CS 2316
Homework 9b – GT Thrift Shop
Due: Wednesday, April 20th, before 11:55 PM
Out of 100 points

Files to submit: 1. HW9b.py
2. any image files (.gif) used in database

<u>This is an INDIVIDUAL assignment!</u>

Collaboration at a reasonable level will not result in substantially similar code. Students may only collaborate with fellow students currently taking CS 2316, the TA's and the lecturer. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. You should not exchange code or write code for others.

For Help:
- TA Helpdesk – Schedule posted on class website.
- Email TA's or use T-Square Forums

Notes:

- **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
- ***Do not wait until the last minute*** *to do this assignment in case you run into* problems.
- **Read the entire specifications document before starting this assignment.**

Premise

This homework is part two of the previous GT Login assignment. You will add on to the login functionality that you built to create more GUI windows that use a database to sell and buy items. Although this program does not require knowing much new information, it does test SQL query and insertion abilities as well as coding practices to create more GUI windows that can be opened, closed, or updated at the click of a button. **This new information may take some understanding, so please do not wait until the last minute to begin this assignment.**

Database Format Information

The database for this part of the assignment uses a different table called GTThrift:

```
CREATE TABLE schema.GTThrift(
ID INT AUTO_INCREMENT PRIMARY KEY,
Username VARCHAR(25) NOT NULL,
ItemName VARCHAR(25) NOT NULL,
Description VARCHAR(50),
Price DECIMAL(6,2) NOT NULL,
Used boolean DEFAULT TRUE,
Purchased boolean DEFAULT FALSE,
Image VARCHAR(20) NOT NULL,
BuyerUsername VARCHAR(25),
foreign key(Username) references schema.user(Username),
foreign key(BuyerUsername) references schema.user(Username));
```

The code above was used to create the table, GTThrift, be sure to change the schema name. Each column name and its functionality in the homework:

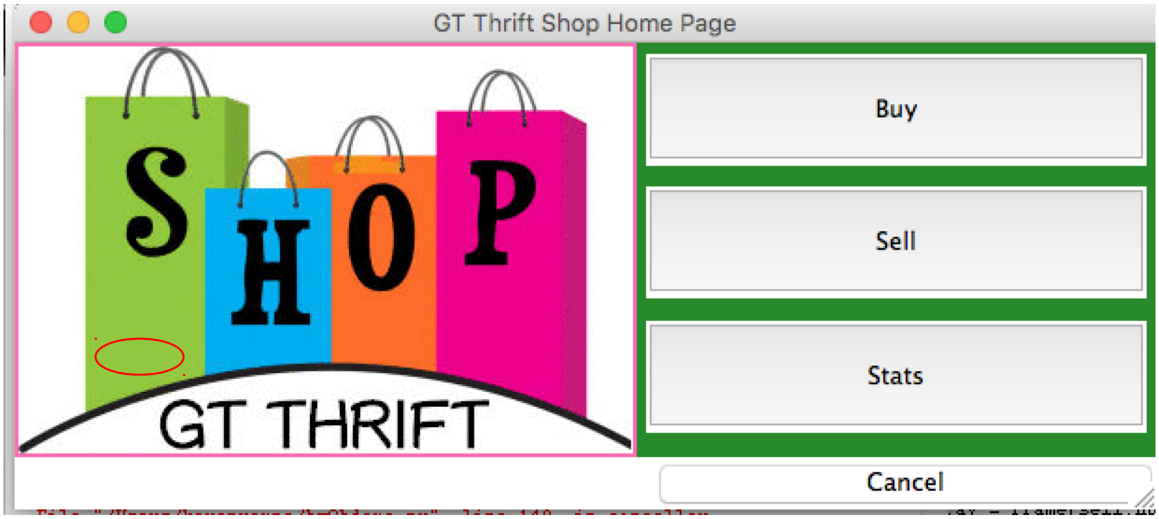
- **ID** automatically increments a number in order of items inserted into the table. Think of ID as a receipt number in the order of the time it was inserted.
- **Username** is NOT NULL meaning something must be inserted into the field. It is also a foreign key to the User table created in HW9A-Login. That means that the Useranme must exist in the User table to be inserted into the GTThrift table.
- **ItemName** is NOT NULL meaning you must insert something into the field.
- **Description** is limited to 50 characters.
- **Price** field is NOT NULL meaning you must insert something into the field. Decimal(6,2) means that there can be up to 4 digits before the decimal and 2 after the decimal (EX. 1000.00 is acceptable but 10000.00 is not).
- **Used** is a boolean with values 1 for True and 0 for False to specify whether the condition of the item is 1 for Used and 0 for New.
- **Purchased** is a Boolean with values 1 for bought and 0 for sale. The default value is 0 for sale when it is inserted into the GTThrift table.
- **Image** is NOT NULL meaning something must be inserted into the field. It is limited to 20 characters. Image is the name of the gif file that must be submitted with the python script.
- **BuyerUsername** is the Username that purchased the item. If the item has not been purchased yet, then the value is NULL. The BuyerUsername is a foreign key to the User table and the Username must exist in the User table to be inserted.

Further Development

You will be writing on top of your already written part A. **Your old code is probably not perfect so feel free to update it if needed! Keep in mind that the GUIs created do not need to look exactly the same.**

The new code will create a space to buy, sell, or see statics. The statistics page displays some of the more avid users and their stats.

Method Name: **HomePage**



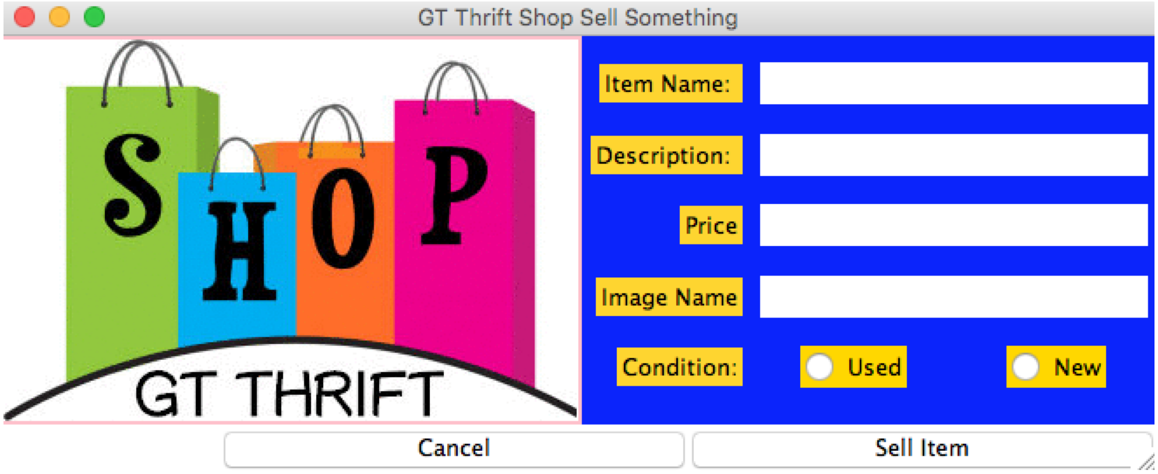
This method is called to show the home page. It should be called when you successfully login from the previous homework's login page. You should create a GUI which looks like the one above. You may generate this window however you wish, but it is *highly* suggested that you use the grid layout manager.

Things to note about the GUI you see:

- The image is the same image as in your other GUI windows from part A. You can continue to use the same image from the previous homework, or switch to a different image if you wish, but keep it GT Thrift Shop related! *Make sure to upload it with your python script.*
- The background of the button's frame is a fun color. Feel free to use any color and font!
- The title is GT Thrift Shop Home Page.
- The Buttons are within a frame and have a width of 30, a height of 3, and a FLAT relief. Each button has a command to its own associated method. The Buy button calls the Buy method, Sell button calls the Sell method, and Stats button calls the Stats method.
- Cancel button calls a helper method that terminates the program.
- *Feel free to use any color scheme for the background (as long as it's not white).* The different background colors were created by making different frames. *http://www.science.smith.edu/dftwiki/images/3/3d/TkInterColorCharts.png*

Method Name: **Sell**

This method should close the homepage window and will show a new GUI window for selling an item in GT Thrift Shop. The GUI should look something like:



You may generate this window however you wish, but it is *highly* suggested that you use the grid layout manager. Things to note about the GUI you see:

- The image is the same image as in your other GUI windows from part A. You can continue to use the same image from the previous homework, or switch to a different image if you wish, but keep it GT Thrift Shop related! *Make sure to upload it with your python script.*
- The title is GT Thrift Shop Sell Something.
- Each entry is width 30 and state normal.
- The labels are all sticky to the east with a pady and padx of 5.
- Clicking on the "Cancel" button calls a helper function to hide this GUI window and redisplay the Home Page Window.
- Clicking on the "Sell Item" button calls the SellInsert method.

- *Feel free to use any color scheme for the background (as long as it's not white).* The different background colors were created by making different frames.

For every item sold there must be an image of the item. The image should have a maximum pixel width of 250 pixels and is proportional to its pixel height. The image should be the type .gif. The file name of the image should match the entry to Image Name. *Make sure to upload the images with your python script.*

Method Name: **SellInsert**

Description:

This method will check the inputted information by the user from the sell window and if it meets the conditions then it will be inserted into the GT Thrift table. The Username should be obtained from the login. The ItemName, Price, and Image must be inputted to sell an item since their values in the GT Thrift table are NOT NULL. Check that the price follows the condition of DECIMAL(6,2). Check there is an Image present in current working directory by using:

```
import os
os.getcwd() # to see the current working directory
os.chdir('/Users/kerenrempe/hw9a) #to change the directory to where the Images are stored
imagelist = os.listdir() #a list of files in a directory
```

For example:

imagelist ← ['Beats.gif', 'Futton.gif', 'iPhone_6s.gif', 'Kate_Spade_Wallet.gif', 'Kindle.gif']

If any required information is missing then use a messagebox with the title "Missing Information" and the message of what required information is not inputted correctly. If the required information is put in by the user correctly, then insert all the information provided into the GTThrift table. Do not forget to commit the information to the database.

Method Name: **Buy**

Description:

This method will display the items for sale in the GT Thrift table. By default, the top 5 items with the highest prices will be listed. There are several options to sort the information. The options of the drop down menu are:

```
OPTIONS = ["Price: Low to High", "Price: High to Low", "New", "Old", "A-Z", "Z-A"]
```

Where the New and Old can be obtained by the order of the ID with lowest numbers being New. A-Z and Z-A refer to the ItemName in the GT Thrift table.

This method should close the homepage window and will show a new GUI window for selling an item in GT Thrift Shop. The GUI should look something like:

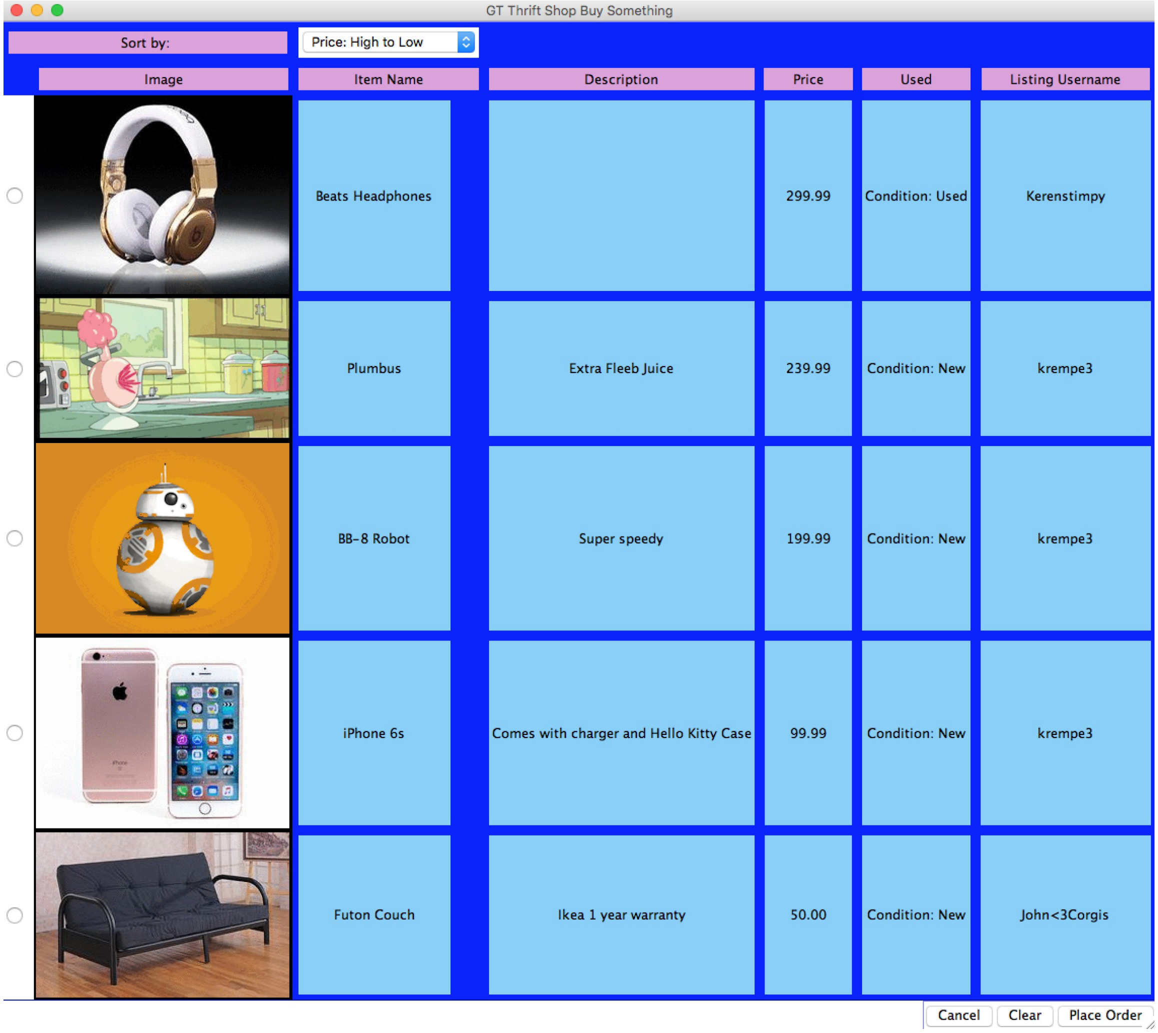


	Image	Item Name	Description	Price	Used	Listing Username
○		Beats Headphones		299.99	Condition: Used	Kerenstimpy
○		Plumbus	Extra Fleeb Juice	239.99	Condition: New	krempe3
○		BB-8 Robot	Super speedy	199.99	Condition: New	krempe3
○		iPhone 6s	Comes with charger and Hello Kitty Case	99.99	Condition: New	krempe3
○		Futon Couch	Ikea 1 year warranty	50.00	Condition: New	John<3Corgis

You may generate this window however you wish, but it is *highly* suggested that follow the directions for the kind of layout manager to use. Things to note about the GUI you see:

- The title is GT Thrift Shop Buy Something.
- There are three frames within the window. The first frame holds the "Sort by" Label and the drop down menu. The frame uses the pack lay out manager with side =TOP and fill = both. The second frame contains the header row and the GT Thrift shop data. Within the frame the Labels use the grid layout manager. Each Label has a pady and padx of 5, uses sticky E+W, and has a width of at least 10. It is advised that the Radiobutton values for each GT Thrift Shop item assigns the ID value. The third frame contains the buttons 'Cancel', 'Clear', and 'Place Order'. The 'Cancel' button closes the Buy window and returns to the home page using a helper method. The 'Clear' button deselects any radio button that may be selected by utilizing a helper method. The 'Place Order' button calls the BuyUpdate method.

Method Name: **BuyUpdate**

Description:

This method is invoked when 'Place Order' button is pushed in the Buy method. The method should check that a radio button was selected in the Buy window. If no radio button is selected then a messagebox should appear with the title 'Whoops!' and the message 'Please select an item to purchase.'

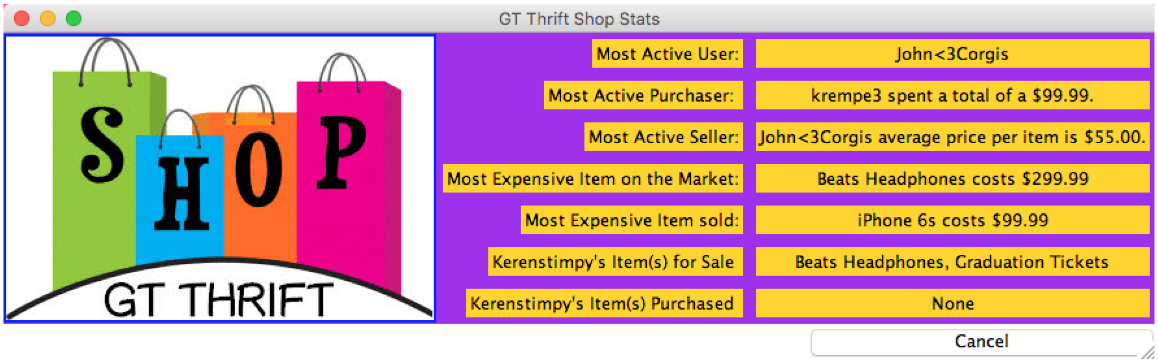
If a radio button is selected, then the value of the radio button should match the ID of an item in the GT Thrift table which can be used to update the item Purchased to true and the BuyerUsername to the current user logged in. The method should show a messagebox with the title 'Successful Purchase' and the message ‘Congratulations you have purchased ItemName for Price from Username.” With the appropriate information from the purchase filled in. Then the Buy window should be withdrawn and should return the user to the home page.

Method Name: **Stats**

Description:

This method will display statistics for the GT Thrift table. The statistics are obtained from the GT Thrift table using SQL statements. The GROUP BY and ORDER BY SQL functions may be helpful and remember that you can aggregate more than one column at one time. In addition, to increase the speed it may be useful to use the LIMIT or MAX() SQL function to get the highest value.

The home page will withdraw and a new GUI window will be created looking something like this:



Most Active User:	John<3Corgis
Most Active Purchaser:	krempe3 spent a total of a $99.99.
Most Active Seller:	John<3Corgis average price per item is $55.00.
Most Expensive Item on the Market:	Beats Headphones costs $299.99
Most Expensive Item sold:	iPhone 6s costs $99.99
Kerenstimpy's Item(s) for Sale	Beats Headphones, Graduation Tickets
Kerenstimpy's Item(s) Purchased	None

You may generate this window however you wish, but it is *highly* suggested that you use the grid layout manager. Things to note about the GUI you see:

- The title is GT Thrift Shop Stats
- The top image is the same image as in your other GUI windows from part A. You can continue to use the same image from the previous homework, or switch to a different image if you wish, but keep it GT Thrift Shop related! *Make sure to upload it with your python script.*
- The title is GT Thrift Shop Stats.
- The labels in column 1 are sticky to the east with a pady and padx of 5.
- The labels in column 2 are sticky east to west with a pady and padx of 5
- Clicking on the “Cancel” button calls a helper function to hide this GUI window and redisplay the Home Page Window.
- *Feel free to use any color scheme for the background (as long as it’s not white).* The different background colors were created by making different frames.

The stats were obtained from the GT Thrift Table and include the following details:

- **Most Active User –** is the most active overall and should be found by evaluating the number of IDs for each User where there have either bought or sold
- **Most Active Purchase –** gets the username and sum of the money spent by the most active purchaser.
- **Most Active Seller –** gets the username and average price per item by the most active seller.
- **Most Expensive Item on the Market –** lists the most expensive item name and the cost of the item of an item that has not been purchased.
- **Most Expensive Item Sold –** lists the most expensive item name and the cost of the item of an item that has been purchased.
- **Current Username’s Item(s) for Sale –** lists all the items the current user has put up for sale. If none is listed, then the label should have the text None.
- **Current Username’s Item(s) Purchased -** lists all the items the current user has purchased. If none is listed, then the label should have the text None.

Grading:

You will earn points as follows for each piece of functionality that works correctly according to the specifications.

Login GUI		**5**
Login GUI works (as specified in previous HW)	5	
Register GUI		**5**
Register Page works (as specified in previous HW)	5	
HomePage GUI		**10**
Logo and title correct	2	
Buttons all linked correctly	8	
Sell GUI		**15**
Logo and title correct	1	
GUI elements proper and filled	10	
Buttons all linked correctly	4	
Buy GUI		**25**
GUI elements proper and filled	15	
Frame 1	3	
Frame 2	10	
Frame 3	2	
Buttons all linked correctly	6	
Drop Down Menu Functionality Correct	4	
SellInsert		**10**
SQL functionality is correct	5	
Messagebox checks for missing info.	5	
Commits to db when done		
BuyUpdate		**15**
Checks for no selected radiobutton (Messagebox)	5	
SQL functionality is correct	5	
Commits db when done	2	
Successful Purchase (Messagebox)	3	
Stats GUI		**15**
GUI elements proper and filled	4	
SQL statements are correct	7	
Cancel button works correctly	4	

If connects to database more than 4 connections in program (-10)