CSCI 3060U/SOFE 3980U – Winter 2016
Course Project Assignment #6 - Integration and Delivery
Albert Fung 100520898
Stuart Calverley 100522058
Janahan Mathanamohan 100523201
Andrew Lau

Location	Problem	Fix
Frontend/Backend	The backend works by checking the merged file until it sees a certain string of character which in the backends case was “00 “. So because the frontend never wrote that line to the transaction file the backend never terminated.	After merging the files though the script, the script also adds the terminate line to the end of the merged transaction file so the Backend can terminate when there are no more transactions to process.
Backend	Was not able to read the merged transaction file created from the script	Changed the location in the Backend code on where it looks and the file it is to read from.
Backend	Was trying to merge multiple transaction files	Deleted the code and let the script merge the files.
Backend	Did not delete accounts when the delete transaction was passed in	The code was only deleting the account from the Map not the ArrayList, so we changed it so it would delete from both.
Backend	Would not enable or disable the account.	We had to trim the name so when checking to see if an account was disabled or enabled it would be able to find the name.
Backend	Transaction fee was still being applied for admin.	Created a flag for admin to be passed around to set transaction fee to zero.

1.) We used Stuart Calverley's front end who was previously part of Hard Water and we used Janahan's back end who was previously part of DeliriumApple.
2.) The daily script is run by typing ./daily (a number between 1-5) as an argument. The number determines which day of transactions are run. The input files are located in frontendUse folder and is organized into their respective days. The daily script writes a merged file that the backend uses in the main directory where the scripts are located, creates a copy of that day's merged transactions in a folder called MergedTransactionFiles and the filename of the merged file contains the day number. The daily script also calls the backend, and the backend writes an updated MasterAccountsFile and CurrentAccountFile to the directory Application/Accounts/. The daily script creates a copy of the MasterAccountFile after every day and stores it in the Folder MasterFile. The file name tells you which day the MasterAccountFile corresponds to. The weekly script just runs the daily script five times and does not write any information to any files.
3.) The set of transaction session inputs can be located in the frontendUse folder where they are separated into day folders.
4.) The merged transaction file for the daily runs are stored in a folder called MergedTransactionFiles and each text file can be distinguished by the number in the file name. If the filename contains the number "1", that merged file corresponds to the inputs of day 1, etc.
5.) The folder MasterFiles contains the Master Bank Accounts File after each of the five daily runs.

Daily script:

```
#!/bin/bash
# Copyright 2016 ALbert, Janahan, Stuart copyright
<<COMMENT
The daily script is a scirpt that will simulate a day of transactions.
Ensure that if you are running this script you provide a number between 1 to 5 as a parameter.
The reason for this is because based on the number you enter it will run that days tests, if you are
running the script for the first time make sure you run day 1 test first as, it is the transaction that
create the accounts.
After creating the accounts you can run any day's tests and it will complete the test as it should.
Another note if you are running the daily script before running the weekly script make sure the folder
Accounts which can be found by:
        Application/Accounts
                has the txt files
                        CurrentAccountFile.txt
                        MasterAccountFile.txt
                                These files need not have any information in them they just need to be created
                                If the files are already there that is fine and you can leave them there.
To run the script ensure that you are in the current directory of the script.
To run it:
        change permissions if not already done:
                chmod +x daily.sh (number between 1-5)
        start execution:
COMMENT

# Path locations we need
appLocation="Application/main.o"
# Picks the folder for the current day we are on
testData="frontendUse/$1"
tfFiles="frontendUse/TF/"
backend="FinalBackend/"
```

```
# Builds the banking system
g++ -std=c++11 $(pwd)/Application/main.cpp -o $(pwd)/Application/main.o
cd FinalBackend/
javac Backend.java
javac BankAccount.java
javac Validator.java
cd ..

# Runs the input file though the Banking program and stores the transaction files in a
folder
function transactions() {
	# Deletes the merged file if one exists
	rm $(pwd)/merged.txt
	# Clears the folder where the transaction information is located
	rm $(pwd)/frontendUse/TF/*

	# Loops for the number of files in the current directory
	for inputFiles in $testData/*.txt
	do
		# path location to where the transaction files should be written
		transactionName=$(pwd)/frontendUse/TF/$(basename
${inputFiles%-*}.txt)
		# runs the program with the input files and writes the transaction files to
the path specified by $transactionName
		./$appLocation ./Application/Accounts/CurrentAccountFile.txt
$transactionName $inputFiles
	done
}

# Runs the backend of the banking system
function backend() {
	# Sets the location for where the merged file is located
	mergedFile=$(pwd)/merged.txt
	cd FinalBackend/
	# Runs the backend and passes in the merged file which contains all the
transactions that happened that day
	java Backend $mergedFile
}
	transactions
```

```
	# Copies all the text in the files located at $tfFiles (this information is the
transaction details)
	# to a new file called merged
	cat $tfFiles*.txt >> merged.txt
	# Adds on the terminate symbol (the backend we recieved requires this)
	echo "00                                    " >> merged.txt
	backend
	cd ..
	#the master file after every day is saved and written to a folder called MasterFiles
	cat Application/Accounts/MasterAccountFile.txt >
MasterFiles/MastersFileDay$1.txt
```

Weekly script:

```
#!/bin/bash
# Copyright 2016 ALbert, Janahan, Stuart copyright
<<COMMENT
The weekly script is a script that will simulate a full 5 days of transactions
when first running this script it will clear/remake any MasterAccountFile and
CurrentAccountFile that is in there
meaning you always get a fresh start.
To run the script ensure that you are in the current directory of the script.
To run it:
        change permissions if not already done:
                chmod +x wekkly.sh
        start execution:
COMMENT


#Clears/Creates the MasterAccountFile and CurrentAccountFile
> $(pwd)/Application/Accounts/MasterAccountFile.txt
> $(pwd)/Application/Accounts/CurrentAccountFile.txt

#The current day is initalized to day 1
COUNTER=1
#The loop will run until it reaches 6
while [  $COUNTER -lt 6 ]; do
        # Will run the script daily with the current day it is on
    ./daily.sh $COUNTER
    # Increases the day by one
    let COUNTER=COUNTER+1
done
```