ELEC 327 – Assignment 6

The Xtreme Team

Dennis Grajo, Hojun Lee, Mike McColm, Stephen Obadinma

## 1. Source Listing For Daily/Weekly Script

The code for these scripts are as follows:

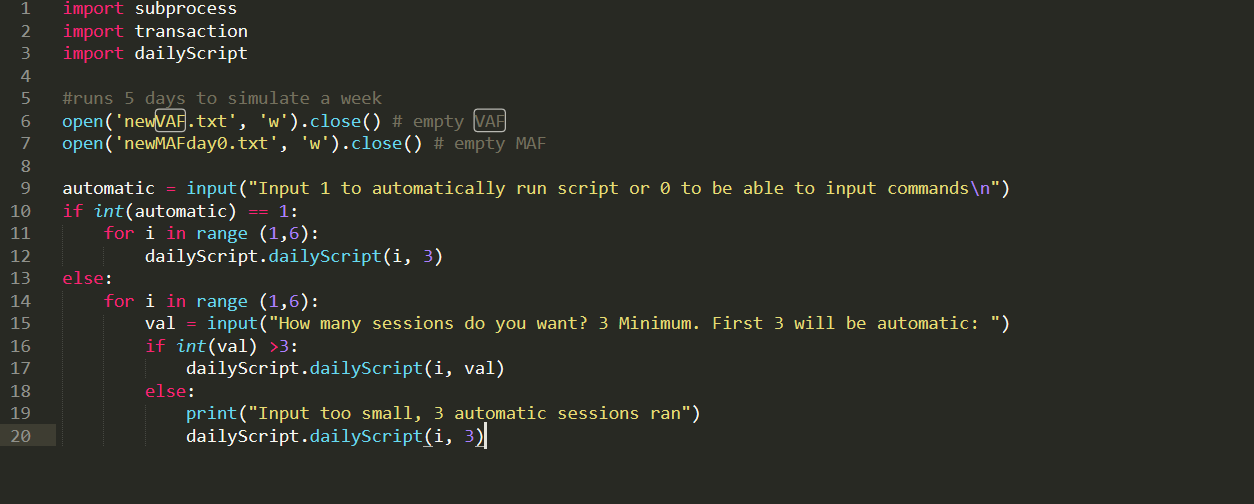
\*Note: the functions call transaction.py, which imports a text file of transaction information. This file can be found in the team’s GitHub. They call inputs from folders 1a, 1b, etc., which contains sample transactions, which can also be found on the GitHub\*

Daily Script:



To run a single iteration of the daily script, enter python dailyScript.py in the command line. It will run the results that can be seen below. When it is finished the results can be seen in the files TTSF1a, TTSF1b, TTSF1c, newVAF, newMAF, and mergedTSF. The daily script also has a function that is called by the weekly script. It runs a given set of inputs and allows a user to enter inputs too and based on the weekly script it can run for any number of sessions above 3. The daily script does all of the required functions. The merging is done in the helper function to the daily script (transaction).

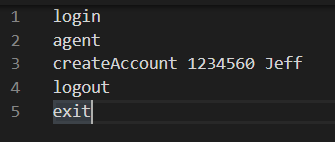
Weekly Script:

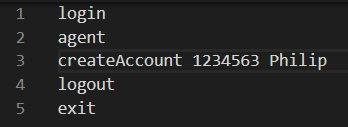


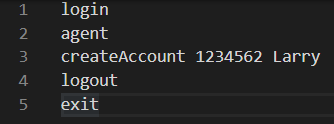
Calls the daily script 5 times for separate inputs as expected.

## 2. Daily Script Inputs

With an empty VAF and MAF, the inputs for the first day are as follows:

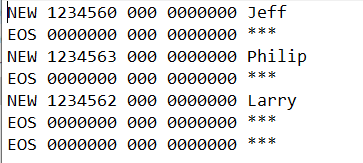






## 3. Merged Transaction Summary File

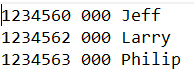
After the above 3 input scripts were run, the following merged transaction summary file was observed:



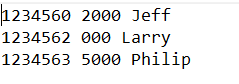
## 4. Master Accounts File

The following master accounts file was observed after the weekly run:

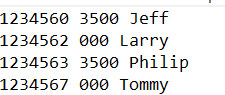
Day 1



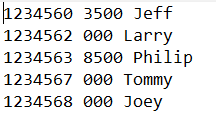
Day 2



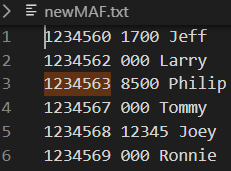
Day 3



Day 4



Day 5



To get these results type 1 when the weekly script prompts you to choose to run the code automatically.

## 5. Integration Default Report

|  |  |
| --- | --- |
| Problem | Solution |
| Couldn’t run more than 3 sessions, and front end did not produce separate files | Made the front-end produce files based on the day and the session number instead of outputting a generic TSF file. |
| Backend did not produce intermediate MAFs | Had to modify code so that when the backend finishes running it saves the new MAF in a new file, and at the start it reads the one from the previous day instead of only using one file that is constantly modified |
| Sending commands using the script (like to login) to the terminal | Running commands from pre-set input command files that are read by the front end |
| No interactive component | The weekly script was changed so that a user can input their own commands to see |
| Not clearing the VAF or MAF at the start of a weekly session | Added code to clear the VAF and MAF at the start of a weekly session |
| Each front-end session modifies the VAF affecting subsequent runs of the | Changed the front end so it runs based on a temporary VAF that is a clone of the original VAF. This temp file is copied anew for every instance of the frontend so the original is not affected. |
| Backend still writes deleted accounts to VAF (not for the next day, but it reappears in 2 days) | No solution found while sticking to requirements. No way to tell if account is deleted just based on MAF, and requirements do not allow backend to take in another parameter of the old VAF or a global list of deleted accounts |
| Backend didn’t delete accounts even when the name was correct | Stripped the whitespace from the names when comparing them |
| Backend didn’t write 0000000 at end of the new VAF | Added a line to write this |
| No ability to set the number of sessions | Added a user input so that they can choose how many sessions to run in a day. |
| Can create an account on an existing account number | Changed a line so you cannot replace any accounts in the MAF (even if they are deleted) by checking if an account number exists before creating a new one |