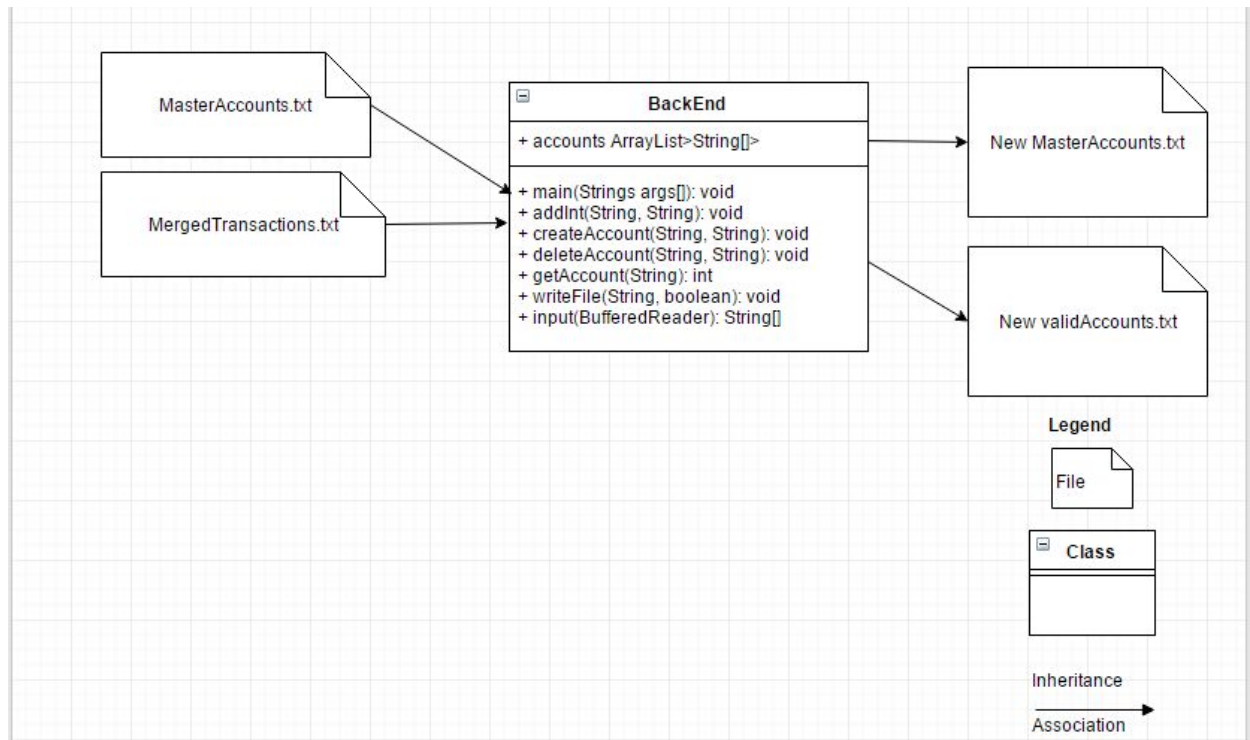


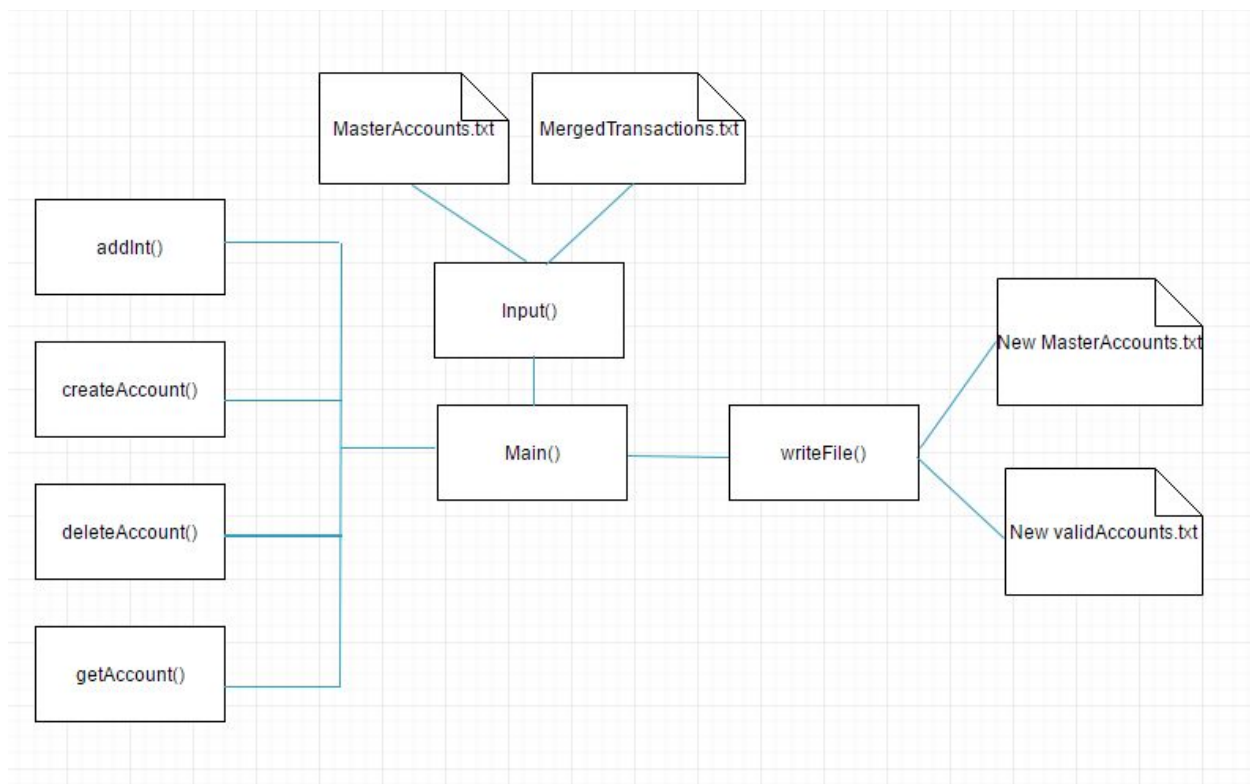
# **CISC 327 BACKEND RAPID PROTOTYPE**

**Ly Sung 10150760  
Tyler Mizuyabu 10141746  
Calvin Che 10158050**

## Backend Architecture Design



## BackEnd Methods



## BackEnd Structure

The BackEnd class will interact with the old MasterAccounts.txt and MergedTransactions.txt to produce a new MasterAccounts.txt and a new validAccounts.txt.

Main() is responsible for reading the original master accounts file and storing it in an ArrayList<String[]> accounts. It then iterates through all the transactions in the merged summary file and applies all the changes to the ArrayList. Once all transactions have been read, main calls writeFile() to write new master accounts file and new valid accounts file.

Deposit, withdraw, and transfer have been merged into one method called addInt(). This was done to get rid of duplicate code since they all perform similar arithmetic. For example, when the user deposits money, a positive amount can be passed in addInt() and negative amount for withdrawal. The same can be applied to transfer since it involves depositing and withdrawing from one account to another.

File	
Old MasterAccounts.txt	Previous day's master accounts where each account contains the account number, name and balance.
MergedTransactions.txt	Combines all the transaction summaries into one file.
New MasterAccounts.txt	Updated master accounts file after applying all valid transactions to it.
New validAccounts.txt	Updated valid accounts from the new master accounts.

## BackEnd Class Description

Attributes	Purpose
accounts	An arraylist of string[] that will be used to store the previous day master accounts.
Methods	
Main(Strings[] args)	Reads the original master accounts file and storing it in the arraylist, applies all the transactions, and writes new files.

addInt(String, Int)	Deposit, withdraw, and transfer all share this method to perform its arithmetic transactions.
createAccount(String, String)	Adds a new account to the arraylist if the account doesn't already exist.
deleteAccount(String, String)	Deletes an account from the arraylist if the account exists and the amount if exactly "000"
getAccount(String)	Iterates through the arraylist and returns the index if a given account number within the arraylist.
writeFile(String, boolean)	Takes the name of a file and what type of data to write to the file.
input(BufferedReader)	Reads a line from a given file and returns a string[] using the split(" ") call.

## New Directory Structure

### /327project

#### /frontend

FRONTEND JAVA, JAR, CLASS, EXECUTABLES, AND MANIFEST FILES

#### /backend

BACKEND JAVA, JAR, CLASS, EXECUTABLES, AND MANIFEST FILES

MasterAccounts.txt

MergedTransactions.txt

#### /testsuite

##### /frontend

ALL FILES, SCRIPTS FOR TESTING FRONTEND ( including testing accounts file)

##### /backend

ALL FILES, SCRIPTS FOR TESTING BACKEND ( including a testing master accounts file)

#### /summaries

ALL PRODUCTION TRANSACTION SUMMARY FILES

validaccounts.txt

We decided to restructure our directory system by creating a new /frontend directory to store all frontend related files. /backend is also created to store BackEnd class, MasterAccounts.txt, and MergedTransactions.txt. We also moved /testsuite from inside /frontend to be outside /327projects and /summaries to store all the transaction summary files. Additionally, validaccounts.txt is moved to /327project as well so that frontend and backend can easily access the file.