Group 50 Assignment 6 Submission Documentation

Sunday Dec 8th 2019

Group Members:
Matt Denny
Kai Laxdal
Brent Littlefield
Tong Liu

The Purpose of this documentation is to give understanding to how we implemented two scripts that will run our Quinterac frontend and backend in both a daily format as well as a weekly format.

Source Listings

In this section we have included source listings of both our Daily script as well as Weekly script

Daily Script

In order to run our script you must pass a single integer argument when running the script in order to select which transaction set you would like to run. We have 5 transaction sets that are held within our script. The transaction sets are found within the switch statement that is in the getInputCases method. The daily script must have a single integer argument passed to it in order to run. It will take any integer as input but will only produce correct output for the integers 0 to 4 as there is a default statement which handles any cases where the argument passed to the script that is not in that range.

```
package main.java.ca.queensu.cisc327;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.PrintStream;
import main.java.ca.queensu.cisc327.Quinterac;
import main.java.ca.queensu.cisc327.QuinteracBackend;
import java.util.*;
public class DailyScript {
	public static void main(String args[]) {
		/Front end args are "valid accounts file" and "transaction file"
	String[] frontEndArgs =
{"C:\\Users\\Denny\\eclipse-workspace\\Quinterac\\src\\main\\java\\ca\\queensu\\cisc327\\ValidAccountsFile.txt"
				, "C:\\Users\\Denny\\eclipse-workspace\\Quinterac\\src\\main\\java\\ca\\queensu\\cisc327\\"};
	//Back end args are "transaction summary file directory" and "master accounts file"
	String[] backEndArgs = {"C:\\Users\\Denny\\eclipse-workspace\\Quinterac\\src\\main\\java\\ca\\queensu\\cisc327",
				"C:\\Users\\Denny\\eclipse-workspace\\Quinterac\\src\\main\\java\\ca\\queensu\\cisc327\\"};
	List<String> terminal_input = getInputCases(args);
	ByteArrayOutputStream outContent = new ByteArrayOutputStream();
	PrintStream defaultOut = System.out;
	//setup user input
	String userInput = String.join(System.lineSeparator(), terminal_input);
	ByteArrayInputStream in = new ByteArrayInputStream(userInput.getBytes());
```

```
		System.setIn(in);
		// System.setOut(new PrintStream(outContent));

		Quinterac.main(frontEndArgs);
		QuinteracBackend.main(backEndArgs);
		//System.setOut(defaultOut);
		if(args.length != 0) {
			System.out.println("Daily run " + args[0] + " complete...");
			return;
		}
		System.out.println("Daily run complete...");
}
private static List<String> getInputCases(String[] iCase){
		List<String> out = Arrays.asList("--quit");
		if(iCase.length == 0) {
			return out;
		}
		switch(iCase[0]) {
			case "0":
				out = Arrays.asList("login", "agent", "createacc", "1111111", "Bob", "logout",
						"login", "agent", "createacc", "2222222", "Rob", "logout",
						"login", "agent", "createacc", "3333333", "Cob", "logout", "--quit");
				break;
			case "1":
				out = Arrays.asList("login", "machine", "deposit", "1111111", "2000", "logout",
						"login", "machine", "deposit", "2222222", "5000", "logout",
						"login", "machine", "transfer", "3333333", "2222222", "500", "logout", "--quit");
				break;
			case "2":
				out = Arrays.asList("login", "machine", "withdraw", "3333333", "500", "logout",
						"login", "machine", "deposit", "1111111", "1000", "logout",
						"login", "machine", "deposit", "2222222", "500", "logout","--quit");
				break;
			case "3":
				out = Arrays.asList("login", "machine", "transfer", "2222222", "1111111", "1000", "logout",
						"login", "machine", "deposit", "2222222", "500", "logout",
						"login", "agent", "deleteacc", "3333333", "Cob", "logout", "--quit");
				break;
			case "4":
				out = Arrays.asList("login", "agent", "deposit", "1111111", "500000", "logout",
						"login", "machine", "transfer", "2222222", "1111111", "5000", "logout",
						"login", "machine", "withdraw", "2222222", "1000", "logout", "--quit");
				break;
		}
		return out;
}// end of getInputCases
}// end of daily script
```

Weekly Script

Our Weekly script for Quinterac simply runs the daily script 5 times. The transaction set run each time within daily script will be changed for each of the 5 days that the weekly script is created to imitate. The way the Weekly script changes the transaction set it will invoke is due to the argument that is passed to the daily script. In

our weekly script we simply iterate through the integers 0 to 4 in order to hit the 5 preset transaction sets we created in Daily script.

```
package main.java.ca.queensu.cisc327;
import main.java.ca.queensu.cisc327.DailyScript;
public class WeeklyScript {
	public static void main(String args[]) {
		String[] dailyArgs;
		for (int i = 0; i < 5; i++) {
			dailyArgs = new String[]{"" + i};
			DailyScript.main(dailyArgs);
		}
		System.out.println("Weekly run complete...");
	}
}
```

Printout of transaction session inputs for one run of your Daily script

For the first daily run we decided to have 3 create account transactions that will take place 1111111 for Bob, 2222222 for Rob, and 3333333 for Cob

```
out = Arrays.asList("login", "agent", "createacc", "1111111", "Bob", "logout",
        "login", "agent", "createacc", "2222222", "Rob", "logout",
        "login", "agent", "createacc", "3333333", "Cob", "logout", "--quit");
```

Printout of the Merged Transaction Summary file from Daily run above

NEW 1111111 000 0000000 Bob
EOS 0000000 000 0000000 ***
NEW 2222222 000 0000000 Rob
EOS 0000000 000 0000000 ***
NEW 3333333 000 0000000 Cob
EOS 0000000 000 0000000 ***

A printout of the Master Accounts File after each of the five Daily runs made by your Weekly script.

Our script starts with an empty Master Accounts file so there are no accounts held in that file prior to the running of the first daily script. A more Descriptive outline of each of the days that are outlined in the chart below can be found in the root directory of our repository inside the folder named *“Assignment 6 Daily Script Printouts”.*

After Day 1 3 Create Accounts	1111111 0 Bob 2222222 0 Rob 3333333 0 Cob
After Day 2 2 Deposits and 1 transfer	1111111 2000 Bob 2222222 4500 Rob 3333333 500 Cob
After Day 3 2 Deposits and 1 Withdrawal	1111111 3000 Bob 2222222 5000 Rob 3333333 0 Cob
After Day 4 1 Transfer, 1 Deposit, and 1 DeleteAcct	1111111 2000 Bob 2222222 6500 Rob
After Day 5 1 Deposit, 1 Transfer, and 1 Withdrawal	1111111 497000 Bob 2222222 10500 Rob

Integration Defect Report

When implementing our scripts for assignment 6 we did not come across anything significant in our code that was problematic as everything was working properly when we were creating the two scripts. The only thing we changed was that the second argument passed to the Backend of Quinterac (QuinteracBackend.java) will be the directory that holds the master accounts file rather than the file path of the master accounts file. We opted to change this argument to the directory of the file as this directory was not only used for the MasterAccountsFile but also for the ValidAccountsFile.