

# QUINTERAC

Loaft Inc.

December 7, 2019

Samir Mekhdi – 10191295

Luvit Chumber – 10190467

Jessica Wong – 10181646

Tayyab Ahmad – 10197212

## 1) Source Listing:

CISC327-LoaftInc

|--QUINTERAC/Testing/

| |--integration/integrationTest.java

| |--backend/backendTest.java

| |--frontend/frontendTest.java

|--QUINTERAC/src/backend/

| |--backend.java

| |--Account.java

|--QUINTERAC/src/frontend/

| |--frontend/frontend.java

| |--frontend/Terminal.java

|--QUINTERAC/src/transactionClasses/

| |--CreateAcct.java

| |--DeleteAcct.java

| |--Deposit.java

| |--Login.java

| |--Logout.java

| |--Transaction.java

| |--Transfer.java

| |--Withdraw.java

## Daily scripts:

Both daily and weekly scripts were implemented via `integrationTest.java`, which can be found in the testing directory. 5 days of transactions were created, with each set running 3 separate transaction sessions (frontends). The loop runs 5 of these daily scripts in succession. The daily commands are named as `t_in_day1`, `t_in_day2`, etc. This structure contains a list of strings to be passed in as terminal input. The contents of each daily test are as follows:

- `t_in_day1`:
  - Create accounts
- `t_in_day2`:
  - Deposit into these accounts
- `t_in_day3`:
  - Transfer between these accounts
- `t_in_day4`:
  - Withdraw from these accounts
- `t_in_day5`:
  - Delete accounts
  - Withdraw accounts
  - Transfer

```
String t_in_day4[][] = {
    { "login agent", //session 1
      "withdraw 9999999 8180000",
      "logout"},
    { "login agent", //session 2
      "withdraw 1000001 101100",
      "withdraw 5000001 400000",
      "withdraw 1000003 61820",
      "withdraw 1000002 11500",
      "withdraw 1000000 98000",
      "withdraw 7777777 7900000",
      "logout"},
    { "login agent", //session 3
      "withdraw 3030303 1064000",
      "withdraw 8300781 107000",
      "withdraw 1000000 150000",
      "logout"} }; //END DAY 4

String t_in_day5[][] = {
    { "login agent", //session 1
      "deleteacct 5000001 JaneDee69",
      "transfer 9999999 1000000 100",
      "transfer 9999999 1000001 100",
      "withdraw 9999999 1349600",
      "logout"},
    { "login agent", //session 2
      "deleteacct 7777777 BethanyWhite212",
      "transfer 9999999 1000003 100",
      "deleteacct 3030303 Walter101Grey",
      "transfer 9999999 1000002 100",
      "logout"},
    { "login agent", //session 3
      "deleteacct 8300781 747JonathanBlue",
      "deleteacct 9999999 JohnDoe420",
      "logout"} }; //END DAY 5

String t_in[][][] = {t_in_day1,t_in_day2,t_in_day3,t_in_day4,t_in_day5};
```

Figure 1. More In-depth Sample of Commands and Structure

Note: Full Source Listing in Appendix A

## 2) Daily script inputs:

### Day 2:

> Transactions being issued:

>> FrontEnd #1

login agent

deposit 1000000 50000

deposit 1000001 100

deposit 9999999 200000

logout

>> FrontEnd #2

login agent

deposit 9999999 9500000

deposit 5000001 320

deposit 8300781 100000

deposit 3030303 671000

deposit 1000002 500

logout

>> FrontEnd #3

login agent

deposit 7777777 8000000

deposit 3030303 500000

deposit 1000003 1500

deposit 5000001 400000

logout

### 3) Corresponding Merged TSF:

Day 2:

DEP 1000000 50000 0000000 \*\*\*

DEP 1000001 100 0000000 \*\*\*

DEP 9999999 200000 0000000 \*\*\*

EOS

DEP 9999999 9500000 0000000 \*\*\*

DEP 5000001 320 0000000 \*\*\*

DEP 8300781 100000 0000000 \*\*\*

DEP 3030303 671000 0000000 \*\*\*

DEP 1000002 500 0000000 \*\*\*

EOS

DEP 7777777 8000000 0000000 \*\*\*

DEP 3030303 500000 0000000 \*\*\*

DEP 1000003 1500 0000000 \*\*\*

DEP 5000001 400000 0000000 \*\*\*

EOS

EOS

#### 4) Master Accounts File:

##### Day 0:

// empty file

##### Day 1:

9999999 0 JohnDoe420  
8300781 0 747JonathanBlue  
7777777 0 BethanyWhite212  
5000001 0 JaneDee69  
3030303 0 Walter101Grey  
1000003 0 JessicaW  
1000002 0 SamirM  
1000001 0 LuvitC  
1000000 0 TayyabA

##### Day 2:

9999999 9700000 JohnDoe420  
8300781 100000 747JonathanBlue  
7777777 8000000 BethanyWhite212  
5000001 400320 JaneDee69  
3030303 1171000 Walter101Grey  
1000003 1500 JessicaW  
1000002 500 SamirM  
1000001 100 LuvitC  
1000000 50000 TayyabA

##### Day 3:

9999999 9530000 JohnDoe420  
8300781 107000 747JonathanBlue  
7777777 7900000 BethanyWhite212  
5000001 400000 JaneDee69  
3030303 1064000 Walter101Grey  
1000003 61820 JessicaW  
1000002 11500 SamirM  
1000001 101100 LuvitC  
1000000 248000 TayyabA

##### Day 4:

9999999 1350000 JohnDoe420  
8300781 0 747JonathanBlue  
7777777 0 BethanyWhite212  
5000001 0 JaneDee69  
3030303 0 Walter101Grey  
1000003 0 JessicaW  
1000002 0 SamirM  
1000001 0 LuvitC  
1000000 0 TayyabA

##### Day 5:

1000003 100 JessicaW  
1000002 100 SamirM  
1000001 100 LuvitC  
1000000 100 TayyabA

## 5) Integration Defect Report:

The following errors were found during testing, with an explanation of how it was fixed and updated.

Detected Defects	Solution
<p>Lowercase convert: Converted input to lowercase in frontend before splitting for transaction code but affected name as well.</p> <p>Before:</p> <pre>System.out.println("Enter next transaction: "); inputRow = in.nextLine().toLowerCase(); input = inputRow.split(" ");</pre>	<p>Updated the convert case to be done after splitting the name and the trans code, essentially changing the order of operations.</p> <p>After:</p> <pre>System.out.println("Enter next transaction: "); inputRow = in.nextLine(); input = inputRow.split(" "); input[0] = input[0].toLowerCase();</pre>
<p>String split: inputs were not being split correctly while validating master accounts containing a new account. In the case where a name had a number in it, the string would split at these points if the number was the same as the amount in the account (0)</p> <p>Before:</p> <pre>String[] in = inRaw.split(" "); // cannot use this method for names, names can have spaces String type = in[0]; String acctTo = in[1]; String amountStr = in[2]; String acctFrom = in[3]; int startIDX = inRaw.lastIndexOf(acctFrom); String name = inRaw.substring(startIDX+acctFrom.length()); name = name.trim(); // removes space from front of string;</pre>	<p>Since the format of the input was known, we accommodated this by looking for a more specific match, in this case a " " (space) before and after the amount field</p> <p>After:</p> <pre>String[] in = inRaw.split(" "); // cannot use this method for names, names can have spaces String type = in[0]; String acctTo = in[1]; String amountStr = in[2]; String acctFrom = in[3]; int startIDX = inRaw.lastIndexOf(" " + acctFrom + " ") + 1; String name = inRaw.substring(startIDX+acctFrom.length()); name = name.trim(); // removes space from front of string;</pre>
<p>Account Balance cannot be 0: Account information was flagged as incorrect during the validation step for master accounts due to the account balance being 0 (which all new accounts have). Initially the code checked for numbers that are at least 3 digits, no greater than 99999999, and not a negative number</p> <p>Before: if ( ( in.length() &lt; 3 )    num &gt; 99999999    num &lt; 0 )</p>	<p>Checks that the account balance is under the same conditions as previously written except for a value of 0</p> <p>After: if ( ( in.length() &lt; 3 &amp;&amp; num != 0 )    num &gt; 99999999    num &lt; 0 )</p>
<p>Withdraw amount from account: User is not allowed to withdraw the exact balance of the account</p> <p>Before: if (amount &lt; currentAmount)</p>	<p>Allow the user to withdraw the amount in the account or less</p> <p>After: if (amount &lt;= currentAmount)</p>

## Appendix A: Source Listing

```
/*
 * Quinterac Integration Testing
 * Loaft Inc.
 * Author: Luvit Chumber
 * Date: December 6, 2019
 * */

package integration;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.PrintStream;
import java.nio.file.Files;
import java.util.Scanner;
import frontend.frontend;
import backend.backend;

public class integrationTest {

    private static String runFrontEndDaily(String[][] args,String [][] transactions) throws IOException {
        String tsf_merged = "";

        for (int i = 0; i < args.length; i++) {
            String userInput = String.join(System.lineSeparator(), transactions[i]);
            ByteArrayInputStream in = new ByteArrayInputStream(userInput.getBytes());
            System.setIn(in);
        }
    }
}
```



```

        frontend.main(args[i]);

        Scanner sc = new Scanner(new File(args[i][1]));

        while(sc.hasNext()) {

            tsf_merged += sc.nextLine() + System.lineSeparator();

        }

        sc.close();

    }

    tsf_merged += "EOS" + System.lineSeparator(); // as per requirements merged tsf contains
    empty tsf file at end

    return tsf_merged;

}

```

```

public static void main(String[] args) throws IOException {

```

```

    // setup stdin & stdout:

    PrintStream console = System.out;

    PrintStream outContent = new PrintStream(new ByteArrayOutputStream());

    PrintStream errContent = new PrintStream(new ByteArrayOutputStream());

    System.setOut(outContent);

    System.setErr(errContent);

    // reset files to clean slate

    File mAccts = new File(System.getProperty("user.dir")+"\\files\\masteraccts.txt");

    File vAccts = new File(System.getProperty("user.dir")+"\\files\\validaccts.txt");

    Files.write(mAccts.toPath(),"".getBytes());

    Files.write(vAccts.toPath(),"0000000".getBytes());

    String t_in_day1[][] = {

        { "login agent",      //session 1

```

```

        "createacct 1000000 TayyabA",
        "createacct 1000001 LuvitC",
        "createacct 1000002 SamirM",
        "createacct 1000003 JessicaW",
        "logout"},
    { "login agent",          //session 2
        "createacct 9999999 JohnDoe420",
        "createacct 5000001 JaneDee69",
        "logout"},
    { "login agent",          //session 3
        "createacct 7777777 BethanyWhite212",
        "createacct 3030303 Walter101Grey",
        "createacct 8300781 747JonathanBlue",
        "logout"} };          //END DAY 1

```

```

String t_in_day2[][] = {
    { "login agent",          //session 1
        "deposit 1000000 50000",
        "deposit 1000001 100",
        "deposit 9999999 200000",
        "logout"},
    { "login agent",          //session 2
        "deposit 9999999 9500000",
        "deposit 5000001 320",
        "deposit 8300781 100000",
        "deposit 3030303 671000",
        "deposit 1000002 500",
        "logout"},
    { "login agent",          //session 3

```

```
"deposit 7777777 8000000",
"deposit 3030303 500000",
"deposit 1000003 1500",
"deposit 5000001 400000",
"logout"} };                                //END DAY 2
```

```
String t_in_day3[][] = {
    { "login atm",      //session 1
      "transfer 1000000 1000001 1000",
      "transfer 1000000 1000002 1000",
      "logout"},
    { "login agent",    //session 2
      "transfer 7777777 1000000 50000",
      "transfer 3030303 8300781 7000",
      "transfer 3030303 9999999 100000",
      "transfer 5000001 1000003 320",
      "logout"},
    { "login agent",    //session 3
      "transfer 7777777 1000003 50000",
      "transfer 9999999 1000000 150000",
      "transfer 9999999 1000001 100000",
      "transfer 9999999 1000002 10000",
      "transfer 9999999 1000003 10000",
      "logout"} };                            //END DAY 3
```

```
String t_in_day4[][] = {
    { "login agent",    //session 1
      "withdraw 9999999 8180000",
      "logout"},
```

```

{ "login agent",          //session 2
  "withdraw 1000001 101100",
  "withdraw 5000001 400000",
  "withdraw 1000003 61820",
  "withdraw 1000002 11500",
  "withdraw 1000000 98000",
  "withdraw 7777777 7900000",
  "logout"},
{ "login agent",          //session 3
  "withdraw 3030303 1064000",
  "withdraw 8300781 107000",
  "withdraw 1000000 150000",
  "logout"} };           //END DAY 4

```

```

String t_in_day5[][] = {
  { "login agent",        //session 1
    "deleteacct 5000001 JaneDee69",
    "transfer 9999999 1000000 100",
    "transfer 9999999 1000001 100",
    "withdraw 9999999 1349600",
    "logout"},
  { "login agent",        //session 2
    "deleteacct 7777777 BethanyWhite212",
    "transfer 9999999 1000003 100",
    "deleteacct 3030303 Walter101Grey",
    "transfer 9999999 1000002 100",
    "logout"},
  { "login agent",        //session 3
    "deleteacct 8300781 747JonathanBlue",

```

```
"deleteacct 9999999 JohnDoe420",  
"logout"} };
```

```
//END DAY 5
```

```
String t_in[][][] = {t_in_day1,t_in_day2,t_in_day3,t_in_day4,t_in_day5};
```

```
// declare all filenames
```

```
String mTsfFN[] =  
System.getProperty("user.dir")+"\\files\\mergedtsfday1.txt",
```

```
System.getProperty("user.dir")+"\\files\\mergedtsfday2.txt",
```

```
System.getProperty("user.dir")+"\\files\\mergedtsfday3.txt",
```

```
System.getProperty("user.dir")+"\\files\\mergedtsfday4.txt",
```

```
System.getProperty("user.dir")+"\\files\\mergedtsfday5.txt"};
```

```
String tsfFN[][] =  
System.getProperty("user.dir")+"\\files\\tsfday1_instance1.txt",
```

```
System.getProperty("user.dir")+"\\files\\tsfday1_instance2.txt",
```

```
System.getProperty("user.dir")+"\\files\\tsfday1_instance3.txt"},
```

```
{  
System.getProperty("user.dir")+"\\files\\tsfday2_instance1.txt",
```

```
System.getProperty("user.dir")+"\\files\\tsfday2_instance2.txt",
```

```
System.getProperty("user.dir")+"\\files\\tsfday2_instance3.txt"},
```

```
{  
System.getProperty("user.dir")+"\\files\\tsfday3_instance1.txt",
```

```
System.getProperty("user.dir")+"\\files\\tsfday3_instance2.txt",
```

```

        System.getProperty("user.dir")+"\\files\\tsfdays\\instance3.txt"},
        {
        System.getProperty("user.dir")+"\\files\\tsfdays\\instance1.txt",

        System.getProperty("user.dir")+"\\files\\tsfdays\\instance2.txt",

        System.getProperty("user.dir")+"\\files\\tsfdays\\instance3.txt"},
        {
        System.getProperty("user.dir")+"\\files\\tsfdays\\instance1.txt",

        System.getProperty("user.dir")+"\\files\\tsfdays\\instance2.txt",

        System.getProperty("user.dir")+"\\files\\tsfdays\\instance3.txt"}};

String vAcctsFN = System.getProperty("user.dir")+"\\files\\validaccts.txt";
String mAcctsFN = System.getProperty("user.dir")+"\\files\\masteraccts.txt";

for (int i = 0; i < 5 ; i++) {
    // create temporary files for tsf transaction

    File transaction_summary_file1 = new File(tsfFN[i][0]);
    File transaction_summary_file2 = new File(tsfFN[i][1]);
    File transaction_summary_file3 = new File(tsfFN[i][2]);

    // setup parameters for the program to run

    String[][] args_to_frontend = { { vAcctsFN,
transaction_summary_file1.getAbsolutePath() },

                                     { vAcctsFN,
transaction_summary_file2.getAbsolutePath() },

                                     { vAcctsFN,
transaction_summary_file3.getAbsolutePath() } };

```

```

// run frontend (3 instances)

String tsf_merged = runFrontEndDaily(args_to_frontend, t_in[i]);

//prepare merged tsf
File mergedtsf = new File(mTsfFN[i]);
Files.write(mergedtsf.toPath(), String.join("\n", tsf_merged).getBytes());

// run backend

String [] args_to_backend = { mAcctsFN, mergedtsf.getAbsolutePath()};

backend.main(args_to_backend);

System.setOut(console);

// Print Debug Report Each Day
mAccts = new File(mAcctsFN);

System.out.println("Day " + (i+1));
System.out.println("> Transactions being issued");

for(int j = 0; j<t_in[i].length;j++) {
    System.out.println(">> FrontEnd #" + (j+1));
    for(int k = 0; k<t_in[i][j].length; k++) {
        System.out.println(t_in[i][j][k]);
    }
}

System.out.println("\n> Merged TSF:");

```

```
System.out.println(tsf_merged);  
System.out.println("\n> Updated Master Accounts:");
```

```
Scanner sc = new Scanner(mAccts);  
while (sc.hasNext()) {  
    System.out.println(sc.nextLine());  
}  
sc.close();
```

```
System.out.println("\n");  
System.setOut(outContent);
```

```
}
```

```
}
```

```
}
```