Arizona State University
School of Mathematical and Natural Sciences
ACO 432: Distributed Systems (Fall 2019)

Prof: Yasin N. Silva

Project # 1: Clickers@ACO

Due date: **Tuesday October 29th**, 2019 before class starts.

This is a group project (two students per group). Weight: 20% of class grade.

Objective: Experience the lifecycle of software development through design and implementation of a classroom response system (CRS or clicker system).

Description: A Client/Server based classroom response system is a tool that allows an instructor to send a question to all participant students and to collect their answers. The system is composed of multiple CRS clients (for students and the instructor) and a CRS server. The CRS client recognizes two types of users: instructor and student. The systems allows both types of uses to register and login. Additionally, the instructor can send a question to all online students. The system collects all the responses and shows the results to the instructor. Online students receive all the questions submitted by the instructor and can submit their answers. The CRS client program also displays the list of online/offline users in the system (and their types). The CRS server manages the users and forwards messages to online users.

More specifically, your client and server programs need to support the following functionalities:

- **User registration**. A user has to register into the system before using it. The Server records the information of all users (username, password and type) in a file. The system should allow the registration of students and only one instructor.
- **User login**. A user has to login to use the classroom response system. A user is online after he/she logs in. The server remembers the online/offline status of each user.
- **Sending questions.** Only the instructor should be able to send questions. Each submitted question is sent to all online students. The system should support two types of questions:
  1. **Multiple choice question.** This question contains the question itself (e.g., "What is the fastest sorting algorithm?") and three possible answers (e.g., [1] Bubble Sort, [2] Quick Sort, [3] Insertion Sort)
  2. **Open ended question.** This question only contains the question itself (e.g., "What is the fastest sorting algorithm?")
- **Answering questions**. All the online students should receive each submitted question and should be able to answer appropriately (Select one of the possible answers in the case of a multiple choice question or write a sentence to answer an open ended question). Students only have about 10 seconds to submit their answers.
- **Collecting and displaying answers.** The instructor has to wait for the answer for about 10 seconds and then the system should display all the received answers. The system should aggregate the answers in the case of multiple choice questions. Answers are only presented to the instructor.
- **Refresh online/offline user list**. After a user logs in, it retrieves the list of registered users (username, type and online/offline status). The user should be able to tell whether a user is online or offline. Your code should retrieve and refresh the list information regularly (every 30 seconds) so the user can see the updated information.
- **Log off.** When a user logs off, his status turns into offline. He needs to login again to use the system.

You can choose to design and implement the message passing in the chat system using either Java RMI or network socket programming.

Deliverables (Canvas):

1. Design documentation including identified classes, their instance fields and methods. Sent by email in a single PDF file and hard copy.

2. Source code.

Upload a single zip file to Canvas containing the PDF file and your code.

Demo: Each team will briefly demonstrate their system (in class or during office hours).

Grading:

Design document	18 points
User interface	16 points
User registration and login/logoff	8 points
Refresh online/offline users info	16 points
Send/receive questions/answers	32 points
Demo	10 points
Total	**100** points