

Homework 8

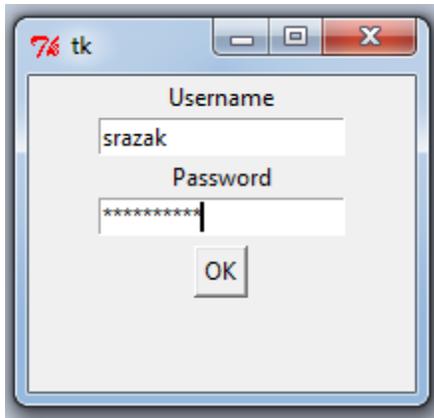
Object Oriented and Event Driven Programming

Due Thursday, November 7, 2019 at 10:00 PM

In this homework, you will build upon the code you wrote for assignment 7 where you created a text based chat client. In this homework, you will create a Graphical User Interface for the same protocols. Submit your final code to autolab by the due date – this assignment will not be autograded.

Task 1: [10 points] The Login Screen:

Create a dialog window that asks for the users' login name and password. This should be the first window that pops up. If the user enters the correct username/password combination, then allow him to go to the next window, if not, your program should exit. The entry box for password should show * for each character typed and should NOT display the password as shown in the window below. Once the user has logged in successfully, this window should close.



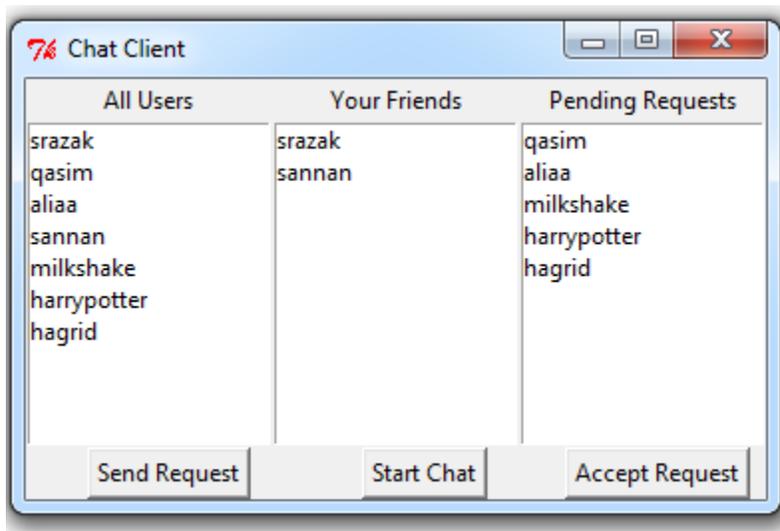
Task 2: [20 points] Main Screen:

In this task, you will create the main chat window. This window should have three lists with associated buttons as explained below:

1. The first list should have all users on the server. The user should be able to select a name and then click on the button titled "Send Request" to send a friend request to this user. When the friend request is sent, show a dialog that says the friend request was successfully sent.

2. The second list should have names of all the friends of the currently logged in user. The user should be able to select a name from the friends list and start a chat session by clicking the Start Chat button. The details of starting a chat will be given in the next task. As other users accept your friend requests, your friends' lists should be updated in real time.
3. The third list should have names of friends who have sent a friend request to this user. The user should be able to accept friend requests by selecting a name from the list and clicking the "Accept Request" button. If the request is sent successfully, update the friend list accordingly. This list should also be updated in real-time from the server.

An example of this screen is given below:



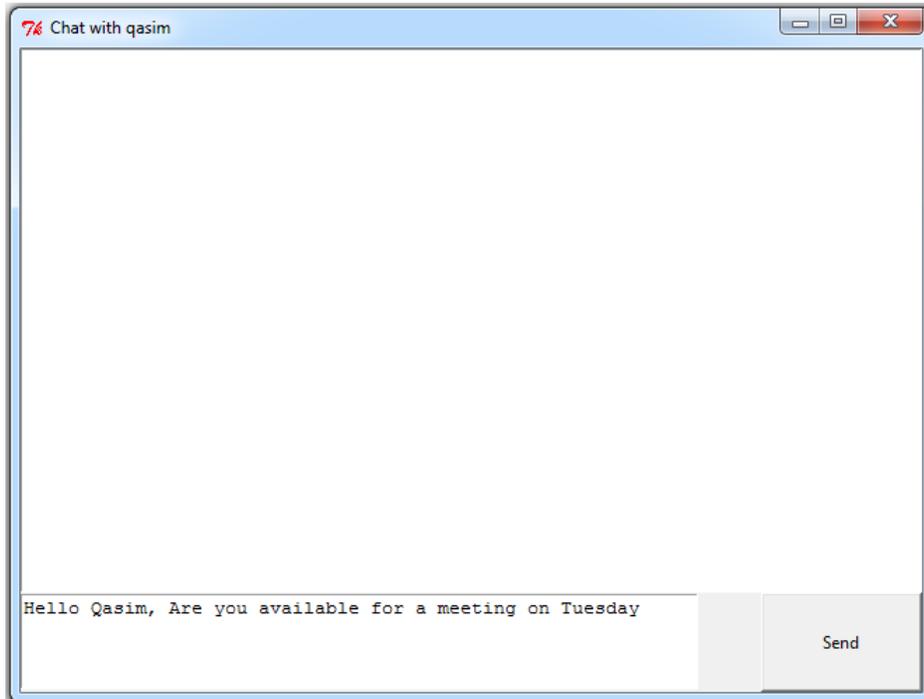
Task 3: [20 points] The Chat Screen:

In this task, you will create a chat screen where the user should be able to chat with a friend. The window will be started in two ways:

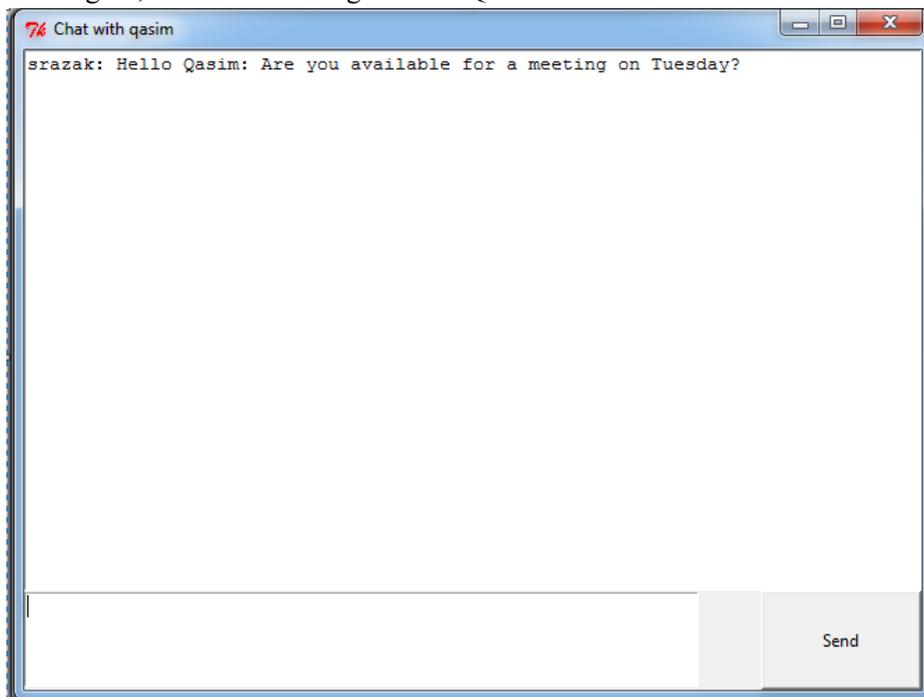
1. User selects a user and clicks on Start Chat button. This should start up a chat window with three main widgets:
 - a. A text box to show all the conversation
 - b. A text box for the user to enter text to be sent
 - c. A send button
2. If the user clicks on a friend with whom the user already has a chat window open, your program should not start a new chat window.
3. A remote user starts a chat session with this user. This means, that your program should periodically check for messages from users (a frequency of once every 5-10 seconds should be reasonable). If a message is received from a user that does not have an active

widow from the previous option (chat window opened according to previous item), this should cause a new window to be opened for this chat session.

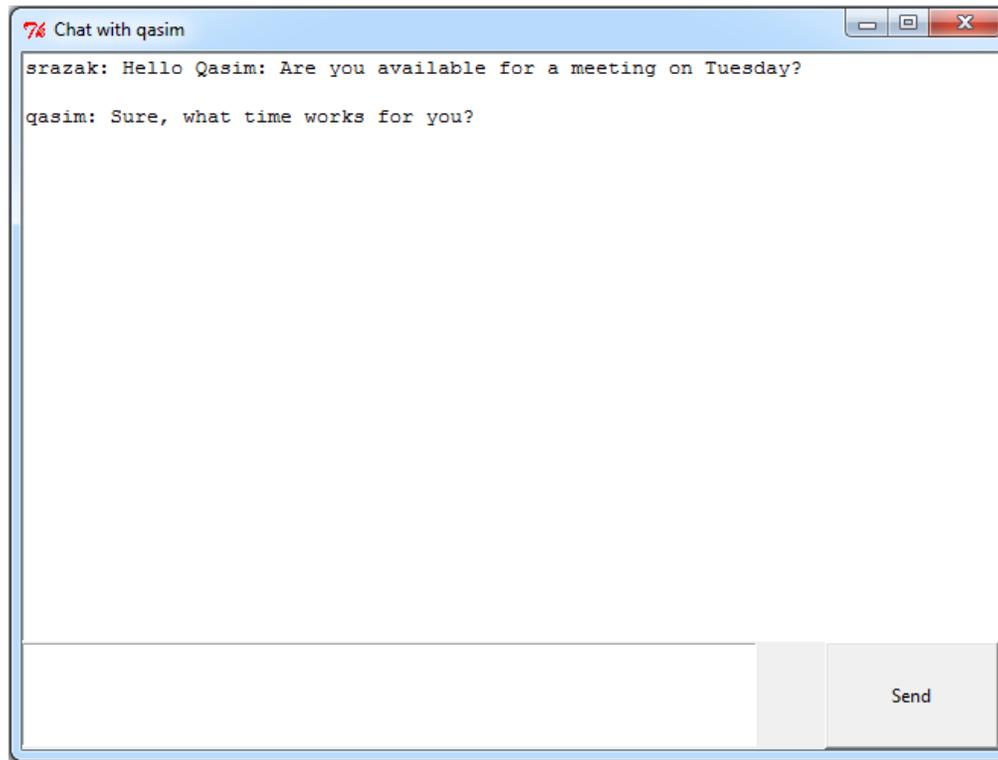
The following figures show an example chat session. The first image is after the user selects “Qasim” from friend list and clicks Start Chat.



The second figure, shows the message sent to Qasim and shown in the conversation window.



The following figure shows the message received from Qasim.



Task 4: [10 points] Object Oriented Design:

Your program should take advantage of Object Oriented Programming to simplify program design. The last 8 points will be allocated for definition and use of appropriate classes and objects of those classes.