

**Software Quality Assurance**  
**CSCI 3060U**  
**Phase 2: Front End Rapid Prototype**

Group Name:

Jessica (100657595)

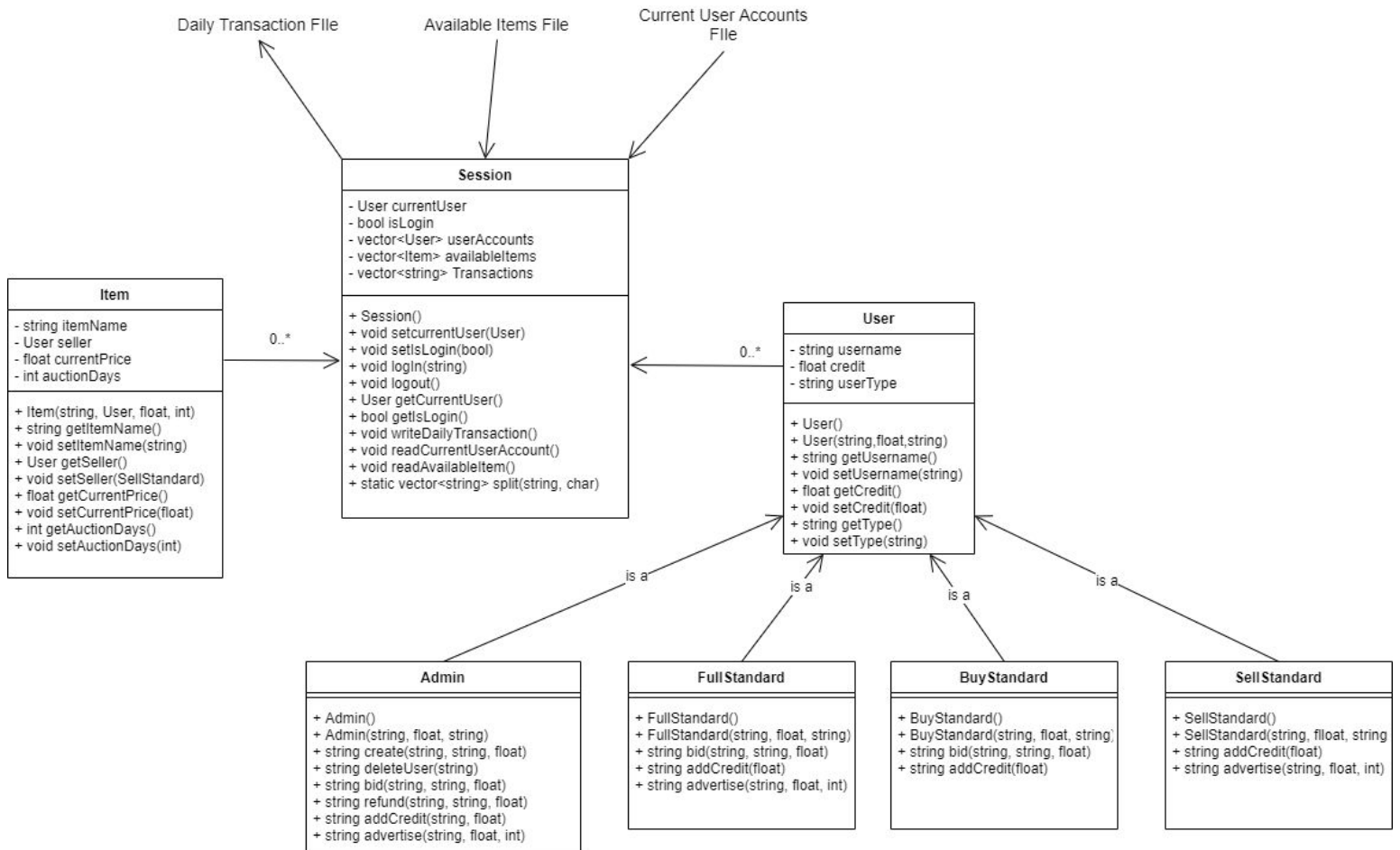
Terry Darmawan Hosea (100670290)

Pulkit Madan (100674520)

Teaching Assistant Name: **Jude Arokiam**

# UML Design Diagram

Also available in: <https://github.com/terryhosea/csci3060u-project/>





## session

Session is the core of the application, in which it will keep track of the login status, read user accounts file, read available items file, write to the daily transaction files whenever the logout transaction is invoked.

Function	Purpose	Input	Output
Session()	Default constructor to initialize an instance of Session	None	None
Session(string)	Overriding the default constructor by initializing an object of class Session with given argument	-string: Specify the username of the user	None
User getCurrentUser() ( )	Get the current user	None	Returns an object of type User
Bool getIsLogin() ( )	Get the status of current user in session	None	Boolean value (true/false)
Void setCurrentUser (User)	Set the user that is currently in the session	-User: Object of type User	None
Void setIsLogin(bool )	Set the login status of the current user to the passed flag	-bool: A boolean flag	None
Void login(string)	Logs in the user with the entered username	-string: Username of the user	None
Void logout() ( )	Logs out the current user from the session	None	None
Void writeDailyTrans action() ( )	Write the content of transaction vector to the daily transaction file	None	None
Void readCurrentUs erAccount() ( )	Read the available items file and add them to the userAccounts	None	None

	vector		
readAvailableItems()	Read the available items file and add the available entries to the availableItems vector	None	None
Static vector<string> split(string, char)	Split a string into several strings which is separated by a character	-string: Original string that will be split -char: Character that will be used to determine the split	A string vector containing separated string

## item

The item class is used to create an instance of Item class that will be stored in an item vector in a session instance after a successful login.

Function	Purpose	Input	Output
Item(string, User, float, int)	Constructor that initiates an instance of class Item with given parameters	-string: Name of the item -User: User that is selling the item -float: Current highest bid for the item -int: Number of days that item is available for auction	None
Void setItemName(string)	Set the name of the item to the passed string parameter	-string: Name of the item	None
String getItemName()	Get the name of the item	None	Returns the item's name

User getSeller()	To get the seller of the item	None	Returns the seller
Void setSeller(User)	Set the seller of the current item	-user:Instance of User	None
Float getCurrentPrice()	Get the current highest bid for the item	None	Current highest bid returned as a float
Void setCurrentPrice(float)	Sets the current highest bid to the given price	-float:Float for current price	None
Int getAuctiondays()	Get the number of days that the item is available for auction	None	Number of days the item is available for auction
Void setAuctionDays(int)	Set the number of days that the item is available for auction	-int:Number days as an integer	None

## User

User class is the parent class for all user types(Admin, FullStandard, BuyStandard, SellStandard). Containing the username, userType, and credit of the user

Function	Purpose	Input	Output
User()	Default constructor for user class, initializes an object of User	None	None
User(string, float, string)	Constructor that initiates an instance of class User with given parameters	-string: The username of the user -float: The amount of credit of the user -string: The account type of the user	None
String getUsername()	Get the name of the username	None	String which is the username of the user
Void setUsername(string)	Set the name of the user to a passed argument	-string:String for username of the user	None
Float getCredit()	Get the available credit for the user	None	Float which is the amount of user's credit
Void setCredit(float)	Set the credit amount of the user to a passed argument	-float: Float to set the amount of the credit of the user	None
String getType()	Get the type of the user	Noen	String which is the account type of the user
Void	Set the type of the	-String: String	None

setType(string)	user to a passed argument	specifying the type of user	
-----------------	---------------------------	-----------------------------	--

## Admin

Admin class that is used to create admin accounts, containing all the commands including privileged and non-privileged commands

Function	Purpose	Input	Output
Admin()	Default constructor that initializes an object of Admin	None	None
Admin(string, float, string)	Constructor that initiates an instance of class Admin with given parameter	-string:string for the username of the user -float:float for credit of the user -String:string for the account type of the user	None
String create(string, string, float)	Create transaction to create a new user	-String:string for the new user's username -String: String for the new user's account type -Float: Float for the new user's credit	Returns a formatted string to write to daily transaction file
String deleteUser(string)	Delete transaction to delete a user with the specified username	-string:The username of the user that will be deleted	Returns a formatted string to write to daily transaction file
String bid(string, string, float)	Bid transaction to place a bid on an existing item	-string:Name of the item on which bid is to be placed -string:Username of the user who advertised the item	Returns a formatted string to write to daily transaction file



		float:Value of bid being placed	
String refund(string, string, float)	Issue a refund amount from seller to buyer	-string:Username of the seller issuing the refund -string:Username of the user getting the refund -float:The amount being refunded	Returns a formatted string to write to daily transaction file
String addCredit(string, float)	To add credit of a specific amount to a user	-String: Username of the user getting the credit -float:Amount of credit being added	Returns a formatted string to write to daily transaction file
String advertise(string, float,int)	Advertise command to create a new advertisement of an item	-string:Name of the item -float:Initial price of the advertised item -int: Number of days the item is going to be in auction	Returns a formatted string to write to daily transaction file

## FullStandard

Admin class that is used to create admin accounts, containing all the commands including privileged and non-privileged commands

Function	Purpose	Input	Output
FullStandard()	Default constructor that initializes an object of FullStandard	None	None
FullStandard(string, float,string)	Overloading the default constructor to create an instance of class FullStandard with	-string:Username of the user -float:Credit in the user's account -string:The account	None

	the given parameters	type of the user	
String bid(string,string,float)	Bid transaction to place a bid on an existing item	-string:Name of the item on which bid is to be placed -string:Username of the user who advertised the item float:Value of bid being placed	Returns a formatted string to write to daily transaction file
String addCredit(string,float)	To add credit of a specific amount to a user	-String: Username of the user getting the credit -float:Amount of credit being added	Returns a formatted string to write to daily transaction file
String advertise(string,float,int)	Advertise command to create a new advertisement of an item	-string:Name of the item -float:Initial price of the advertised item -int: Number of days the item is going to be in auction	Returns a formatted string to write to daily transaction file

## BuyStandard

Admin class that is used to create admin accounts, containing all the commands including privileged and non-privileged commands

Function	Purpose	Input	Output
BuyStandard()	Default constructor for class that initializes an object of BuyStandard	None	None
BuyStandard(string,float,string)	Overloading the default constructor to create an object of the class with the given parameters	-string:Username of the user -float:Credit in the user's account -string:The account	None

		type of the user	
String bid(string,string,float)	Bid transaction to place a bid on an existing item	-string:Name of the item on which bid is to be placed -string:Username of the user who advertised the item float:Value of bid being placed	Returns a formatted string to write to daily transaction file
String addCredit(string,float)	To add credit of a specific amount to a user	-String: Username of the user getting the credit -float:Amount of credit being added	Returns a formatted string to write to daily transaction file

## **SellStandard**

Admin class that is used to create admin accounts, containing all the commands including privileged and non-privileged commands

<b>Function</b>	<b>Purpose</b>	<b>Input</b>	<b>Output</b>
SellStandard()	Default constructor for class that initializes an object of SellStandard	None	None
SellStandard(string,float,string)	Overloading the default constructor to create an object of the class with the given parameters	-string:Username of the user -float:Credit in the user's account -string:The account type of the user	None
String advertise(string,float,int)	Advertise command to create a new advertisement of an item	-string:Name of the item -float:Initial price of the advertised item -int: Number of days the item is going to be in auction	Returns a formatted string to write to daily transaction file

String addCredit(string, float)	To add credit of a specific amount to a user	-String: Username of the user getting the credit -float:Amount of credit being added	Returns a formatted string to write to daily transaction file
---------------------------------------	--	--	---