

Defect Document

User Guide

Frontend:

1. Ensure repository is cloned
2. Navigate to the *src* folder in the repository
 - a. `cd root/frontend2/src` (assuming you are in the *Tree* directory after cloning)
3. Execute the following command to compile the frontend
 - a. `g++ -o distribution-system main.cpp`
4. Once compiled, enter the following to run the program
 - a. `./distribution-system ../data/currentaccounts.txt ../data/availablegames.txt ../data/gamescollection.txt ../data/dailytransactions.txt`
 - b. *You may get an error upon running that notifies how some of the data in the files are not the correct length and thus are skipped. This is due to the line ending conversions that occur when uploading to Github. If you encounter these messages, exit the program and enter the following command:
 - i. `dos2unix <name of file>`
 - ii. Ex: `dos2unix ../data/currentaccounts.txt` if current accounts file was giving errors
5. Once the program is run, a session is created. Feel free to login and test commands (there is an *admin* user you can login to gain access to the full system). Once you logout from a user, a daily transaction file is created for the session and you can either continue to login as another user, or exit the program using *exit*

Backend:

***Note: The backend is meant to run off of a merged daily transaction file. If you would like to run the backend through the script rather than manually, please read the next section on scripts. This is for running the backend manually.**

1. Ensure repository is cloned
2. Open a Python terminal.
3. Navigate to the `project` directory within the repository (`cd root/project`).
4. Locate the *main.py* file
5. Run the *main.py* file
 - a. Execute the file to process all transactions listed in *dailytransactions.txt*, updating relevant files such as *availablegames.txt* and *currentaccounts.txt*.

Scripts:

***Note: The following are instructions for running the daily & weekly scripts. For instructions on how to run test suite scripts, see the frontend readme document and the repo readme to see test suite running instructions (for frontend and backend)**

1. Ensure repository is cloned
2. Navigate to the *scripts* folder
 - a. `cd root/frontend2/testing/scripts`
3. Enter the following commands to give permissions to run the daily and weekly script
 - a. `chmod 777 daily.sh`
 - b. `chmod 777 weekly.sh`
4. Enter the following to execute the daily script, simulating 3 sessions of the front end during the day
 - a. `./daily.sh inputs actual_output_daily expected_output_daily`
`../../data/currentaccounts.txt ../../data/gamescollection.txt`
`../../data/availablegames.txt`
5. Enter the following to execute the weekly script, simulating the daily script run over the course of a week
 - a. `./daily.sh "weekly/${day}/inputs" weekly/actual_output_weekly`
`weekly/expected_output_weekly ../../data/currentaccounts.txt`
`../../data/gamescollection.txt ../../data/availablegames.txt`

Defect	Description	Solution
Syntax Error when Accessing Create Entry	Attempted to give Create Entry two Strings and ended with a script error	Had to check source code and verify that Create Entry took a String and an Integer in that order that had to be submitted
Error For Logout Script	Logout was setting the state back to not logged in and would not allow any states to occur afterward	The main function had to be modified to allow the logout to exit and not get into code that would get stuck forever
Unable to Run Scripts	Had to search through multiple directories to get the Front End to run	Had to find the proper directory and use the proper syntax file path to run the front end code
Unable to run front end on Mac Devices	Mac would give an error saying architecture error unable to run	People on Mac had to switch to windows devices and then download either the windows subsystem for linux or boot off of linux to get the front end to work
Daily script does not create daily transaction files for each	When the daily script is run, the script does not create the	Run the frontend manually to create the daily transaction

session	daily transaction files for each session run, despite the transactions from the input files running correctly	files for each session
Script would run forever	The script would run but would never end	Had to go through each script to find which one was breaking the compilation and never ending. After this was isolated code had to be modified to allow the whole program to run
Front End 3 Not Compatible with project	Unable to get get Front End 3 to work at all regardless of what was done (errors in makefile making it unable to compile)	Had to use Front End 2 and modify it where needed to make the program work
Compiling code and running scripts needed to be performed in different directories	Compiling the code had to be done in one Directory and running the code needed to be done in another.	Had to navigate to the first directory to compile the code, then backtrack and enter the directory with the scripts to run them
Front end Manually works but logout script broken	The logout function works perfectly when manually going into the program, but it does not work in a script and gets stuck.	Figured out it was not the logout script itself that was breaking but when writing to the daily transaction file.

Assumptions List

- Based on the way that the front end is working, assuming that the user knows how the program should function. This is because the way that the console works is the first message that pops up is enter transaction code, and then it expects you to log in before you can do anything else
- Need to assume the layout of the front end is correct, even though the program asks for a transaction code before asking you to login
- Need to assume that all the transaction codes would work specifically in the .h files
- Assuming that after logout the program would not end (original frontend creator assumption)
- Did not delete daily transaction files and stored them in a folder, as client did not specify its deletion or saving them
- Assumed daily transaction file is uploaded at the logout phase when the user logs out and not when the program exits

- Assumption that only 1 game can be put up for sale per session (original frontend creator assumption)