

CS 2316

Homework 9b – GT Room Reservation

Due: Wednesday November 12th

Out of 100 points

Files to submit: 1. HW9b.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 part B of the previous GT Login assignment. You will add on to the login functionality that you built to create three more GUI pages that use a database to have the functionality of a simple room reservation application.

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 two more GUI pages 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 the ReservationUser table from Part A, along with a different table of reservations, called RoomReservations:

```
CREATE TABLE RoomReservations
(Building          VARCHAR(9)           PRIMARY KEY,
Floor             INTEGER             PRIMARY KEY,
RoomNo           INTEGER             PRIMARY KEY,
Day              VARCHAR(10)         PRIMARY KEY,
Time             VARCHAR(5)          PRIMARY KEY,
ReservedBy       VARCHAR(15)         NOT NULL )
```

The code above was used to create the table. As can be seen, combination of Building, Floor, RoomNo, Day and Time forms the primary key for this table and therefore CANNOT BE NULL, meaning you must insert

something into each field. Note that the “ReservedBy” field is the person’s USERNAME (which is stored in the ReservationUser table).

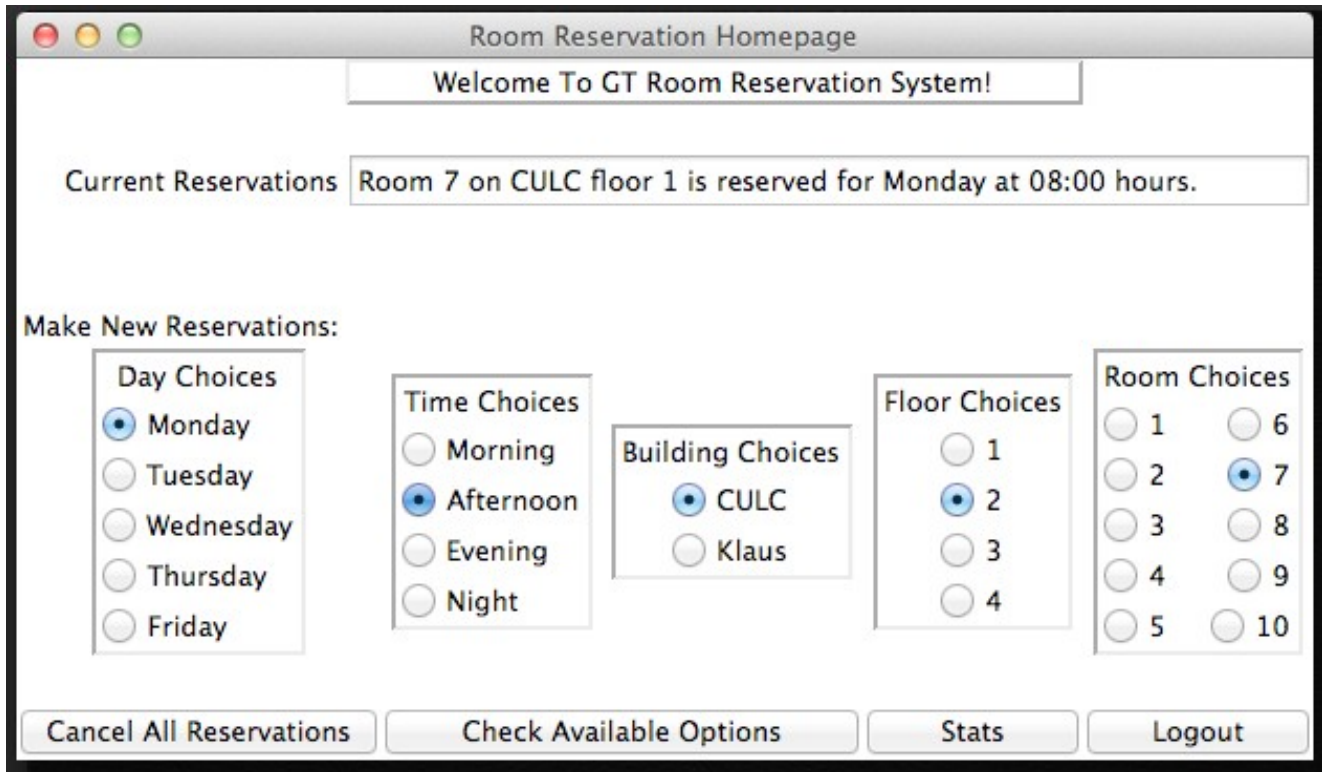
Further Development

You will be writing on top of your already written Homework 9a. **Your old code is probably not perfect so feel free to update it if needed!**

The new code will create a "home" reservation page. On this page, users have to choose their preference from each category to make a new reservation. It will also create an Available Rooms page and a Statistics page, if called upon by the respective buttons. These pages will be explained in detail later in the assignment description.

Some useful pointers about the database:

- The database is designed to allow reservation of rooms from Monday through Friday.
- Each building (CULC and Klaus) has 4 floors with 10 rooms on each floor.
- A user is allowed to make **no more than 2** reservations in this system.
- Time Frames are structured in the following manner:
 - Morning (08:00 – 11:00)
 - Afternoon (12:00 – 15:00)
 - Evening (16:00 – 19:00)
 - Night (20:00 – 23:00)
- You can assume that each reservation is made for exactly one hour. Therefore, one room can be reserved up to 16 times in a day.



Method Name: Homepage

This method is called to create a new homepage window. It should be called when you successfully login from the previous homework's login page. This method is responsible for arranging and initializing your GUI. 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 Homepage window has the title 'Room Reservation Homepage'
- Centered is a label with raised relief
- The value(s) for Current Reservation are in entry boxes with text containing the reservations listed under the current user in RoomReservations Table. You will find this data by querying the database. Hint: You may find the 'ReservedBy' attribute helpful in finding the reservations for the current user.

- If a user has more than one current reservations, you should display the second one in a different entry box placed directly below the first one.
- If the user, however, does not have any reservations in the system yet, you should display a single entry box with text as ‘No Reservations’.
- Each entry box corresponding to current reservations is in the read-only state with width 50.
- Each category (Day Choices, Time Choices, etc.) is in a separate frame with sunken relief.
- None of the radiobuttons, in any of the five frames, should be selected when the homepage window is first initiated.
- The ‘Cancel All Reservations’, ‘Check Available Options’ and ‘Statistics’ buttons call the `cancelReservation`, `availableReservations` and `stats` functions respectively.
- The ‘Logout’ button closes the Homepage window and returns the user to the Login page. You may find it useful to utilize a helper function to carry out this functionality.

Method Name: availableReservations

The ‘Check Available Options’ button calls this method. This function will extract all the time slots the selected room is available at.

Possible Errors:

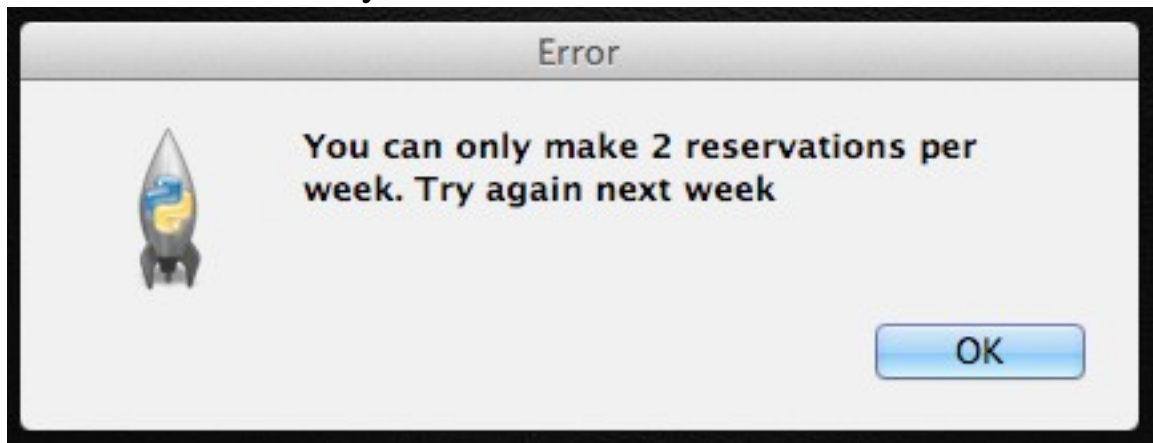
- If the user failed to select a radiobutton from each category the following error message should be displayed.



- If the chosen room is booked out for all time slots in the selected time frame.

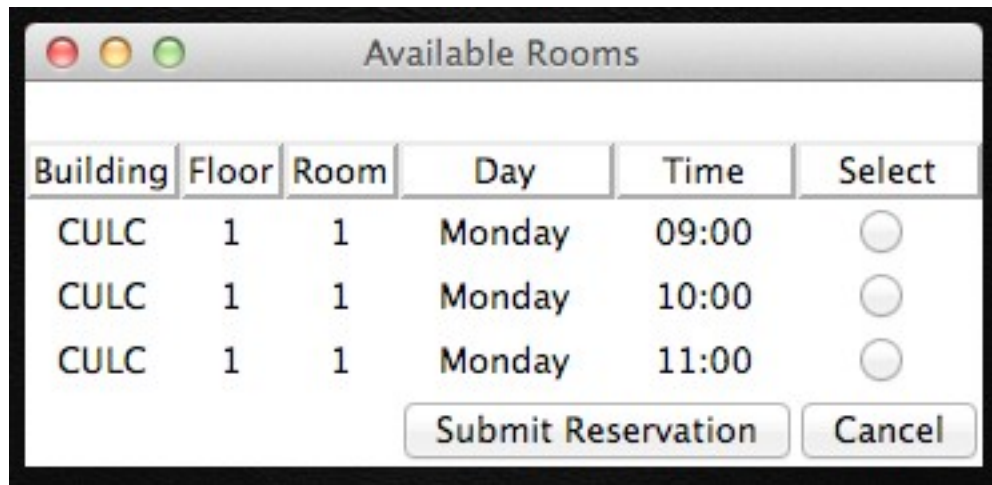


- If the user has already made two reservations.



If all goes well, you will hide the HomePage, create a new window and display all the available time slots to the user. You will find this data by querying the database.

HINT: You would display the results only if there does not exist data in RoomReservations Table for the time slots in the selected time frame.



Things to note about the GUI:

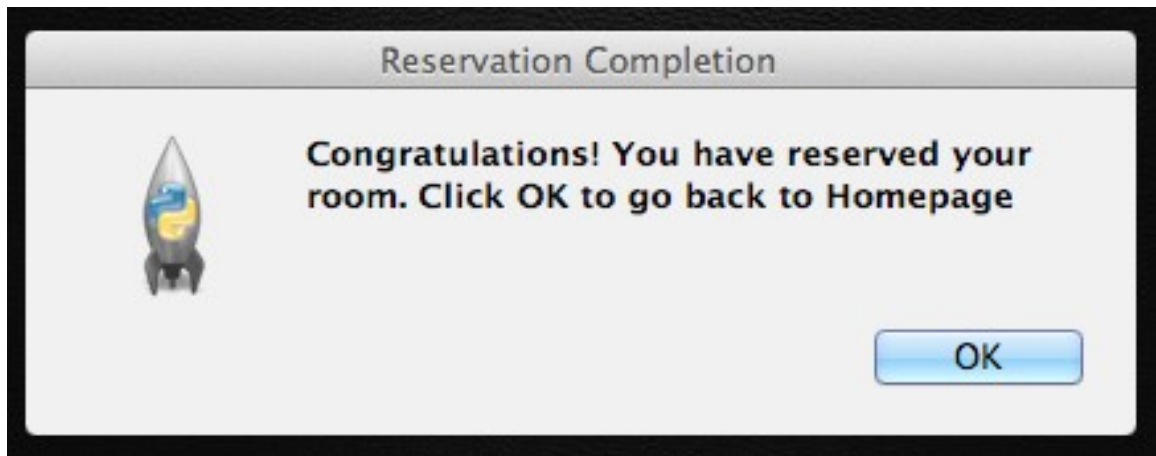
- The headers are all label with raised relief
- The 'Submit Reservation' button calls the makeReservation function.
- The 'Cancel' button closes the Available Rooms window and returns the user to the Homepage. You may find it useful to utilize a helper function to carry out this functionality.

Method Name: makeReservation

This method will insert into the RoomReservations Table the most recent reservation made by the user, along with updating the NumberOfReservations attribute in ReservationUser Table for the current user.

First, however, it would ensure selection of at least one radio button before proceeding. If the user hasn't selected any, you should display an error message asking the user to either select one radiobutton or hit cancel.

If the user successfully reserved the room, you should display the following messagebox and close the Available Rooms window and return the user to Homepage.

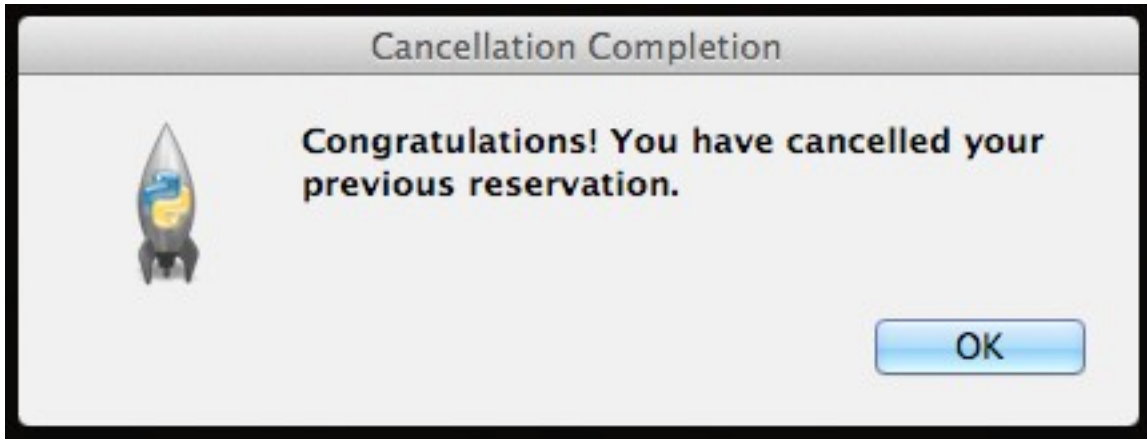


Method Name: cancelReservation

The 'Cancel All Reservations' button calls this method. This function will delete all the reservations from RoomReservations Table made by the user. Additionally, it will update NumberOfReservations attribute in ReservationUser Table to 0. If, however, user currently had 0 reservations, you should display an error message notifying them the same.

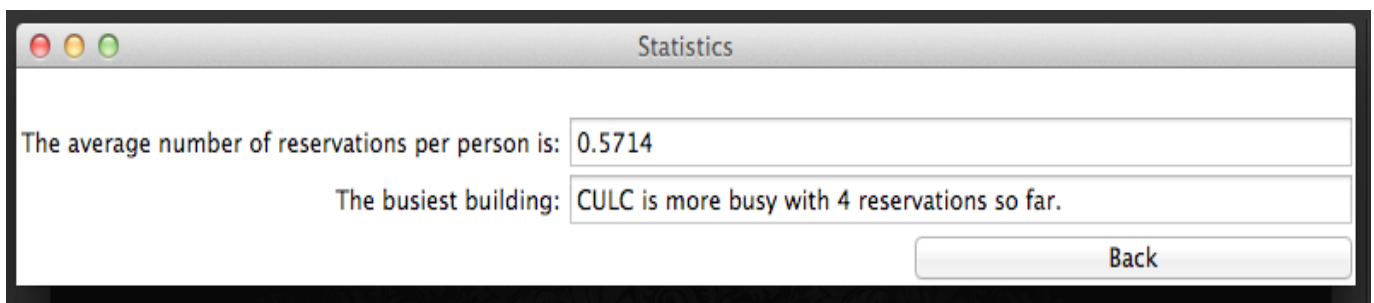
If cancellation was successful, you should display the following messagebox and return the user to Homepage, with updated current

Reservations entry boxes.



Method Name: stats

This method will hide the HomePage window, and create a new GUI window for viewing the statistics page. The GUI should look something like this:



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:

- The title of this window is 'Statistics'
- There are two entry boxes (both set to readonly) next to their respective labels.
- The first entry box displays the average number of reservations per person. This is calculated by dividing the total number of reservations made in the system by the total number of students registered. (HINT: the SQL AVG() function may be of use here)
- The second entry box displays the busiest building, the building with most number of reservations in the system. If both have equal number of reservations, you should insert the following in the entrybox:
'Both are busy with xx[number of reservations] reservations so far'
- The 'Back' button exits the Statistics window and returns the user to Homepage. A helper function may be useful.

Every time the user closes this page and re-opens after making new reservations or cancelling previous reservations, new values should be calculated and displayed.

Grading:

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

LoginPage GUI		5
Login GUI works (as specified in previous HW)	10	
RegisterPage GUI		5
Register Page works (as specified in previous HW)	10	
Homepage GUI		30
GUI correctly displays current reservations every time	10	
GUI has all components with proper characteristics	5	
None of the Radio Buttons are auto selected	5	
Available Options button checks for all possible errors	8	
Logout Button functions appropriately	2	
availableReservations()		25
GUI has all components with proper characteristics	5	
Successfully extracts correct data from the database	15	
Cancel brings user back to home page	2	
Submit Reservation button displays error if no radio button is selected	3	
makeReservation()		13
Successfully updates RoomReservations Table	5	
Successfully updates ReservationUser Table	5	
Upon successful reservation, messagebox should return the user to updated Homepage	3	

cancelReservation()

7

Successfully updates the database

5

Displays error if user currently has 0 reservations

2

stats()

15

GUI has all components with proper characteristics

3

Successfully updates the Statistics Page every time

4

Calculates and displays correct statistics

8