CS 1803 Timed Lab 3 (30 Points)

MySQL-Powered Chat System

Objective: For this timed lab, you are responsible for implementing a very simple chat program that is powered by a MySQL database. Your program will be capable of displaying the three latest messages in the chat server as well as adding new messages to the database, which can then appear on everyone else's client.

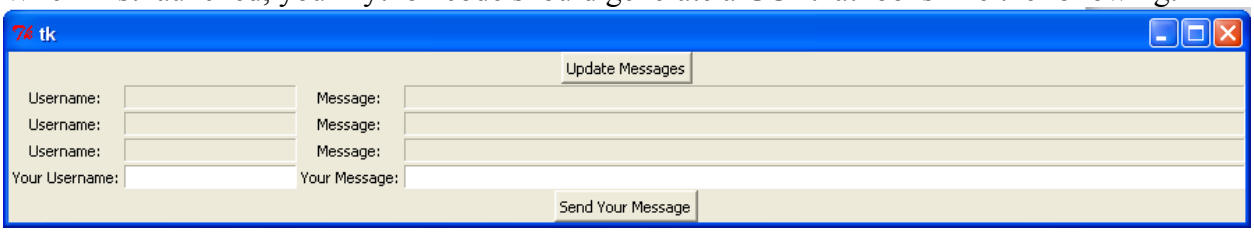
All of your code (except for the code to start the GUI) must be contained within a class named CSChatClient.

Rules for the Timed Lab (READ THIS CAREFULLY): This is a timed lab; you may use any class resources (including your previous homeworks) as well as the Internet; however, if you use code from the Internet, you must cite it in your collaboration statement. You may not communicate with people using the Internet while working on this timed lab with the one exception noted in the next paragraph:

The rules on communication with other students/people have been changed for this timed lab only. Because this is a timed lab where you are implementing a chat program, you may communicate with *other students using* ***only the Python code (chat tool) you write, or by executing SQL statements in the phpMyAdmin website.*** You ***may not*** share code, but you may assist others with the program logic. When sending messages in the chat program or by using SQL statements, you ***must use your full first and last name as your username.*** Obviously, you will have to have at least the correct SQL statements correct before you can communicate with others in the class.

The GUI

When first launched, your Python code should generate a GUI that looks like the following:



This GUI is made up of a gridded layout with 6 rows and 4 columns.

The first row is a button that says Update Messages. It is the default size and spans 4 columns. It will call the updateClicked function when it is clicked.

The second through fourth rows will display the most recent three chat messages in the database. The second row will contain the most recent message, the third row will contain the second most

recent message, and the fourth row will contain the third most recent message. The first column in these three rows will contain a label that has "Username:" as the text. The second column in these three rows will contain an Entry box that has a width of 20 and is read only; this entry box will contain the username of the sender of the message. The third column in these three rows will contain a label that has "Message:" as the text. The fourth and final column in these three rows will contain an Entry box that has a width of 100 and is also read only; this is where the message text will be displayed.

The fifth row is very similar to the second through third rows, but differs in that this row is where the user will type messages that he or she wants to put into the database. The first column is a label that has "Your Username:" as the text. The second column is an Entry box that has a width of 20 and is normal; this is where the user will put his or her username (the one that will be put in the database along with the message). ***When working on the timed lab, use your real first and last name as your username (e.g. Jay Summet)***. The third column is a label with "Your Message:" as the text. The fourth and final column is an Entry box that has a width of 100 and is normal; this is where the user will type his or her message that he or she wishes to insert into the database.

The sixth and final row is similar to the first row: it contains a single Button which contains the text "Send Your Message" and calls the sendMessageClicked method when pressed. The button is the default size and spans all four columns.

The chat table in MySQL:

The chat table contains three fields: ***id*** which is an integer and an auto increment field that serves as the primary key, ***user*** which is a text field that contains the username of the sender of the message, and ***message*** which is a text field that contains the chat message. The ***id*** field is also useful to determine which three messages are the latest, as the last three messages will have the highest three ***id*** numbers.

You may assume that there will always be at least three messages in the table (although there may certainly be more) and that the ***id*** field in the database will never be NULL. We have set the user and message fields so that they may not be NULL, but they may contain empty text strings. The first three messages in the database will be loaded by us, but all of the messages after that will be set up by your fellow students as they complete the assignment.

Obviously, your program must connect to the database to access this table.

updateClicked method:

Your class must contain a method named updateClicked which will query the database and place the latest three messages into the entry boxes located on rows two through four. Make sure that the most recent message is on row two, the second most recent message is on row three, and the third most recent message is on row four.

Hint: You should use MySQL to sort the results of your query to retrieve the messages by the ***id*** column; if it is sorted in ascending order, the most recent message will be the last result. If it is sorted in descending order, the most recent message will be the first result (i.e. later messages have higher **id** values).

You may NOT assume that all id number are contiguous, so the last three ID numbers could be: 438, 440, and 441.

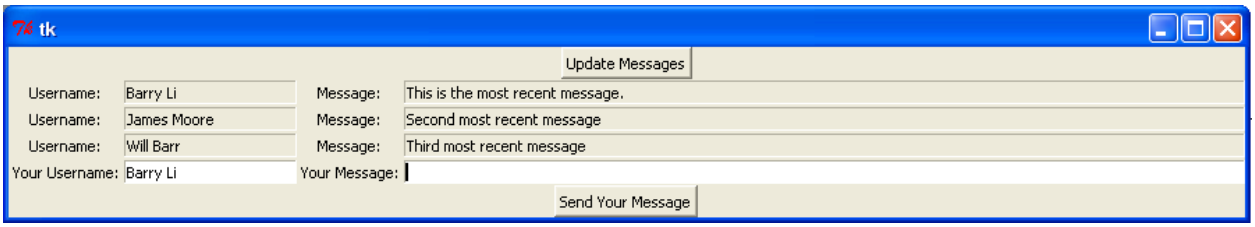
sendMessageClicked method:

Your class must contain a method named sendMessageClicked that will take the contents of the fifth row (the user's username and message) and insert it into the database. **In your query, you should ONLY be inserting values into the user and message fields; the database will automatically fill in a value for the id field. Do NOT attempt to insert any values into the id field.**

If either of the entry boxes on the fifth row is empty, you should not attempt to insert anything into the database.

After executing the query to insert the message into the database, you should erase the contents of the message entry box on the fifth row and then call the updateClicked method. (This will update your GUI so that the message you just inserted in the database appears as the last message...)

The chat client looks like the following after sending a message:



Pressing Update Messages should not modify the values in the entry box next to "Your Username" or in the entry box next to "Your Message".

Grading Rubric:

The GUI	10 points
Both buttons are correctly implemented	2 points
Entry boxes correctly configured (width, state, etc)	5 points
Messages update when Send Message clicked	1 point
GUI correctly laid out	2 points
The SQL portion/clicked methods	20 points
Successfully connects to the database	2 points
Cursors/database connection closed after use	2 points
Correct query to insert message into database	5 points
Correctly inserts item into the database	2 points
Correct query to retrieve messages from database	5 points
Displays 3 most recent messages in correct order	4 points