

CS 2316

Homework 9a – GT Room Reservation Login

Due: Wednesday November 5th

Out of 100 points

Files to submit: 1. HW9.py

This is an INDIVIDUAL assignment!

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 the first of a two-part assignment in which you will do some real world applications with databases and GUIs. For this first part of the assignment, you will be building a two page GUI that offers the user the ability to login, or in the case that the user does not have an account yet, register (create an account) and then log in. You'll find this quite applicable to building programs with user accounts, and potentially many future CS4400 projects.

This program exercises simple SQL query and insertion abilities as well as new coding practices to create two GUI windows that can be opened and closed with 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.** *Additionally, if you have not yet confirmed that your database username/password work, you should do that immediately, as it can take multiple days to resolve issues with database user accounts!*

Database Format Information

The database for this part of the assignment only has 1 table of users, called ReservationUser. It was created with the following SQL statement:

```
CREATE TABLE ReservationUser
(Username          VARCHAR(15)    PRIMARY KEY,
 Password          TEXT          NOT NULL,
 LastName          TEXT,
 NumberOfReservations  INTEGER    DEFAULT 0)
```

As you can see, the Username and Password are non-NULL variables, but LastName can be left blank and NULL will be assigned instead of a name. Similarly, at the time of account creation (registration), NumberOfReservations is assigned a default value of 0. Note that the Username and LastName are different. A Username is required, but a

LastName is not. A Username can only be stored in the database if it is 15 characters or less.

Multiple GUI Pages

Multiple GUI windows may not seem difficult, but it takes a certain method to show new windows while hiding old ones. It will involve the coding and of several helper methods to transition between windows. You may find it easier to separate the creation code for each window into it's own method, so that your `__init__` method, for instance, would not actually create each GUI window, but instead call a method to do so. There will be two methods for GUI creations, two windows for writing and retrieving user information, and several helper methods for transitioning between the windows. It's up to you as to how many transition methods you will need, but one per transition should be enough. In these transition methods, you will `.withdraw()` a certain `TopLevel` window instance created for a certain window, while calling a method that deiconifies a different window instance for a different window. This means that you will be using one main `Tk()` instance for your main window and a `TopLevel` window instance for your other window(s).

The following is a suggested way to build your code, but you may choose to use a different number of methods with different names if you wish. Please document your code so that the TA's will understand what each method does.

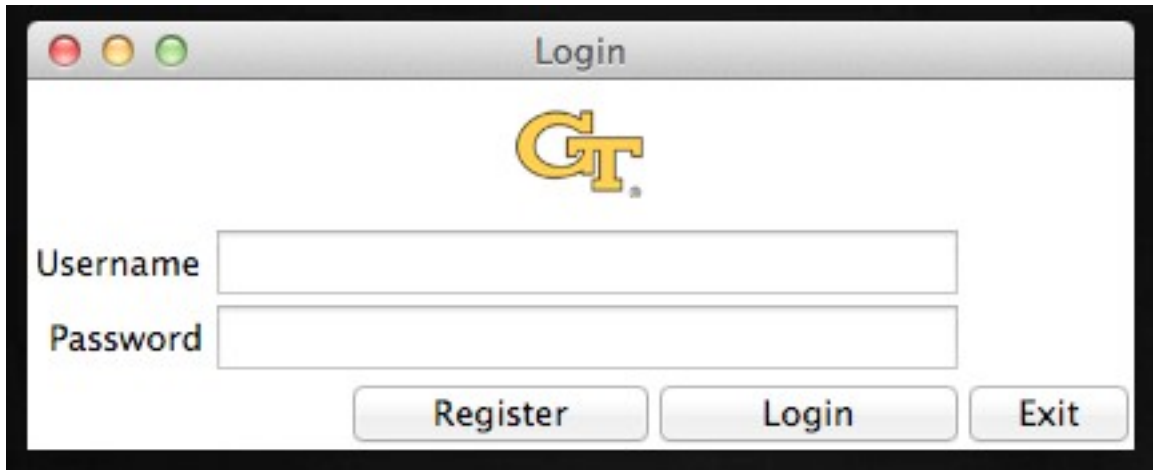
Method Name: `__init__`

This method will set things up be used in your program. For example, it may call helper methods to set up each window, and then a transition method to show the `LoginPage` upon initial program startup.

Method Name: `LoginPage`

This method is called to create a new instance of your GUI class. This

method is responsible for arranging and initializing your GUI for the Login window. You should create a GUI which looks like the one below:



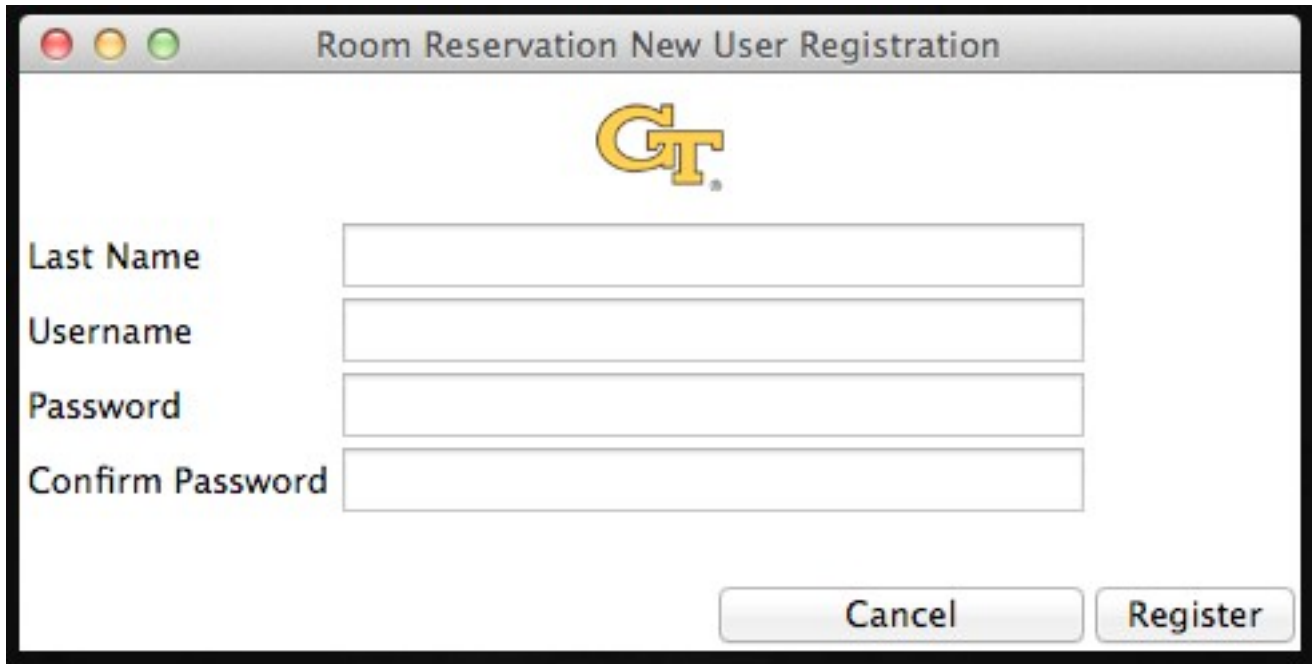
You may generate this window however you wish, but it is suggested that you use the grid layout manager. Things to note about the GUI you see:

- The top image can be any logo that is GT related. You must use an image found anywhere online that allows webscraping and retrieve the image using the `urllib.request` module. (in other words, you must download and display the image dynamically from the URL, you may not include the image as a file!)
- The labels are sticky to the east and the entry boxes are width 30 and state normal.
- The window is titled “Login”
- Clicking on the “Register” button should switch to the Register window (possibly by calling a transition method).
- Clicking on the “Login” button should call the `LoginCheck` method
- Clicking on the “Exit” button should close the Login window. You may find it useful to utilize a helper function to carry out this functionality.

Method Name: Register

This method will simply create a new GUI window for filling out registration

information. The GUI should look like the following You may generate this window however you wish, but it is suggested that you use the grid layout manager. Things to note about the GUI you see:



The screenshot shows a window titled "Room Reservation New User Registration". At the top center is a yellow "GT" logo. Below the logo are four text input fields, each with a label to its left: "Last Name", "Username", "Password", and "Confirm Password". At the bottom right of the window are two buttons: "Cancel" and "Register".

- The top image is the same as the one in the login page.
- Each entry is width 30 and state normal
- The labels are all sticky to the west
- The window is titled “Room Reservation New User Registration”
- Clicking on the “Cancel” button will call a helper method (which you could call BackToLogin for example..) to hide this GUI instance and redisplay the Login window.
- Clicking on the “Register” button calls the RegisterNew method.

Method Name: Connect

Parameters: None

Return Value: Database connection

Description: This method will use a try/except statement to attempt to connect to the database. If it cannot connect, have it display a message box asking the user to check their Internet connection. This method returns the database connection object. Use the database username and password we provided you.

Method Name: RegisterNew

Parameters: None

Return Value: None

Description: This method will take the entries from the register page and put them into the database. You must check that both passwords entered in the two entries are the same (match) and that the username and password entries are not left blank. Check that the password contains at least one number and at least one uppercase letter. Use message boxes to describe any problems with the way the register page was filled out. Also, be sure to check to see if the username is already used within the database. Note that text in our database is considered the same regardless of case, so “SummetJ” and “summetJ” would be considered the same username. You will need to check for existing usernames in a case- insensitive way if you are using Python to do the check instead of using the SQL database engine with an SQL query. If so, use a message box to tell the user that the username already exists, and don't try and add it again! If everything is ok, you can insert the information into the database. (NOTE: The LastName entry can be left blank. In this case, no LastName should be inserted into the database). Once registered, a message box should tell the user that they are now registered and a helper method should be called to hide the registration window and display the Login window. Be sure to .commit() the database when you're done inserting data!

Method Name: LoginCheck

Parameters: None

Return Value: None

Description: This method will check to see if what was typed into the login page matches any users currently in the database. (both the Username AND the password must match!) To do this, you call the Connect method, create a cursor from the database connection that is returned, and check what was passed to see if any matches are returned. If it finds no matches, it should display a message box explaining that the user entered an unrecognizable username/password combination. If it does match, you should display a message box saying you logged in successfully. You should also close the login GUI window, as you are finished with it.

A sample Room Reservation ID that already exists in the database is username: jsummet and password: Jay007. You can try using this to test your login code before you have your registration code working. Please do not mess up this account information or other students will not be able to successfully use it for testing.

Grading:

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

LoginPage GUI		20
GUI has image at top	4	
GUI has all components with proper characteristics	5	
Login button works as described	4	
Register button works as described	4	
Exit button exits the system and closes all windows	3	
RegisterPage GUI		20
GUI has image at top	6	
GUI has all components with proper characteristics	5	

Register button works as described	4	
Cancel brings user back to login page	5	
LoginCheck()		25
Uses Connect() properly	5	
Correctly checks to find matching user and password	10	
Uses message boxes for errors with user match	5	
Closes GUI with match	5	
RegisterNew()		25
Uses Connect() properly	5	
Correctly puts user in database	10	
Uses message boxes for errors with user entries	4	
Checks for usernames already taken	4	
Commits db when done	2	
Connect()		10
Correctly connects to database and returns database	10	