**CIS 422 Project 1:**

**Classroom Cold-Call Assist Software**

**SRS**

Patrick Thomasma, David Han, Kassandra Morando, Brianna Vago, and Geli Zhang

January 10,2022 – v1.0

**Table of Contents**

**1. SRS Revision History**

**2. Specific Requirements**

2.1 Problem Statement

2.2 User Description

2.3 Use Cases

**3. Description of Requirements**

3.1 Functional Requirements

3.2 Non-Functional Requirements

**4. Acknowledgements**

**1. SRS Revision History**

**Date Author Description ————————————————————————————————————**

**1-6-2022 dh Created the initial document.**

**1-9-2022 dh Wrote the SRS rough draft.**

**1-9-2022 bv Began proofreading rough draft.**

**1-11-2019 bv Finished editing rough draft.**

**2. Specific Requirements**

**2.1 Problem Statement**

The problem at hand is the struggle to maintain consistent, student-wide participation in a class environment. Students that ask questions and express interest reflect their engagement in the class and facilitate further learning opportunities for their classmates. High counts of student participation are ideal as a result. However, participation during class can also hinder the learning environment in large quantities. Student interjection in excess disrupts the flow of a class, which frequently breaks attention away from the professor and their lecture. Likewise, classes where the majority of student participation only roots from a small percentage of the class discourages class-wide engagement of the material and potentially turns away the participation of the remaining students in the interest of time.

One solution to the problem is the practice of “cold calling”. When a professor cold calls students in their class, the selection of students is random and disregards the absence of a raised hand. Students selected through a cold call are provided with an advanced notice, and after a short amount of time they are asked to provide input to the class discussion by sharing any questions or concerns at the current point of the lecture. Cold calling encourages attentiveness due to the sudden, involuntary nature of the concept requiring the students to review class material ahead of time and come prepared to learn. Furthermore, professors that use cold calling have greater control over the course through the ability to control the quantity and frequency of student input during their lecture.

Although cold calling solves the problem of maintaining frequent participation from each student in the class, the practice alone holds its own problem of inefficiency when setting up. The random selection of students to be cold-called requires the professor to take time away from the lecture to manually choose random student names, introducing an additional form of disrupting the flow of a lecture and breaking the engagement of the students. Professors that use cold calling are also still required to manually track the participation count. Both problems of cold calling can be fixed by automating the process through the use of a user-friendly program.

**2.2 User Description**

The target user for the cold call program will be class instructors. Class instructors can use the program in their lectures in order to increase attentivity and encourage students to ask questions and participate in class discussion, while saving the instructor the time of manually organizing the students they would like to cold call. Instructors can also use the program to track the frequency of participation for each individual student in class throughout the duration of the course, and the flagging of student names can be done during class in order to refer back to at a later time or date. The instructor will be allowed to dynamically update the roster of their class in the cold call system.

**2.3 Use Cases**

**Use Case: Using cold calling during class**

*Description:*

A professor would like to call on his students to contribute to the class lecture/discussion randomly but equally throughout the duration of the course.

*Steps to Complete the task:*

1. The instructor started teaching the contents of class for the day. They stopped the class to ask for questions and waited 10 seconds to see if anyone would volunteer.
2. The instructor opened the cold calling system on their macbook and launched the system.
3. A list of four names students’ appear on the top of the screen while the instructor goes to the next topic in the presentation.
4. The instructor stops the lecture again and asks for the students to ask a question or share a thought about the topic. Nobody responded so they chose a student from the cold call list.
5. No student responded, so the professor called on the third person on the cold call list to say something about the class.
6. After the student responds, the instructor replies to the question and removes the student’s name from the list:
   1. Press the left and right arrow keys to highlight the student’s name in the list.
   2. Press a down arrow key to “drop” the name from the visible list without a flag or an up arrow key to remove the name but also “raise” a flag for that student
7. The fourth student on the list moves to the third position and a new student’s name fills the fourth spot on the list.

**Use Case: Using cold calling in a professional work meeting**

*Description:*

Roles need to be assigned randomly prior to the meeting, and the roles must be taken up by group members in an equally distributed manner over time.

*Steps to Complete the task:*

1. The host of the meeting collects a list of participants in the meeting and what roles need to be filled.
2. The host uploads a file with a roster of the members attending the meeting to the system and runs the system.
3. The first four people will be displayed on the screen. The host will assign the first person to the first role then drop the first person from the list. Everyone in the list will move forward one position and the fourth position will get filled by a new person.
4. The host will repeat step 3 until all of the roles are filled and then closes the system.

**Use Case: Using the system at the end of the term**

*Description:*

A professor needs to record the track of participation of each individual student throughout the term.

*Steps to Complete the task:*

1. The professor opens the summary performance file and reviews it.
2. The summary performance file has a list of students with their performance data. Formatted by: <total times called> <number of flags> <first name> <last name> <UO ID> <email address> <phonetic spelling> <reveal code><list of dates> with tabs between each data. The list also includes the days the student was called and are in chronological order.
3. The professor imports the data in an excel file and uses the first two cells for the participation data. They analyze the average of total times called and the number of flags to see the performance of the students.

**3. Description of Requirements**

**3.1 Functional Requirements**

* The system must be able to accommodate any size group of participants.
* The code must call on people randomly.
* Every person must be called on by the system once before repeating people.
* The system should be able to allow participants to be added or removed from the group.
* The name of the participant being called on should be displayed.
* The list of participants “on deck” should be displayed at all times.
* The system should display the list above all other things on display.
* The professor should be able to flag the students they want to follow up with.
* This system should use a roster file with the name, ID, email, and possibly photo of participants
* The data should be able to be imported from the roster file to the system.
* The system should be able to run on Macintosh OSX 10.13 or 10.14.
* All documents must be in PDF format.
* Python should be used to code the system.
* The system should use Python tkinter for GUI.

**3.2 Non-Functional Requirements**

* This system should get people who would not normally participate to speak in class.
* This can help check to see if people have a firm grasp on the material being taught.
* Can help a professor make sure that all students are learning and engaged in the material.
* Can help a professor figure out how effective a question is in covering the material.
* The user interface should be easy to navigate and easy to use.
* System has to be able to be easily maintained for long term use.

**4. Acknowledgements**

The formatting of this document was based on a Software Requirement Specification template provided by Professor Anthony Hornof.