer chose to transfer in chequing 2

Customer chose to deposit in chequing 1

Customer changed accounts 4

Customer chose to change main account 3

Customer chose to exit 4

End of session.

- Chequing account processing fees (Customer):
 - Customer Logs onto account 1 on their laptop
 - Selects Chequing account
 - Selects to Withdraw from Chequing.
 - Enters \$300
 - Warning message pops up to check if customer wants to proceed
 - Customer proceeds with transaction

```

***** Welcome to Jieni's Bank! *****

Choose an option below to begin banking:

1. Customer Login
2. Manager Login
3. System Maintenance
4. Exit

*****
1

***** Jieni's Bank! *****

Choose an option below:

1. Login

*****
1

***** Jieni's Bank! *****

Please enter your account number:
1000
Please enter your first name:
Jieni
Please enter your last name:
Hou

*****

***** Jieni's Bank! *****

Welcome Jieni!

Which account would you like to access today?

Account 1:
Balance in Savings: 2412.00
Balance in Chequing: 1121.00

Account 2:
Balance in Savings: 3523.21
Balance in Chequing: 121.00

Please type in the number that you would like to access.

*****
1

***** Jieni's Bank! *****

Account #: 1

1. Savings
2. Chequing
3. Change Account
4. Exit

*****
2

***** Jieni's Bank! *****

You are currently in your Chequing.
Your balance in chequings is: 1121.00

1. Deposit
2. Withdraw
3. Transfer from chequing to savings
4. Back

*****
2

***** Jieni's Bank! *****

Please enter amount to withdraw:
300
Are you sure you would like to withdraw? You will be charged an extra $2 fee for transactions that
will cause your account to be below $1000
1. Yes
2. No
1
Your new balance in Chequing is: 819.00

*****

```

- Overall bank information (Manager):

- Manager login onto system on computer.
- Selects option to see all customer information and total funds within bank

```

***** Welcome to Jieni's Bank! *****

Choose an option below to begin banking:

1. Customer Login
2. Manager Login
3. System Maintenance
4. Exit

*****
2

***** Jieni's Bank! *****

Choose an option below:

1. Login

*****
1

***** Jieni's Bank! *****

Please enter the account number:
1131
Please enter the first name:
Vanessa
Please enter the last name:
Hou

*****

***** Jieni's Bank! *****

Welcome Vanessa
1. Access a customer account
2. View all accounts within Bank
3. Create new Customer Account
4. Exit

*****
2

Customer: 1
1000 Jieni Hou
Account 1:
Balance in Savings: 2412.00
Balance in Chequing: 1121.00

Account 2:
Balance in Savings: 3523.21
Balance in Chequing: 121.00

Customer: 2
1001 Alex Tan
Account 1:
Balance in Savings: 100.12
Balance in Chequing: 442.00

```

```
Account 2:  
Balance in Savings: 23.00  
Balance in Chequing: 5212.00  
  
Customer: 3  
1002    Liam    Sue  
Account 1:  
Balance in Savings: 23.32  
Balance in Chequing: 1.00  
  
Account 2:  
Balance in Savings: 55.00  
Balance in Chequing: 3512.00  
  
Customer: 4  
1031    Paul    Tan  
Account 1:  
Balance in Savings: 3412.83  
Balance in Chequing: 1212.00  
  
Customer: 5  
1042    Adam    Freeman  
Account 1:  
Balance in Savings: 123.00  
Balance in Chequing: 1421.00  
  
Customer: 6  
1213    Jerry    Lim  
Account 1:  
Balance in Savings: 1111.11  
Balance in Chequing: 2222.00  
  
Customer: 7  
1545    Jaisen  Bynoe  
Account 1:  
Balance in Savings: 200.00  
Balance in Chequing: 4000.00  
  
Total Customers: 7  
Total bank funds: 30247.59
```

- Transferring funds (Customer):
 - Customer login to account 2 on ATM
 - Selects savings account
 - Selects option to transfer from savings to chequing
 - Transfers 40
 - System displays the funds that are transferred and gives summary

***** Welcome to Jieni's Bank! *****

Choose an option below to begin banking:

1. Customer Login
2. Manager Login
3. System Maintenance
4. Exit

1

***** Jieni's Bank! *****

Choose an option below:

1. Login

1

***** Jieni's Bank! *****

Please enter your account number:

1000

Please enter your first name:

Jieni

Please enter your last name:

Ho

Login Unsuccessful. Please Try Again.

***** Jieni's Bank! *****

Please enter your account number:

1000

Please enter your first name:

Jieni

Please enter your last name:

Hou

***** Jieni's Bank! *****

Welcome Jieni!

Which account would you like to access today?

Account 1:

Balance in Savings: 2412.00

Balance in Chequing: 1121.00

Account 2:

Balance in Savings: 3523.21

Balance in Chequing: 121.00

Please type in the number that you would like to access.

2

***** Jieni's Bank! *****

Account #: 2

1. Savings
2. Chequing
3. Change Account
4. Exit

1

***** Jieni's Bank! *****

You are currently in your Savings.

Your balance in savings is: 3523.21

1. Deposit
2. Withdraw
3. Transfer from savings to chequing
4. Back

3


```

***** Jieni's Bank! *****

Please enter amount to transfer:
40
Transfer was successful.
Savings: 3483.21
Chequing: 161.00

*****

```

5. Implementation of Enhanced Requirements:

Data Persistence -

- This is made possible through the use of vectors to store information regarding the Customers/Managers/Maintenance. For each user, I created a vector object that would store all the information for each different user. Reason behind using vector instead of other data structures is its easy indexing and implementation. All login/verification information is stored through the reading of a text file for the specific user (ex.) customer.txt, manager.txt and maintenance.txt).
- After the actions by a Customer, for example withdrawing a specific amount out of a chequing account, their account information will be modified using the removeAccount method, which takes away the original line of information and prints the new customer information into the file using the ifstream and ofstream functions.

Creating New Customers -

- New customers can be added through the Manager accounts and is implemented by prompting the manager for the first name, last name and amounts they would like to deposit into chequing or savings.
- Manager does not have the option to choose an account number. It will be generated automatically by the rand() function (I did $1000 + \text{rand}() \% 1000$ because I wanted the numbers to be within the 1000s range). Account number will also be adjusted by 1 if the same number is found within the database.

Customer Deletion -

- When a customer decides to leave the bank, they are able to request the managers to delete their accounts completely from the database. Managers can easily proceed with this operation through the option to delete the customer's account. This feature is implemented using the deleteAnAccount method, where the ifstream and ofstream functions are used.

Additional Savings and Chequing -

- This feature is made possible through the use of vectors within the User class. By providing additional space for the customer, they can now open accounts for their business savings, or additional accounts for different other purposes. Again the vector data structure has made accessing easy for these accounts.

6. What was learnt through this project:

- Through the development of this Banking System, I had noticed that my knowledge in System Design was lacking. If I had the chance and a little more time to correct the

project, I would've re done the entire project. Before beginning, I would've sketched a clear system design for a better organization of code. I understand that main.cpp should only have under 50 lines of code, but my eagerness to begin coding had prevented me from proper planning. However, I had learned a lot more about C++ coding concepts, and loved the separation between method declarations and implementations. C++ syntax was widely different from Java, however it was a lot neater in a way that there were different files, .h,.cpp and also the main.cpp. The project was a great way to lead into the group project as it has increased my knowledge for the language, and I am more comfortable with using it within the software that we are going to build.

Some handy test accounts that you can use to test my program (all of these can be found in the text files, unless you did a delete operation throughout the program and deleted the accounts):

** Make sure that you enter these as you see them below. **The input is case sensitive.** **

Customers (customers.txt):

Account #: 1000

Fname: Jieni

Lname: Hou

Managers (managers.txt):

Account #: 1131

Fname: Vanessa

Lname: Hou

Maintenance (maintenance.txt):

Account #: 1111

Fname: Mark

Lname: Hou