Course Project Assignment #5 - Back End Unit Testing

Jeremy Kwok 100341977
Mohannad Abdo 100523158
Mirna Zohiry 100535658

Table of Cases (Code Coverage)

Class: Account
Cases:
getters and setters

Class: Transaction
Cases:
getters and setters

Class: Validator

Cases:

Method Tested	Test Description
sessionLoggedIn	When logged in is true
sessionLoggedIn	When logged in is false
findAccount	When the target account exists in our list
findAccount	When the target account does not exist
findNextNumber	One test will cover all the code in this function
accountJustCreated	One test will cover all the code in this function
enoughBalance	When the account has more than the given balance
enoughBalance	When the account does not have more than the given balance
accountMatchName	When the given string matches the account holder
accountMatchName	When the given string does not match the account holder
checkCompany	When the given company is valid
checkCompany	When the given company is invalid

Class: Backend

Method Tested	Test Description
applyWithdrawal	One test will cover all the code in this function
applyPayBill	One test will cover all the code in this function
applyDeposit	One test will cover all the code in this function

applyCreate	Successful creation
applyCreate	Creation without admin credentials
applyDelete	Successful Deletion
applyDelete	Deletion without admin credentials
applyDisable	Successful Disable
applyDisable	Attempt to disable disabled account
applyEnable	Successful Enable
applyEnable	Attempt to enable enabled account
applyChangePlan	Successful change plan
applyChangePlan	Attempt to change plan on student account
init	Will cover load and handle transactions
applyLogin	Logging in when logged in already
applyTransfer	Successful transfer
applyTransfer	Transaction last in list
applyTransfer	Single transfer transaction

applyLogin and applyEndofSession will be covered incidentally by other test cases.

Class: Writer

Method Tested	Description
write	Normal runthrough of write to success. Will include writing both student and non student accounts to cover both sides of the if statement
write	Covers exception when reading file
Writer constructor	Normal runthrough of the writer constructor to success. Will include reading both student and non student accounts, and reading both student and non student transactions to cover all branches of those if statements.
Writer constructor	Covers exception: error reading master bank accounts account file.
Writer constructor	Covers exception: IOException when loading mergedbanktransaction file
Writer constructor	Covers exception when trying to open the bank account transaction files.

Loop/Decision Coverage

For this section the findAccount method of the Validator class was tested.

Test	Description
findaccounttestzeroloops	Runs the findAccount method with zero loops through the for loop.
findaccounttestsingleloop	Runs the findAccount method with one loop through the for loop.
findaccounttesttwoloops	Runs the findAccount method with two loops through the for loop.
findaccounttestmanyloops	Runs the findAccount method with many loops through the for loop.

Failure Table:

Test	
Any test involving the student flag	he writer was writing the opposite value into the file. Had to fix the if condition checks in writer's while loop when loading transactions.
All Backend class tests involving amounts	Program was not storing doubles correctly. Now truncates to two decimals as per requirements.
All Enable/Disable tests	Condition was backwards
All Enable/Disable tests	Code wrote student flag intstead of enabled flag
All create tests	Create function didn't actually push the tempAcc onto the accounts list
All Paybill tests	Paybill incorrectly used getCode instead of getMisc for the company name
All transfer	Tempdestacc pointed to the source rather than the destination
Write method	Write does not correctly pad whitespace in the output files.