

Currently, all backend related .py files are in the project directory. Refer to each table to see if the files can run at their current prototype stage. To run, simply navigate to the desired python file and run it in your editor. If running, please ensure you are in the project folder before running the file as most of the python files must read from .txt files that are located within the project folder (ex: availablegames.txt or currentusers.txt)

Class/Method	Description
DailyTransaction	Represents the collection of transactions made by the user in a single session
clearEntries()	Wipes the contents of the latest DTF
recordBuyerUsername(): String	Method to record a buyer username to daily transaction file
recordSellerUsername(): String	Method to record a seller username to daily transaction file
mergeDTF()	Merges multiple DTFs into one file (if required)

File has functions put in place now. All methods will run but there are slight issues with writing to the file itself. All features seen below are shown within this file, and are working when the file is read through the function. However, the file in file path does not seem to be updating in real time on the folder path view. This will be addressed in the next phase.

Class/Method	Description
GameCollection	Represents a collection of all the games in the system
Game Collection init(int Gamelength, List GameList)	Make variables within the class to be accessed by other functions
createEntry(String gameName, String ownerUsername): String[]	Method to create a game entry into the GameCollection file
createEndUser()	Method to create an "END" game that is placed at the end of the file
getGameName(String fileName): String	Getter for game name
getUserName(String name): String	Getter for username
readFile(String filename)	Read contents of a file (GameCollection in this case)

removeEntry(String time, String ownerUsername): String[]	Method to remove a game entry from GameCollection (in the event of a refund or account deletion)
--	--

Method functionality available. All methods can read from & write to availablegames.txt. Simply use the main method to call your desired method and open the text file or look at your terminal to see the updated file contents. Only missing feature for this class is reading from the merged DTF file and applying the changes to the file automatically as well as using sellerUsernameSect variable to properly format the games entries into the .txt file. removeEntry also does not work due to formatting issues.

Class/Method	Description
AvailableGames	Represents a collection of all currently available games in the system
createEntry(String gameName, String sellerUsername, double gamePrice)	Method to create a game entry into the AvailableGames file
readFile()	Read contents of a file (AvailableGames in this case)
createEndUser()	Method to create an "END" game that is placed at the end of the file
removeEntry(String gameName, String sellerUsername, double gamePrice)	Method to remove a game from AvailableGames (in the event of a purchase or deletion of a seller)

Method functionality available. All methods can read from & write to currentusers.txt. Simply use the main method to call your desired method and open the text file or look at your terminal to see the updated file contents. Only missing feature for this class is reading from the merged DTF file and applying the changes to the file automatically.

Class/Method	Description
CurrentUsers	Represents a collection of all the current users in the system
init(String username, char usertype, double credit)	Make variables within the class to be accessed by other functions
createEntry(String username, char usertype, double credit)	Method to create a user entry into the CurrentUsers file
createEndUser()	Method to create an "END" user that is placed at the end of the file

readFile()	Read contents of a file (CurrentUsers in this case)
increaseBalance(String username, double credit)	Method to increase the balance of a user (in the event of a refund/sale/add credit)
decreaseBalance(String username, double credit)	Method to decrease the balance of a user (in the event of a purchase)
removeEntry(String username)	Method to remove a user from CurrentUsers (in the event of a deletion by admin)

Fully functional file. This file is meant to supplement the other files in printing error messages. These methods are already being called in the other classes and does not need to be run from within this file

Class/Method	Description
ErrorLog	Organizes where all the error logs will be created from
logConstraint(String type, String transaction, String description)	Method to create a console log for a constraint error
logFatal(String type, String filename, String description)	Method to create a console log for a fatal error