

## CSCI 3060U - Integration Defect Report (Phase 6)

Defect No.	Testing Objective	Nature of the failure	Actions Taken	Pass / Fail
D1	Retrofit front end to support multiple and randomly generated daily transaction files	Only one fixed transaction file	Changed front end program to support random transaction file numbers via daily_transaction_xxx	Pass
D2	Retrofit of daily and weekly script		Weekly script retrofitted to support merged transaction files sorted by day.	Pass

### Notes:

- Initially used front end 2 to start project, pivoted to front end 1 for ease of use in terms of changing transaction file to support its random generation
- Due to how the back end is constructed, only minor modifications are needed to account for the change of front end (the use of partitioning of daily transaction variables into chunks)

## CSCI 3060U - Quick User Guide (Phase 6)

### Front End

Instructions / Setup:

- cd front\_end\_1/src
- Open command prompt at this path
- make / mingw32-make.exe command
- Open program:  
./auction\_system ./iofiles/current\_users\_accounts.txt ./iofiles/available\_items.txt  
./iofiles/daily\_transaction.txt **or**  
.\auction\_system .\iofiles/current\_users\_accounts.txt .\iofiles/available\_items.txt  
.\iofiles/daily\_transaction.txt

### Command List

Command	Command Prompt	Description
Login	login	Asks for username and password, starts a front end session upon successful login
Create*	create	Creates a new user
Delete*	delete	Deletes an existing user
Advertise*	advertise	Put an item up for auction
Bid*	bid	Bid on an existing item
Refund*	refund	Returns credit from one user to another
Add Credit*	addcredit	Adds credit to target user
List All Items	listallitems	List all items up for auction
List All Users	listallusers	List all current users
Reset Password	resetpassword	Resets current user password
Logout	logout	Ends front end session

\*These commands might / might not work depending on user type

### Back End

Prerequisites:

- Transaction files in correct directory
- Python BackEndV2.py file

IDE Method:

- Visual Studio Code supports running the back-end inside the IDE through Run Python File button in the top right of the application
- Requires no user input, will perform operations automatically, barring no errors.

### Daily Script

Instructions:

- cd scripts
- Bash dailyScript.sh

### Weekly Script

Instructions:

- cd scripts
- Bash weeklyScript.sh