***Note: All .cpp files are to be run separately from each other. Currently, most functionality is stored in the usertype files (buystandard.cpp, fullstandard.cpp, etc). Files related to different transaction files currently have limited/placeholder functionality. If you would like to run a File .cpp file, dailytransaction.cpp & file.cpp currently have the most functionality**

***Note: Some methods listed in this document as well as the UML diagram may not have been created/implemented yet. The methods in these documents are an outline of the overall system structure and may not be included in the prototype as of yet.**

Class/Method	Description
User	Represents an instance of a user on the system
login()	Method to log the user into the system
checkPerms(String username)	Method that will check the permissions of the current user

*Not functional

Class/Method	Description
BuyStandard	Represents an instance of a buy-standard user on the system
list()	Method to display all games in the available games file
buy(String gameName, String sellerUsername)	Method to allow the user to purchase a game from the available games library
addCredit(int credit, int &userAccount)	Method that lets the user add a specified amount of credit to their account
logout()	Method to log the user out of the system

*Semi functional (menu inputs/outputs)

Class/Method	Description
SellStandard	Represents an instance of a sell-standard user on the system
list()	Method to display all games in the available games file
logout()	Method to log the user out of the system

sell(String gameName, int price)	Method to put a game up for sale by the user
addCredit(int credit, int &userAccount)	Method that lets the user add a specified amount of credit to their account

*Semi functional (menu inputs/outputs)

Class/Method	Description
FullStandard	Represents an instance of a full-standard user on the system
list()	Method to display all games in the available games file
logout()	Method to log the user out of the system
sell(String gameName, int price)	Method to put a game up for sale by the user
buy(String gameName, String sellerUsername)	Method to allow the user to purchase a game from the available games library
addCredit(int credit, int &userAccount)	Method that lets the user add a specified amount of credit to their account

*Semi functional (menu inputs/outputs)

Class/Method	Description
Admin	Represents an instance of an admin user on the system
list()	Method to display all games in the available games file
listActive()	Method to display all active users on the system, along with their relevant information
create(String username, String accountType): User	Method to allow the admin to create another user on the system
delete(String existingUsername)	Method to allow the admin to delete an existing user on the system
sell(String gameName, int price)	Method to put a game up for sale by the user
buy(String gameName, String sellerUsername)	Method to allow the user to purchase a game from the available games library
refund(String buyerUsername, String sellerUsername, int credit)	Method to allow the admin to refund a game from a buyer and credit the seller accordingly

addCredit(int credit, int &userAccount)	Method that lets the user add a specified amount of credit to the specified account username
logout()	Method to log the user out of the system

*Semi functional (placeholders)

Class/Method	Description
File	Represents one of several different file types that records/monitors transactions throughout the system
writeFile(DailyTransaction file)	Method to write the daily transaction file upon logout
writeFile(String username)	Method to write username to file
writeFile(String timestamp)	Method to write timestamp of transaction to file
saveToFile()	Method to save any data to a file
recordUsername(String username): String	Method to record username
recordDailyTransaction(String transaction): String	Method to record transaction to daily transaction file

*Semi functional (placeholders)

Class/Method	Description
GameCollection	Represents a collection of all the games in the system
createEntry(String gameName, String ownerUsername): File	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

*Semi functional (placeholders)

Class/Method	Description
AvailableGames	Represents a collection of all currently

	available games in the system
createEntry(String gameName, String sellerUsername, int gamePrice): File	Method to create a game entry into the AvailableGames file
createEndUser()	Method to create an “END” game that is placed at the end of the file

*Not functional

Class/Method	Description
CurrentUsers	Represents a collection of all the current users in the system
createEntry(String username, char userType, int credit): File	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

*Not functional

Class/Method	Description
DailyTransaction	Represents the collection of transactions made by the user in a single session
addToList(File file)	Method to add a transaction to the DailyTransaction file
createFileRefund(char transactionCode, String buyerUsername, String sellerUsername, int credit): File	Method to create a refund transaction instance during the session
createFileSell(char transactionCode, String gameName, int gamePrice): File	Method to create a sell transaction instance during the session
createFileBuy(char transactionCode, String gameName, String sellerUsername, String buyerUsername, int gamePrice): File	Method to create a buy transaction instance during the session
createFile(char transactionCode, String username, char userType, int credit): File	Method to create any other existing transaction instance that was made during the session
recordBuyerUsername(): String	Method to record a buyer username to daily transaction file
recordSellerUsername(): String	Method to record a seller username to daily transaction file

*Semi functional (placeholders)