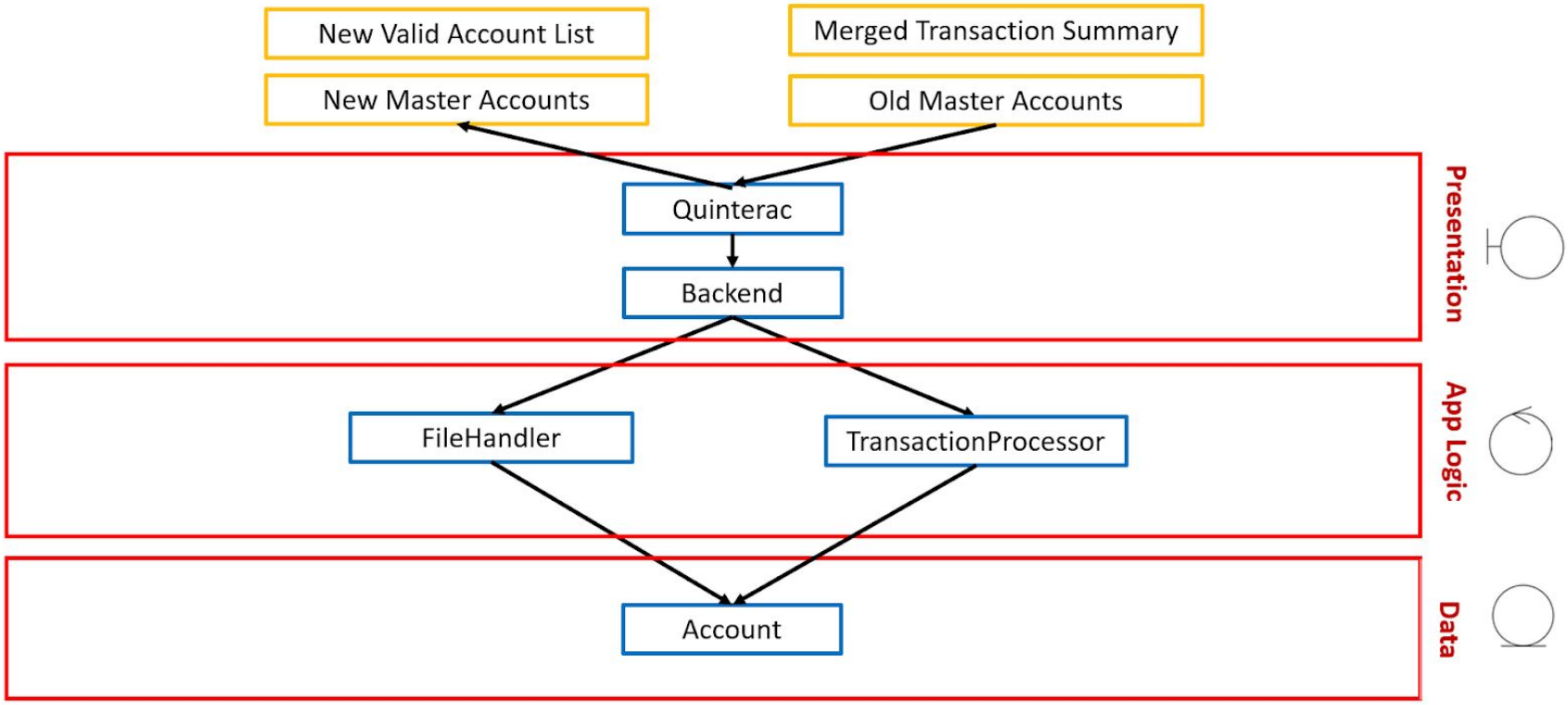
Assignment 4 - Design Document

The architecture we are adopting is the entity-control-boundary class layered architecture. An entity class stores persistent data in the data layer, a control class controls the logic of the program in the app logic layer while a boundary class interacts with the users through the presentation layer. Input and output files to the program are passed in and out of the program through the boundary class in the presentation layer. We chose to adopt a layered architecture for the back-end to match the front-end so that integration of both can be done smoothly.

The diagram below illustrates the classes that belong to the different layers. Each layer only interacts with the layer above and below it, hence reducing dependencies.



	Class	Intention	Attributes	Methods
Boundary	Quinterac	Launch Quinterac program; handle interfacing between front-end and back-end	-	main - method to start Quinterac program
	Backend	Launch backend program; reads in merged transaction summary file and old master accounts file; calls methods of TransactionProcessor and FileHandler	-	mainBackend - method to start backend program
Control	TransactionProcessor	Manipulates account data based on transactions from the merged transaction summary file	-	executeTransactions - manipulates account data based on transaction summary file create - creates new account in list of accData delete - removes old account from list of accData withdraw - withdraws money from relevant account deposit - deposits money into relevant account transfer - transfers money into relevant account crash - terminates program when error encountered
	FileHandler	Handles reading in of merged transaction summary file and old master accounts file and writing out of new masters account file and new valid accounts list	-	readOldMasterFile - reads in the old master accounts file and processes it to the right format writeNewFiles - writes out new valid account list file and new master accounts file

				readTransactionSummaryFile - reads in the transaction summary file and processes it to the right format
Data	Account	Stores account name, account number and account balance	accNum accName amount	Account - constructor for each account getAmount - return the amount of money in the account getAccName - return the name of the account getAccNum - return the number of the account setAmount - change the money in the account for the value given as a