

## 3.0. UML METHOD DESCRIPTIONS

### 3.1. AuctionSystem

*The AuctionSystem Class is used to determine menu views and is the controller in the application that allows the user to perform transactions while ensuring proper authorization.*

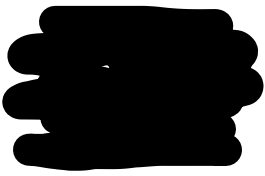
- **+ AuctionSystem(auctionStatus : bool)**
  - This function is the constructor for the class AuctionSystem which initializes the auctionStatus in order to control the run of the program.
- **+ Login(userRecord : USER\_RECORD, &currentUser : User) : bool**
  - This function assigns values to a structure named USER\_RECORD by taking the current user information retrieved from the login and returns user login status.
- **+ Logout(&currentUser : User) : void**
  - This function logs the current user out of the program and returns a success message to the console.



### 3.2. User

*The User Class is an object that's used when a user is logged in and informs the user what transactions they can perform in the system.*

- **+ toString() : string**
  - This function returns the user information in a string format.
- **+ CreateAccount() : USER\_RECORD**
  - This function allows an admin user to create another user object and handles errors with user input during the process of user creation.
- **+ DeleteAccount() : string**
  - This function allows an admin to delete another user object and their items in the auction system.
- **+ Bid() : ITEM\_RECORD**
  - This function creates an ITEM\_RECORD that can be used to verify the users bid attempt and allows the user to bid on an item that is active in the auction system.
- **+ Advertise() : ITEM\_RECORD**
  - This function allows a user to advertise an item in the auction system and handles errors with user input during the process of item advertising.
- **+ AddCredit() : float**
  - This function allows an admin user to add credits to any user or any non-admin users to add credits to their own user accounts.
- **+ Refund() : REFUND\_RECORD**
  - This function gives the user their credit back from a seller's account and handles errors with the user input during the process of refunding.
- **+ is\_number(s : const string) : bool**
  - This function is used in error handling to verify user input regarding credits is of the proper type.



### 3.3. FileController

*The FileController class is used for file Input and Output controls for text documents.*

- **+ FileController(userFile: string, itemFile : string, transFile : string)**
  - This function is the constructor for the class FileController which initializes a user-accounts file, available-items file and daily-transactions file in order to control the reading and writing to these files.
- **+ getUser(lookupName : string) : USER\_RECORD**
  - This function looks up a user from the user-account file and returns a USER\_RECORD of the user.
- **+ findUser(lookupName : string) : bool**
  - This function searches the user-accounts file and returns true if user found.
- **+ addUser(newUser : USER\_RECORD) : void**
  - This function adds a USER\_RECORD of the user to the user-accounts file.
- **+ deleteUser(username : string) : void**
  - This function looks up the user to delete from the user-accounts file and removes it from the user-accounts file if it exists.
- **+ updateCredit(username : string, credit : float) : void**
  - This function updates the credit balance of the user based on their provided username and current credit balance.
- **+ addItem(itemRecord : ITEM\_RECORD) : void**
  - This function adds an ITEM\_RECORD of the item to the available-items file when the user inputs the 'advertise' operation.
- **+ displayAvailableItems() : void**
  - This function displays all of the existing items that are in the available-items file when the user inputs the 'listall' operation.
- **+ getItemBid(itemRecord : ITEM\_RECORD) : float**
  - This function returns the current highest bid on an item active in the auction by checking the available-items file. It returns -1 if the item record is not found in the available-items file for the item requested to be bid.
- **+ updateItemBid(itemRecord : ITEM\_RECORD, buyer : string) : void**
  - This function updates the current highest bid amount on an item active in the auction by checking the available-items file and validating the buyer.
- **+ logTransaction(transactionNumber : string, output : string) : void**
  - This function writes a transaction to the daily-transaction file using the transaction number based on the operation and the recorded information from that transaction.