**Quinterac Back End System Design Documentation**

CISC327 Fall 2019 Assignment 4

**Instructor**: Dr. Ding

**Contributors:**

Denny, Matt (16mmm1)

Laxdal, Kai (16kibl)

Littlefield, Brent (16bml1)

Liu, Tong (9tl14)

**1. System Specifications**

Please refer to <https://l1nna.com/327/Course%20Project.pdf> for functional specifications of the Quinterac Back End System (QBES).

Qualitative requirements are not considered to be within the scope of this assignment.

**2. System Design**

**2.1. Overview**

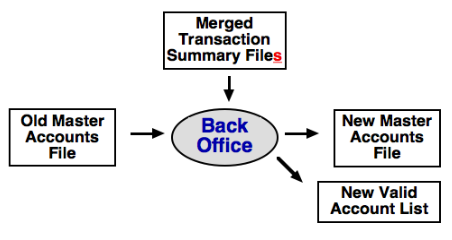
QBES takes in two inputs as command-line arguments: the directory name containing transaction summary files from the previous operational session of the front end, and the file name of the old master accounts file. The system must produce a new valid account list file as well as update the master accounts file to reflect recent transactions. This is depicted in figure 1 below.

Figure QBES inputs and outputs

QBES implements an object-oriented design which can be broken down into the following general algorithms. First, it reads each transaction summary file in the supplied directory and combines them to form a single merged transaction summary file. Using this merged transaction summary file, the backend creates a list of transactions to be processed. Next, the old master accounts file is parsed to create an account object. This account object is added to a hashmap, hashed according to the account number. A min heap of non-deleted accounts is constructed to write the new valid account list and to ensure that the new master account is sorted by account number.

Once both input files have been parsed, the system enters the processing stage of its execution. For each transaction in the transaction list, the transaction is parsed and the corresponding “process” method is invoked. For instance, if the first element of the transaction is “DEL”, then “processDEL()” will be invoked. Each process method makes the appropriate changes to the account objects and then calls AccountObj.validate() to ensure that the account attributes are valid in accordance with the backend constraints. Once all the transactions are processed, QBES overwrites the master account file using the data from each account object and creates a new valid account list based on the account numbers in the min heap.

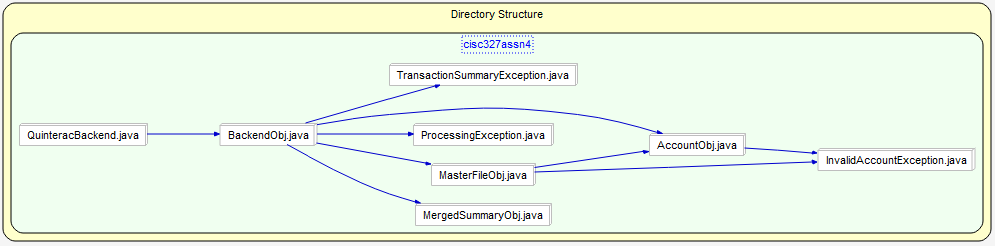
**2.2. UML Diagrams**

Figure General dependencies of QBES

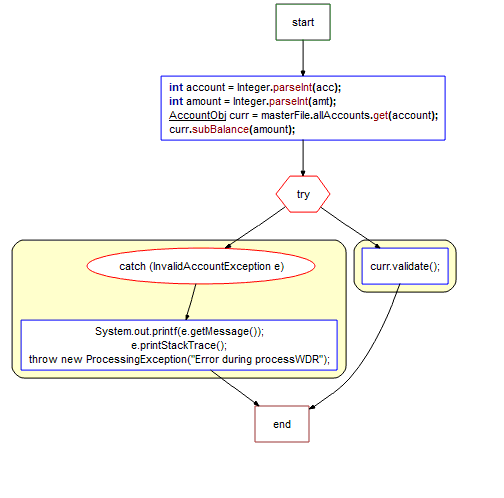


Figure Flow of execution for processWDR() method

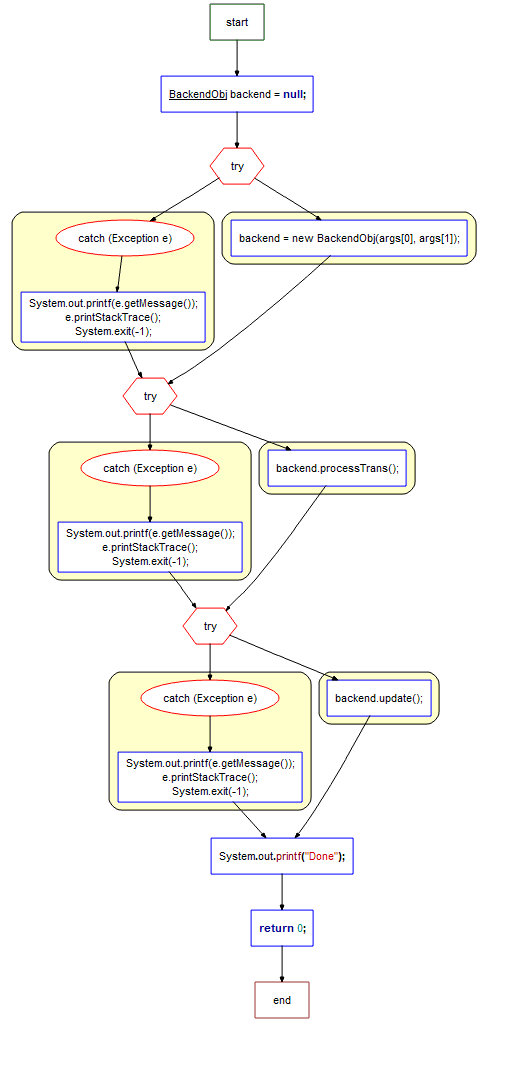


Figure Flow of execution for QuinteracBackend

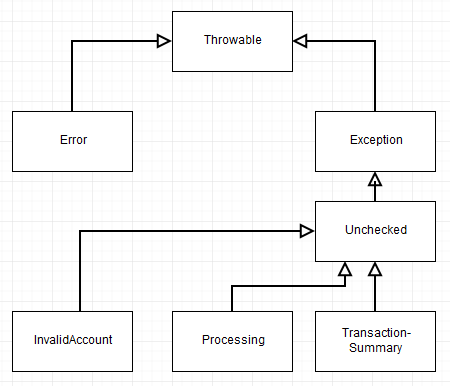


Figure Class UML for back end related exceptions

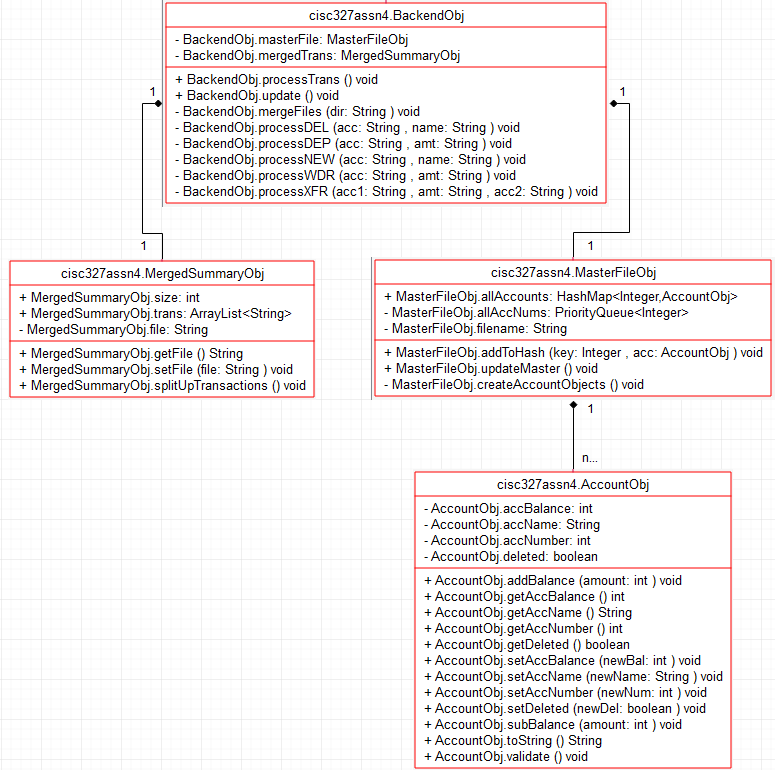


Figure Class UML for back end objects

**3. Methods Documentation**

The following tables summarize the class methods of the objects defining the behaviour of the QBES as a whole.

|  |  |
| --- | --- |
| **File: QuinteracBackend.java** |  |
| **int** main**(**String**[]** args**)** | The entry point into the system. Handles exceptions throw from the backend processing by terminating the current session.  Parameters: command-line arguments  Return: exit status |
| **File: BackendObj.java** |  |
| BackendObj **(**String dir**,** String master**)** | Backend object constructor which performs the following functions: merges transaction summaries, instantiates merged summary object, and instantiates master file object.  Parameters: transaction summary directory, old master account file  Return: new Backend object |
| **void** mergeFiles**(**String dir**)** | Iterates over each file in the transaction summary directory and writes their contents to a new merged transaction summary file (“MergedSummary.txt”).  Parameters: transaction summary directory  Return: none |
| **void** processTrans**()** | iterates over each element in the transactions list and calls a corresponding process method depending on the transaction code.  Parameters: none  Return: none |
| **void** update**()** | Invokes the updateMaster() method of the backend object.  Parameters: none  Return: none |
| **void** processDEP**(**String acc**,** String amt**)** | The process deposit method is invoked when the transaction code of the current transaction is “DEP”. This method passes ‘amt’ to the addBalance() method of the account object with the corresponding ‘acc’ number.  Parameters: account number, deposit amount  Return: none |
| **void** processWDR**(**String acc**,** String amt**)** | The process withdrawal method is invoked when the transaction code of the current transaction is “WDR”. This method passes ‘amt’ to the subBalance() method of the account object with the corresponding ‘acc’ number.  Parameters: account number, withdrawal amount  Return: none |
| **void** processXFR**(**String acc1**,** String amt**,** String acc2**)** | The process transfer method is invoked when the transaction code of the current transaction is “XFR”. This method passes ‘amt’ to the subBalance() method of ‘acc2’ and ‘amt’ to the addBalance() method of ‘acc1’.  Parameters: destination account, withdrawal amount, origin account  Return: none |
| **void** processNEW**(**String acc**,** String name**)** | The process create method is invoked when the transaction code of the current transaction is “NEW”. This method instantiates a new account object and passes it to the addToHash() method of the master file object.  Parameters: account number, account name  Return: none |
| **void** processDEL**(**String acc**,** String name**)** | The process delete method is invoked when the transaction code of the current transaction is “DEL”. This method attempts to find the account object with account number ‘acc’ and account name ‘name’ and sets the ‘deleted’ attribute to true.  Parameters: account number, account name  Return: none |
| **File: AccountObj.java** |  |
| AccountObj**(int** accNumber**,** **int** accBalance**,** String accName**)** | Account object constructor. Sets each of the attributes of the object depending on the parameter values.  Parameters: account number, account balance, account name  Return: new Account object |
| **int** getAccNumber**()** | Account number getter.  Parameters: none  Return: account number |
| **void** setAccNumber**(int** newNum**)** | Account number setter.  Parameter: new account number  Return: none |
| **int** getAccBalance**()** | Account balance getter.  Parameters: none  Return: account balance |
| **void** setAccBalance**(int** newBal**)**  **void** addBalance**(int** amount**)**  **void** subBalance**(int** amount**)** | Family of account balance setters.  Parameter: new account balance or deposit/withdrawal amount  Return: none |
| String getAccName**()** | Account name getter.  Parameters: none  Return: account name |
| **void** setAccName**(**String newName**)** | Account name setter.  Parameters: new account name  Return: none |
| **boolean** getDeleted**()** | Deleted flag getter.  Parameters: none  Return: deleted flag |
| **void** setDeleted**(boolean** newDel**)** | Deleted flag setter.  Parameters: new deleted flag value  Return: none |
| String toString**()** | toString method for account object.  Parameters: none  Return: String with account information formatted for master account file |
| **void** validate**()** | Method which checks whether the account attributes fall within the constraints defined in the system requirements.  Parameters: none  Return: none |
| **File: MasterFileObj.java** |  |
| MasterFileObj**(**String fn**)** | Master file object constructor. Creates a priority queue for the valid account numbers and a hashmap of all account objects hashed by account number.  Parameters: old master file filename  Return: new Master file object |
| **void** createAccountObjects**()** | Parses the old master accounts file and instantiates account objects corresponding to accounts present in the master file. Hashes theses account objects by account number.  Parameters: none  Return: none |
| **void** addToHash**(**Integer key**,** AccountObj acc**)** | Attempts to add an account object to the hashmap. Throws an exception if the key already exists in the hash.  Parameters: account number, account object  Return: none |
| **void** updateMaster**()** | Iteratively polls the account number priority queue. For each account number, writes toString to the new master account file, and writes the account number to the new valid account file if the account is not deleted.  Parameter: none  Return: none |
| **File: MergedSumamryObj.java** |  |
| MergedSummaryObj **(**String file**)** | Merged summary object constructor. Creates an arraylist for all transactions to be processed by the backend object.  Parameters: merged summary file name  Return: new Merged summary object |
| **void** splitUpTransactions**()** | Parses each line in the merged summary file by separating it into a String array delimited by spaces. Adds each String array into the transactions arraylist  Parameters: none  Return: none |
| String getFile**()** | Merged summary file getter.  Parameters: none.  Return: merged summary file name |
| **void** setFile**(**String file**)** | Merged summary file setter.  Parameters: new merged summary file name  Return: none |