CISC 327Course Project Assignment #2

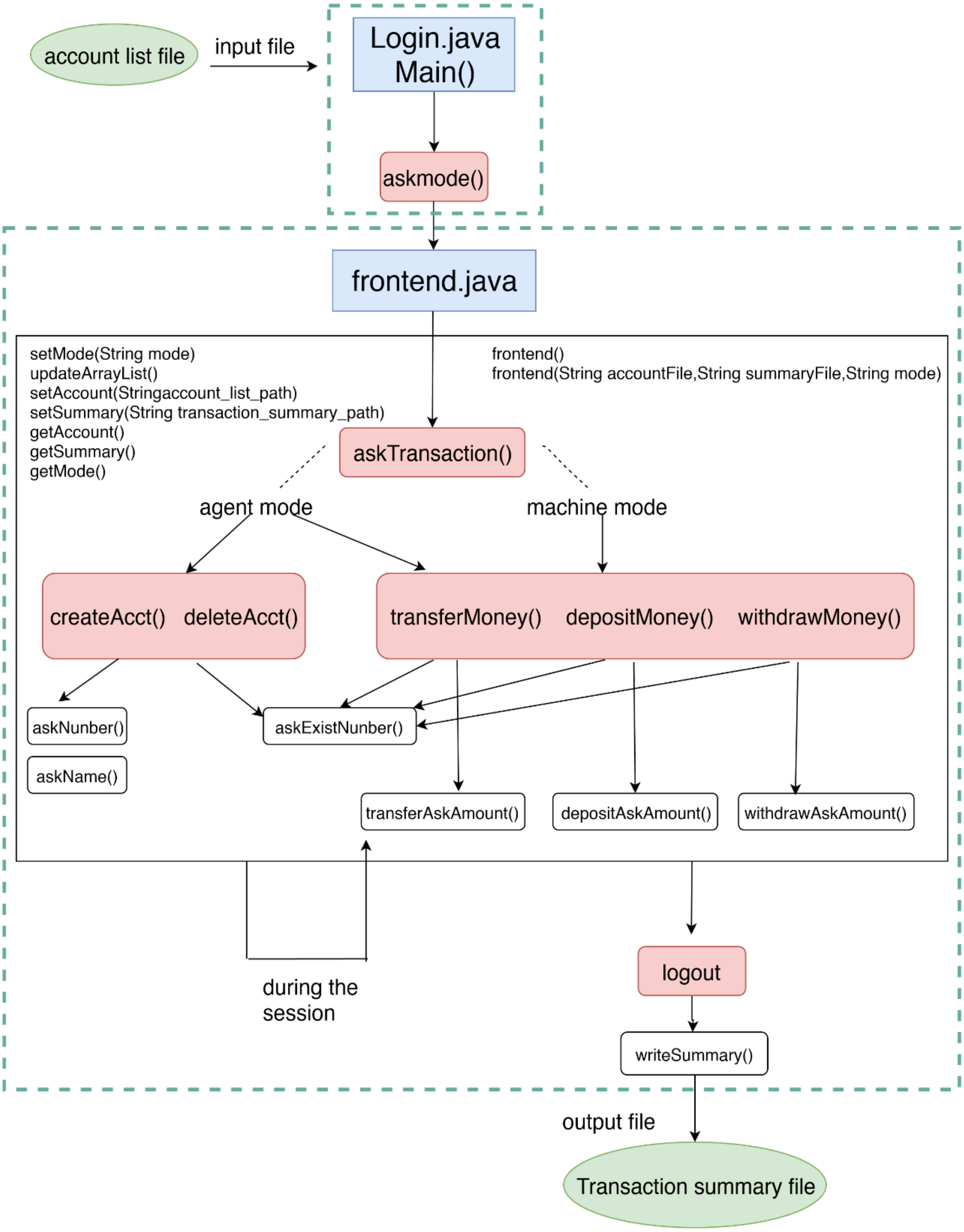
Front End Rapid Prototype

Company: Canada Trust

Yudong Zhou (20083467)
Yitong Liu (20028039)
Runze Yi (20073329)
Tong Bu (20079649)

Overall structure

The overall structure of our solution is shown in the picture below. We use the blue box to represent the two main classes of our project. They are surrounded by the green dotted box to indicate the scope of each class. The Login class reads the account list file as the input file while the Frontend class writes the transaction summary file for the day as an output file. The operation and transaction functions are colored into pink and white in the light of their importance, and other small functions are also listed inside the green box of their class.



Classes and methods

The table below shows all classes and methods in our source code followed by a brief description for each of them.

Class Name: Login.java	
Function Name	**Description**
main()	Main function of the program that reads the account list file as input file, and prompt user to Login
askMode()	Function that asks the Login mode (machine/agent). Takes no parameters, takes user input from console of the choice of mode, and returns that as string.

Class Name: Frontend.java	
Function Name	**Description**
Frontend ()	This is the non-parameter constructor for Frontend class
Frontend (String accountFile,String summaryFile,String mode)	This is the Frontend 3 parameter constructor, it takes 3 parameters. The paths of accountList file, the path of summary file and the string mode. Calls 4 functions within it. Stores the paths as attribute and updates information in accountlist file to attribute
asktTransaction()	Takes user input from console and calls different functions upon different situations. Will call createAccount(), deleteAccount(), writeSummary(), depositMoney(), withdrawMoney() or transferMoney(). Returns an int to determine whether to end session.

createAccount()	Function for create account transaction. Takes no parameters, calls askNumber() and askName() to obtain user input, and save information to the transaction summary file.
deleteAccount ()	Function for delete account transaction. Takes no parameters, calls askExistNumber () and askName() to obtain user input, and save information to the transaction summary file.
depositMoney()	Function for deposit transaction. Takes no parameters, will call askExistNumber() and depositAskAmount() to get information, and save information for the transaction summary file.
withdrawMoney()	Function for deposit transaction. Takes no parameters, will call askExistNumber() and withdrawAskAmount() to get information, and save information for the transaction summary file.
transferMoney()	Function for deposit transaction. Takes no parameters, will call askExistNumber() and transferAskAmount() to get information, and save information for the transaction summary file.
writeSummary()	Function that writes information stored in the temporary summary file to the one that's to be output. Also write valid account list file from updated AccountList array list attribute
askNumber()	Function that asks user to enter an unexist account number while creating account. Takes no parameters, will take user input from console and verify validity of that information. Returns a string of account number.
askName()	Function that asks user to enter an alphanumeric account name while creating account. Takes no parameters, will take user input from console and verify validity of that information. Returns a string of account name.
askExistNumber()	Function that asks user to enter an exist account number. Takes no parameters, will take user input from console and verify validity of that information. Returns a string of account number.
depositAskAmount()	Function that checks the deposit amount with the constraints. Takes no parameters, will take user input from console and verify validity of that information. Returns an int of amount to be performed transaction on.
withdrawAskAmount()	Function that checks the withdraw amount with the constraints. Takes no parameters, will take user input from console and verify validity of that information. Returns an int of amount to be performed transaction on.
transferAskAmount()	Function that checks the transfer amount with the constraints. Takes no parameters, will take user input from console and verify validity of that information. Returns an int of amount to be performed transaction on.

setMode(String mode)	Function that sets the front-end mode, either agent or machine. Takes one string argument as the mode, no returns.
updateArrayList()	Updates information of the input account list file to the attribute AccountList arrayList. Takes no parameters, returns nothing.
setAccount(String account_list_path)	Function that sets the valid account list file. Takes the path of input account list file as string argument, returns nothing.
setSummary(String transaction_summary_path)	Function that sets the summary transaction file. Takes the path of input summary list file as string argument, returns nothing.
getAccount()	Function that gets the valid account list file. Takes no parameters, returns string path of input account list file.
getSummary()	Function that gets the transaction summary file. Takes no parameters, returns string path of input summary file.
getMode()	Function that gets the front-end mode. Takes no parameters, returns string mode information.

Java Class definition:

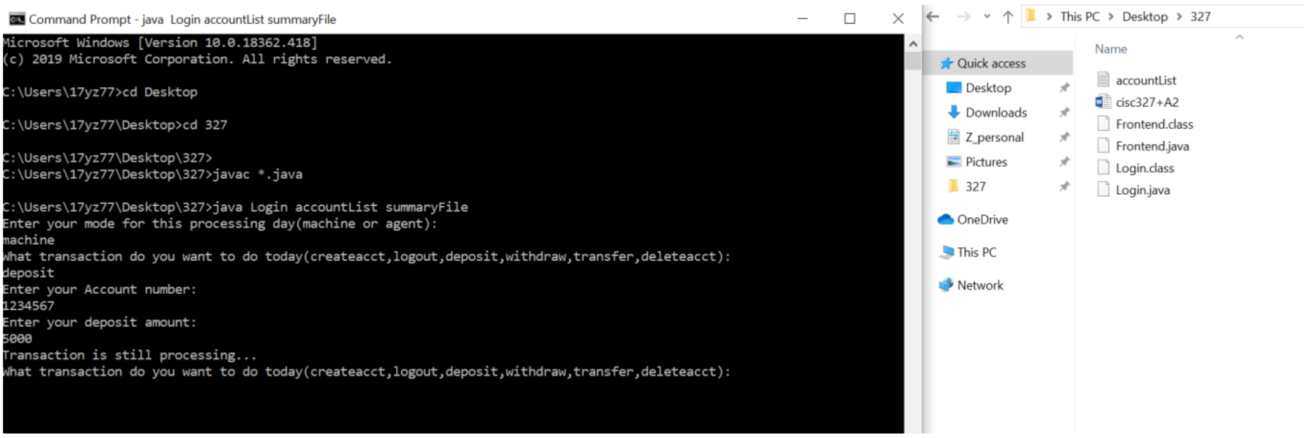
Frontend:

Frontend is the class that contains different feature's implementation. In this class, Frontend will ask user for different transaction and checking if user prompt correctly, and after logout, the summary transaction file will be written.

Login:

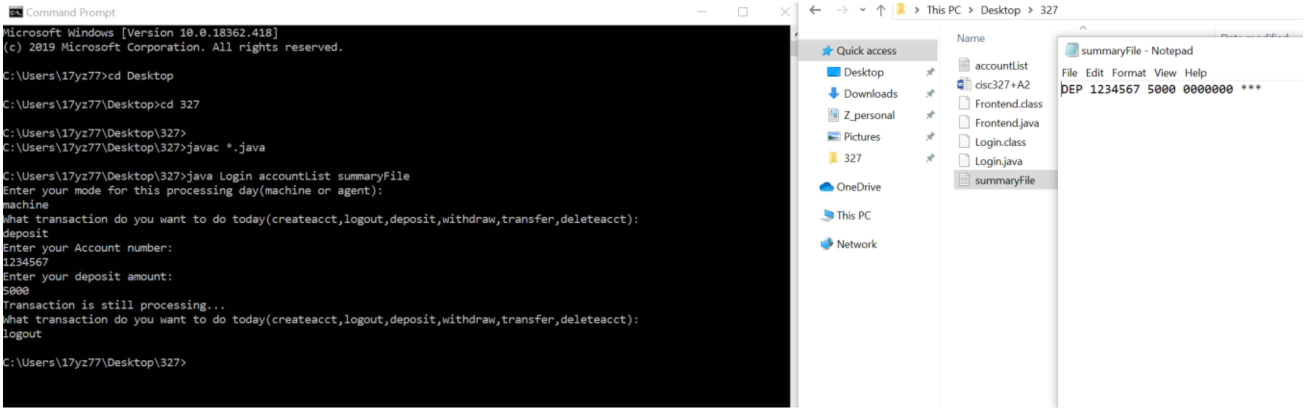
Login is the main class that you need two arguments to run, first argument is the valid account list that you need to have, second argument is a non-exist file name that you want to write your summary transaction file on.

Running is like this:



```
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\17yz77>cd Desktop

C:\Users\17yz77\Desktop>cd 327

C:\Users\17yz77\Desktop\327>
C:\Users\17yz77\Desktop\327>javac *.java

C:\Users\17yz77\Desktop\327>java Login accountList summaryFile
Enter your mode for this processing day(machine or agent):
machine
What transaction do you want to do today(createacct,logout,deposit,withdraw,transfer,deleteacct):
deposit
Enter your Account number:
1234567
Enter your deposit amount:
5000
Transaction is still processing...
What transaction do you want to do today(createacct,logout,deposit,withdraw,transfer,deleteacct):
```

After logout, summaryFile will be written:



```
Microsoft Windows [Version 10.0.18362.418]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\17yz77>cd Desktop

C:\Users\17yz77\Desktop>cd 327

C:\Users\17yz77\Desktop\327>
C:\Users\17yz77\Desktop\327>javac *.java

C:\Users\17yz77\Desktop\327>java Login accountList summaryFile
Enter your mode for this processing day(machine or agent):
machine
What transaction do you want to do today(createacct,logout,deposit,withdraw,transfer,deleteacct):
deposit
Enter your Account number:
1234567
Enter your deposit amount:
5000
Transaction is still processing...
What transaction do you want to do today(createacct,logout,deposit,withdraw,transfer,deleteacct):
logout

C:\Users\17yz77\Desktop\327>
```

```
DEP 1234567 5000 0000000 ***
```