Project Name:

Present Quiz

Team Members:

Charles Ritchie, Nick Norton, & Christopher Cox

Client Name and Organization:

VT Professor TBD

Platforms and Modules:

Android phone using the following modules:

1. GUI
2. Lifecycle
3. Connectivity
4. Sensors
5. Graphics

Overview:

The app will be used as a free replacement for an i>clicker. This addresses the issue of i>clickers costing a student money and this is a free alternative. Easy to implement with a simple download from the student while the professor receives an email with the student's answers. The app will be able to track the user's GPS location, scan a supplied QR code, and answer a question(s) from the professor. The purpose of the QR code is to keep students from spoofing their GPS location to simulate them being in class. This is just one of the checks that can be implemented with our application to prove a student is present and in class while also being a free alternative to an i>clicker.

Description of leveraging platforms:

1. GUI - Without a GUI the student would not know what data to enter or how to enter it. The GUI is a major necessity for this project.
2. Lifecycle - The app will make us of the android lifecycle module by allowing save states, and having multiple screen for verify that the student is actually in class.
3. Connectivity - The student would send their answers via the internet to an email account the professor can reach that logs the students answers.
4. Sensors - Use the phone GPS location to check if the student in the in classroom.
5. Graphics - Use the camera to check scan a QR code to provide another level of security that the student is actually in class. Also may do some fancy graphical animations.

Team members: Christopher Cox, Charles Ritchie, Matthew Szostak, & Nick Norton
Due Date: 3/3/17
Assignment: Project Phase 2

Robust Definition

Our mobile phone application will be used as a free replacement for an i>clicker. This addresses the issue of i>clickers costing a student money that is not included in tuition. Our application would also limit the number of students that "fake" being in class by having their friends use their i>clicker. Most students will not give their mobile phones to their friends to use in class while they skip. This is one reason why our application will use both QR codes and GPS location. Just in case students think they are clever and spoof their GPS location from home.

Our application will also have more features than an i>clicker. Those features being questions that cannot be answered in multiple choice format or if the students have questions. A student can type in the questions and it will be stored with their name when they submit for the professor to view. Any answer the student submits whether it's a multiple-choice answer or a short answer it will also be viewable by the professor.

Detailed Specifications

The main scope of our project will be for universities to use our application a replacement to an i>clicker. This is not intended for high schools or lower levels of schools, universities and up. Our project's main goal will be to have a working application that can be implemented in a small-scale classroom. This means our minor goals would be the following:

1. Get professors application hosting
2. Get students application sending to host
3. Get professors application receiving from clients
4. Add more features to professor's application (different question types)
5. Add more features to student's application (different answer types)
6. Ensure both programs work fully together

Our application will not infringe on any student's privacy because of a few things. The first being the student should be in class anyways which means they can physically be seen by the professor. The second being that if the student isn't in the classroom when they try to access the application our application doesn't care about their specific location. It will only recognize the student is not within the threshold of being in the classroom and therefore will not accept the students replies.

Instead of the professor getting emails we have a proposed solution if the professor feels this would be an email overload. The professor could run a simple program designed and built by us that is ran on their computer in class. On startup, this program simply outputs an IP address and port for the student to connect to through their application. Then all the student's data can be stored in a simple text file on the professor's computer. This would limit all the students answers to a text file that is small in size and organized based on student name.

The professor will have the capability to either have multiple choice questions or short answer. This way the professor will have flexibility in the questions they ask the students. For instance, a simple roll call would only need multiple choice. However, an answer to a quote from a reading that was to be completed before class would need to be short answer.

Intended Users

The main user group for our application will be students and professors at the college level. Most high schools or lower level schools do not use i>clickers nor do they have a need for them. Our first thought consisted of keeping the intended user group to a small to medium sized class so that, the professor wouldn't be overloaded with 200+ emails from the application. However, with some slight modifications to our application design we can our application available to all class sizes. This is simply completed by having the professor run an additional program on their computer which will store all the students answers in a text file.

The reason we chose our demographic to be students and professors is because there are problems with i>clickers that sometimes outweigh their benefits. The main problem being that i>clickers can be costly. The second problem would be i>clickers take batteries and even though it is rare batteries die when it happens it's usually when you're in class answering a question. Problem number three is that students will give their friends their i>clicker to go to class and answer the questions for them. Then the next class period the students will rotate and so on. The fourth and final problem with i>clickers would be they are not always 100 percent reliable. Students will press their answer key and the professor will never receive their answer. The problem with this is the student never knows for 100 percent certain that their answer was received by the professor. There are also different acceptance frequencies, which if the student is on the wrong frequency their submission will not be recorded. Having a mobile app there would be no different frequencies to submit the answers.

Mobile Application

The idea behind this being specific to a mobile application is simple. A student is not as willing to give up their personal mobile device to even a friend to take to class. Whereas with an i>clicker this is a very frequent occurrence. With the application being on the mobile phone we can an authentication layer that is set once a semester that is tied to that student's phone and can only be edited by the professor. This is put into place to keep the student from logging in, completing the task, logging out, then logging back in as another student and completed the same task.

Another reason for using a mobile phone is because we have access to GPS and the ability to scan QR codes with the phone's camera. The professor will place a QR code on the board and the student will take a picture to log them in. Once the student takes the picture a check will be in place to see if the students GPS location matches up relatively with the classroom and then the student can log in to complete the assignment. It really should not be a threat to the student's privacy because the app will not be storing data of the student's location inside or outside the classroom, just a simple check if they are in the classroom. The pictures can technically have a GPS fingerprint on them, but the student should turn that off if they are worried about google tracking their location in that manner. We will not require that the student turn on this feature when taking the picture of the QR code as that would be possibly violating their privacy.

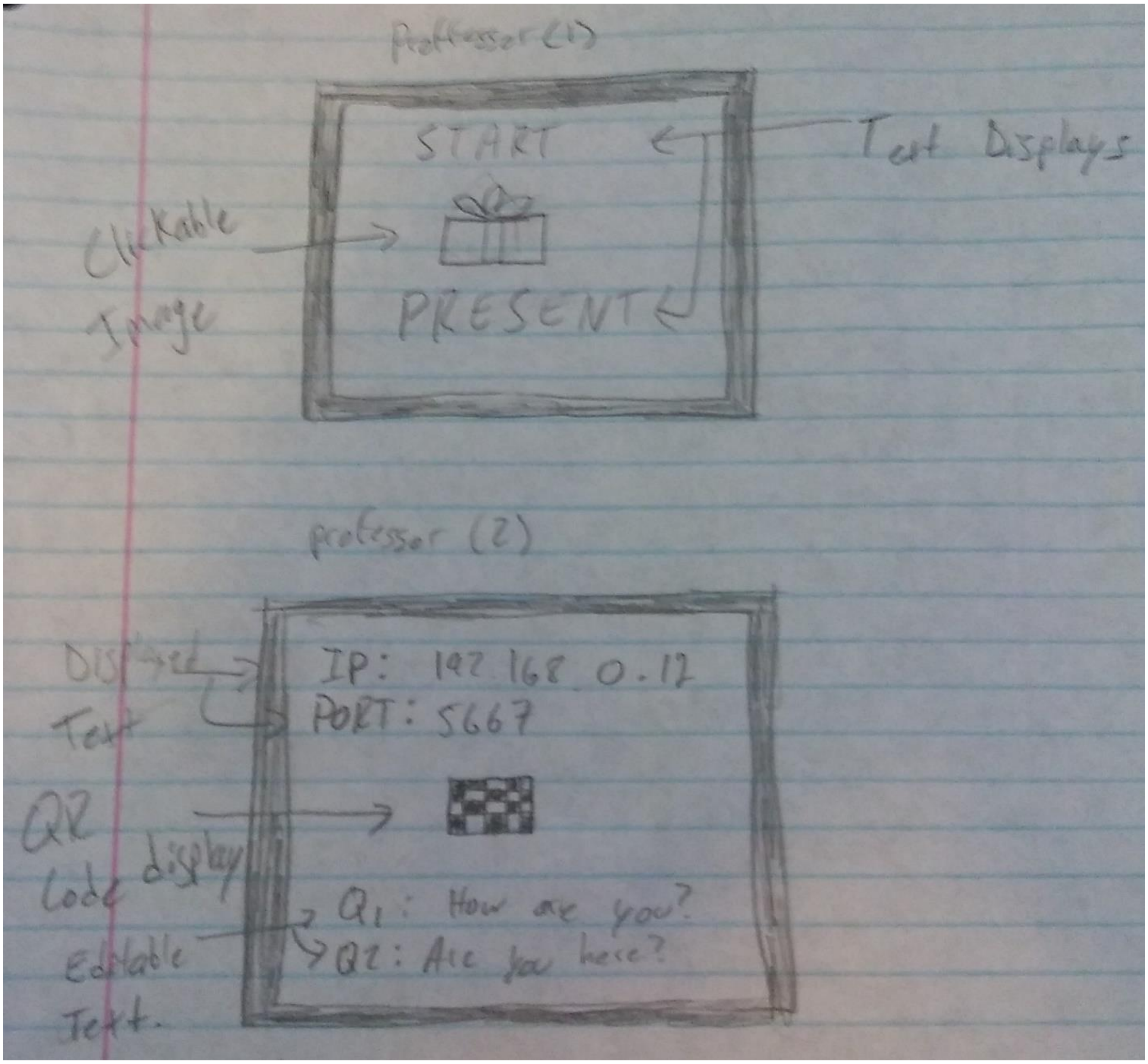
Functional Requirements

Our application can be used to do any of the following:

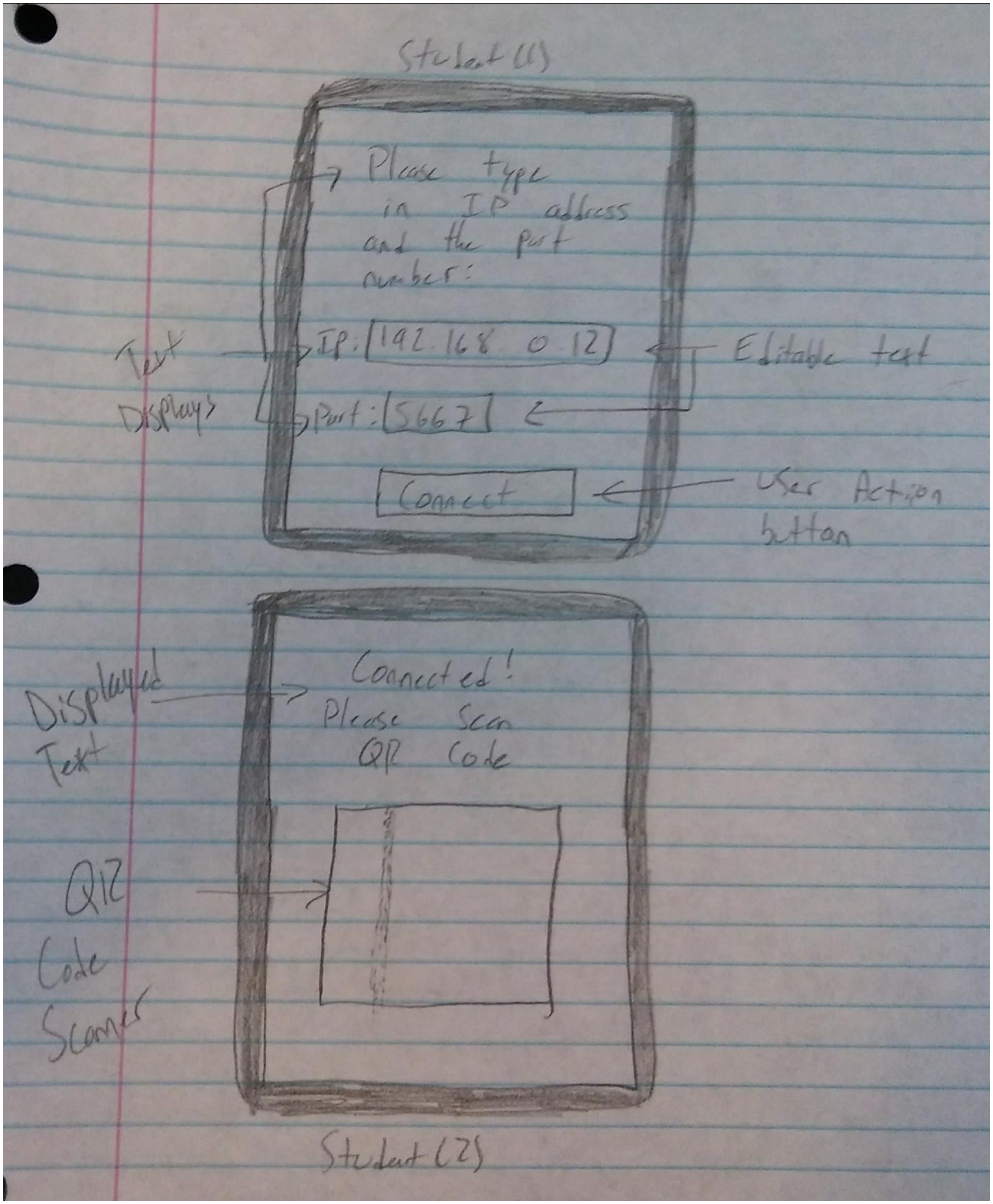
1. Attendance
2. Quizzes
3. In Class Responses
4. Questions

The first two are things an i>clicker can accomplish. However, a student cannot place in their own specific response. Nor can a student ask a question using an i>clicker. Once again all benefits to our mobile phone application.

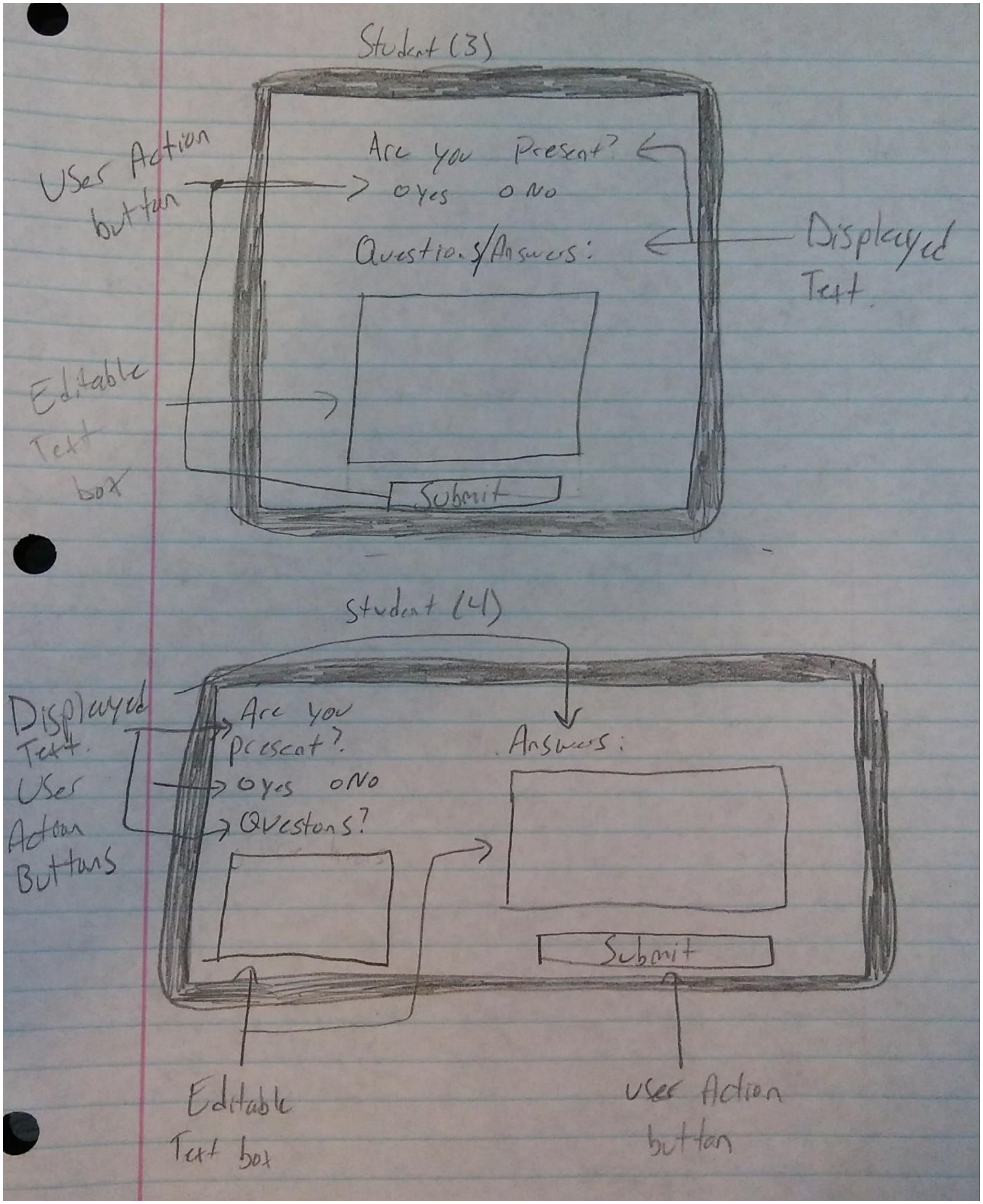
Sketches



Wireframe 1: Professor Application

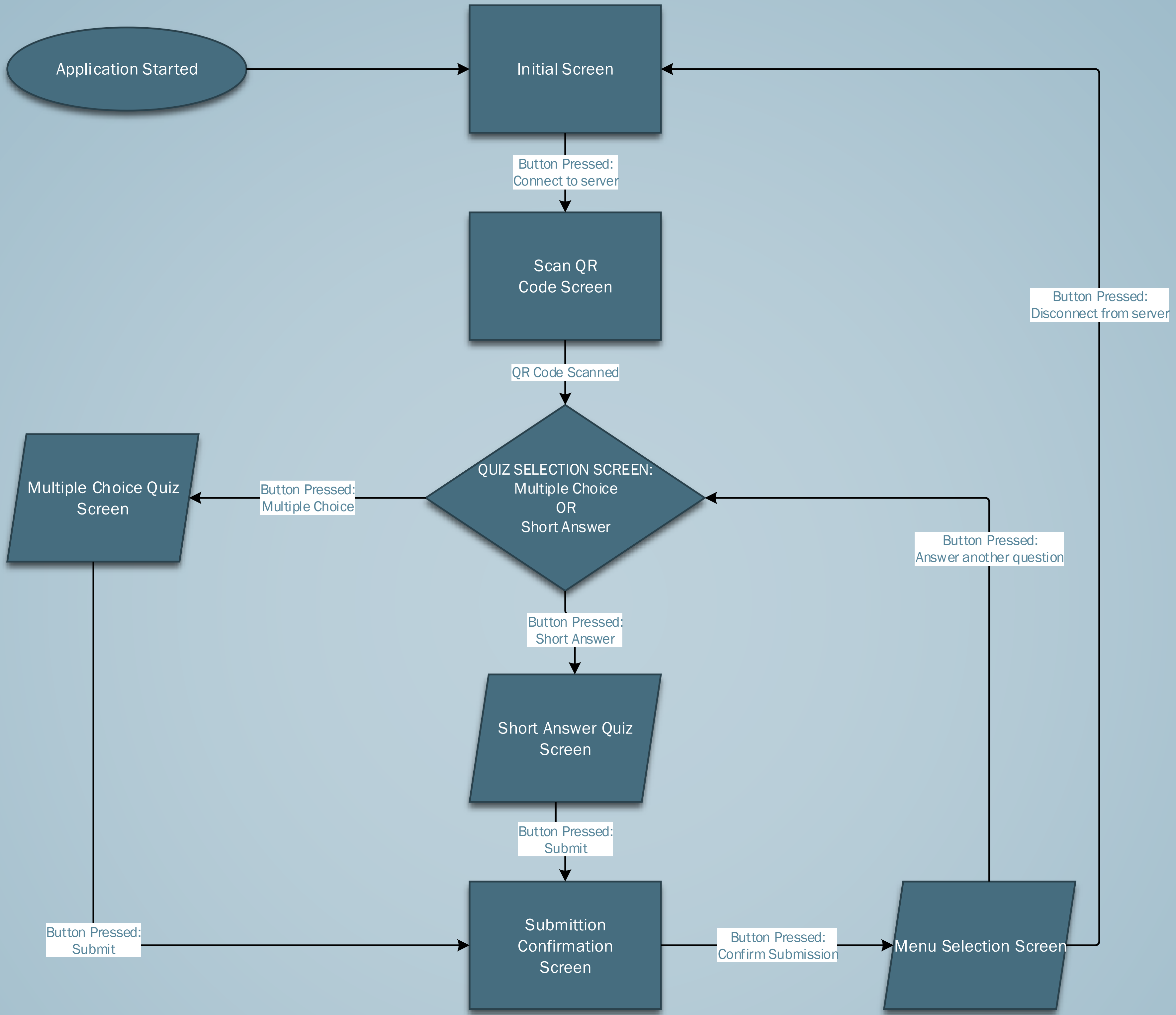


Wireframe 2: Student Application



Wireframe 3: Student Application

Application Lifecycle



Application Started

Initial Screen

Button Pressed:
Connect to server

Scan QR
Code Screen

QR Code Scanned

QUIZ SELECTION SCREEN:
Multiple Choice
OR
Short Answer

Button Pressed:
Multiple Choice

Multiple Choice Quiz
Screen

Button Pressed:
Short Answer

Short Answer Quiz
Screen

Button Pressed:
Submit

Button Pressed:
Submit

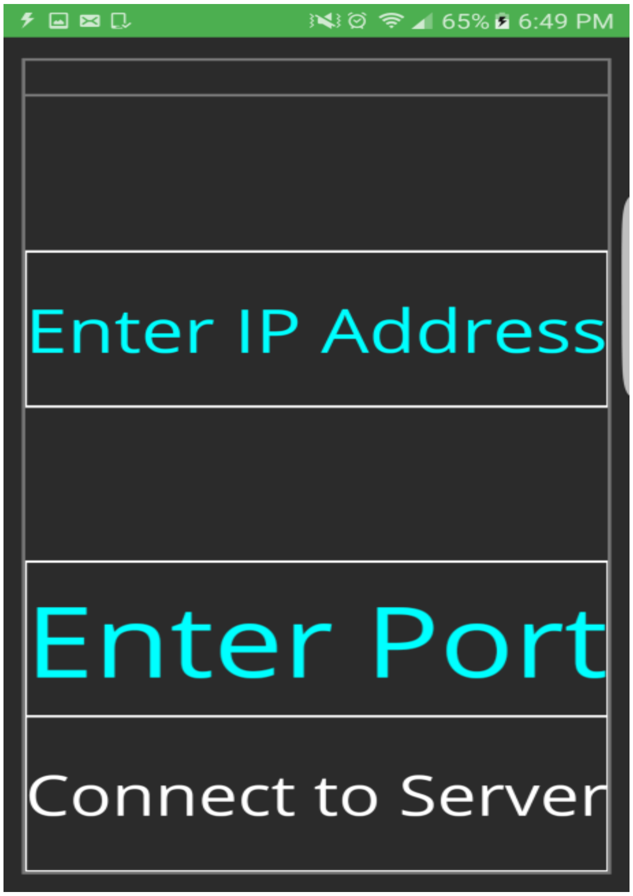
Submittion
Confirmation
Screen

Button Pressed:
Confirm Submission

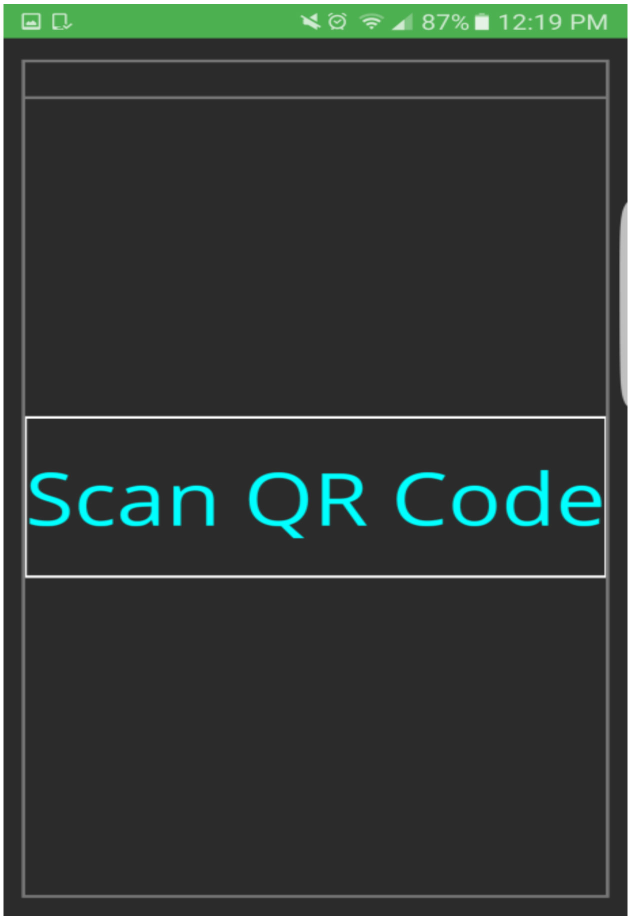
Menu Selection Screen

Button Pressed:
Answer another question

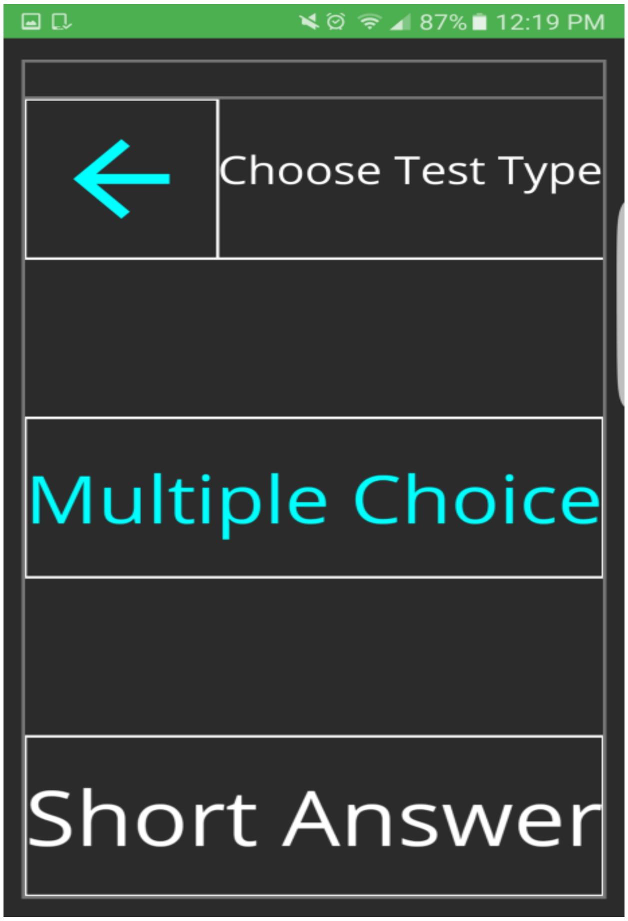
Button Pressed:
Disconnect from server



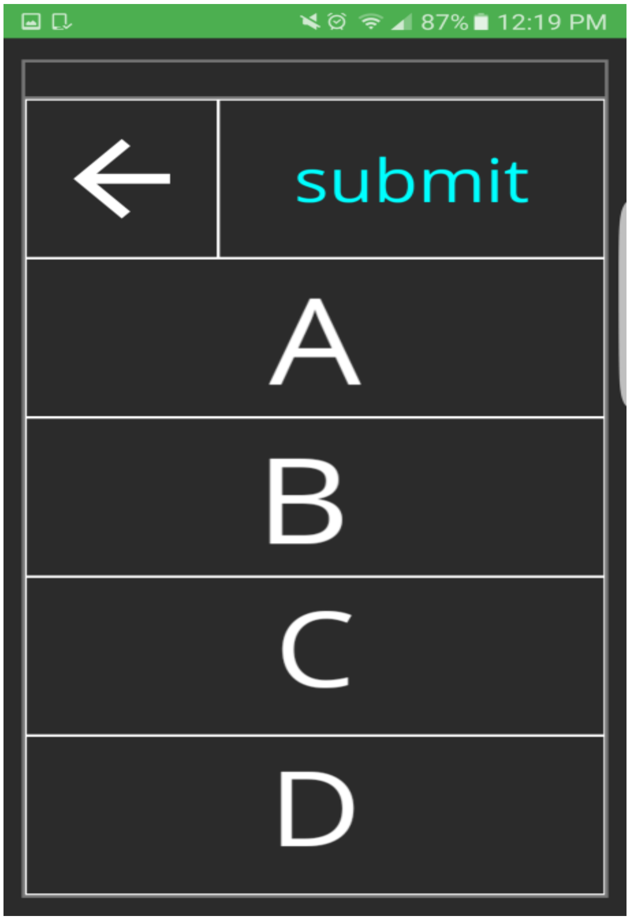
Screen 1: Initial



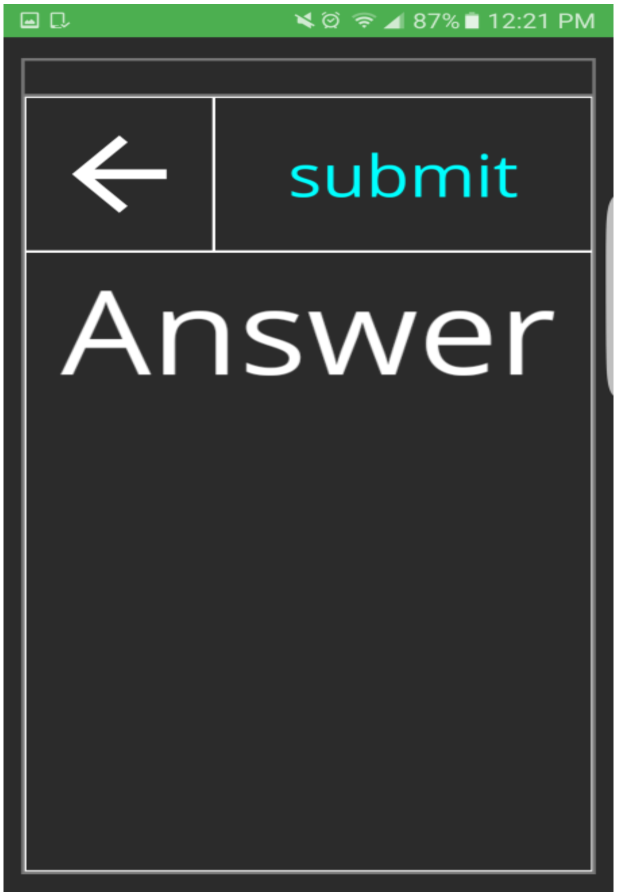
Screen 2: Scan QR Code



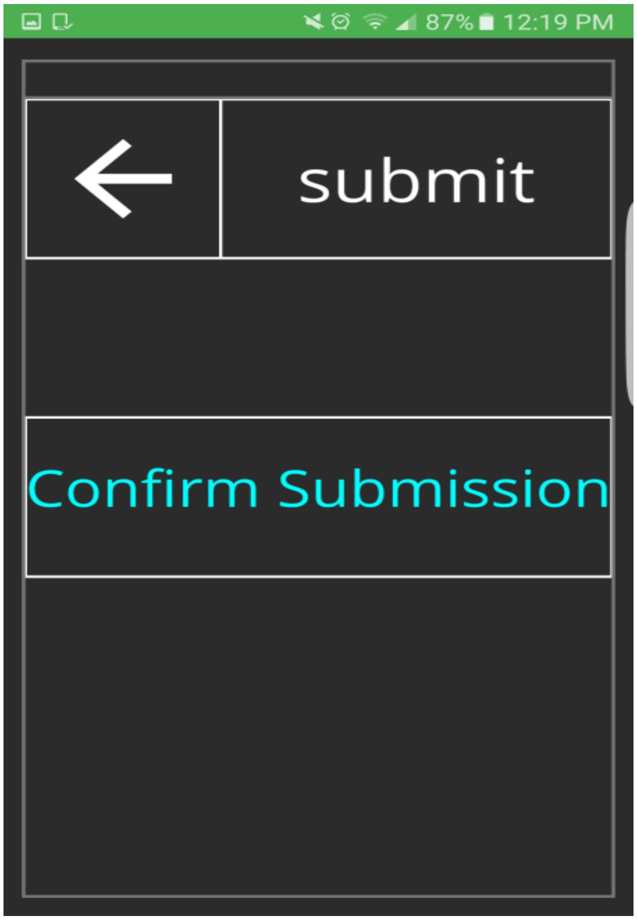
Screen 3: Quiz Selection



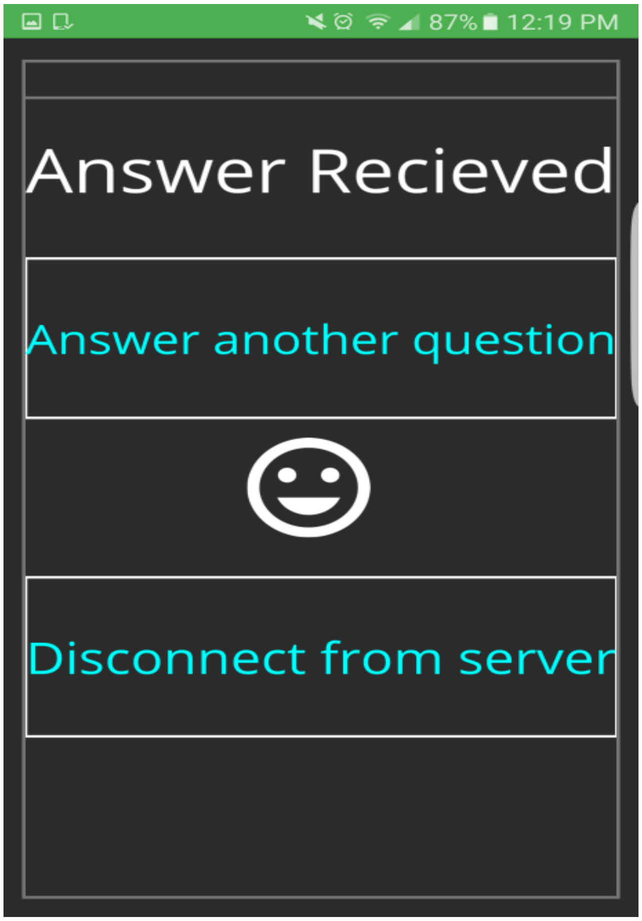
Screen 4: Multiple Choice Quiz



Screen 5: Short Answer Quiz



Screen 6: Confirmation



Answer Recieved

Screen 7: Menu Selection

Name: Chris Cox & Charles Ritchie
Due Date: 5/8/17
Assignment: Final Group Project
Team Name: Present Quiz(Clicker); Team 4
Class: CS 3714

Main Functions

The main functions of our application include:

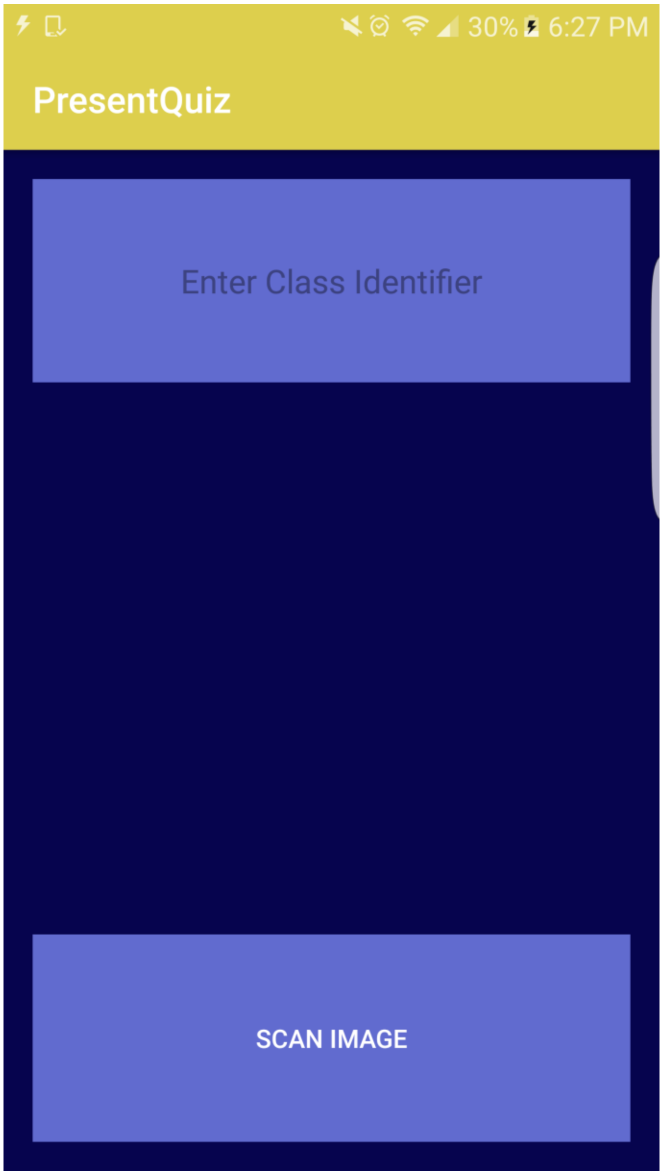
1. GUI
2. Lifecycle
3. Connectivity
4. Sensors
5. Graphics

The GUI was essential in our project because without it the user would not know how to navigate our application. The lifecycle is necessary to correctly save and restore the GUI while also not allowing the student to go back and change their answer during the quiz. Connectivity was our biggest function which housed our server/client communication. This is how the professor could receive and send the quiz to the students. Sensors and graphics were key features to our project because we used the camera to take a picture of a custom QR code graphic to decrypt and get the information to connect to the server. This is a check to make sure the student is in the room and not connecting while they are not present in class.

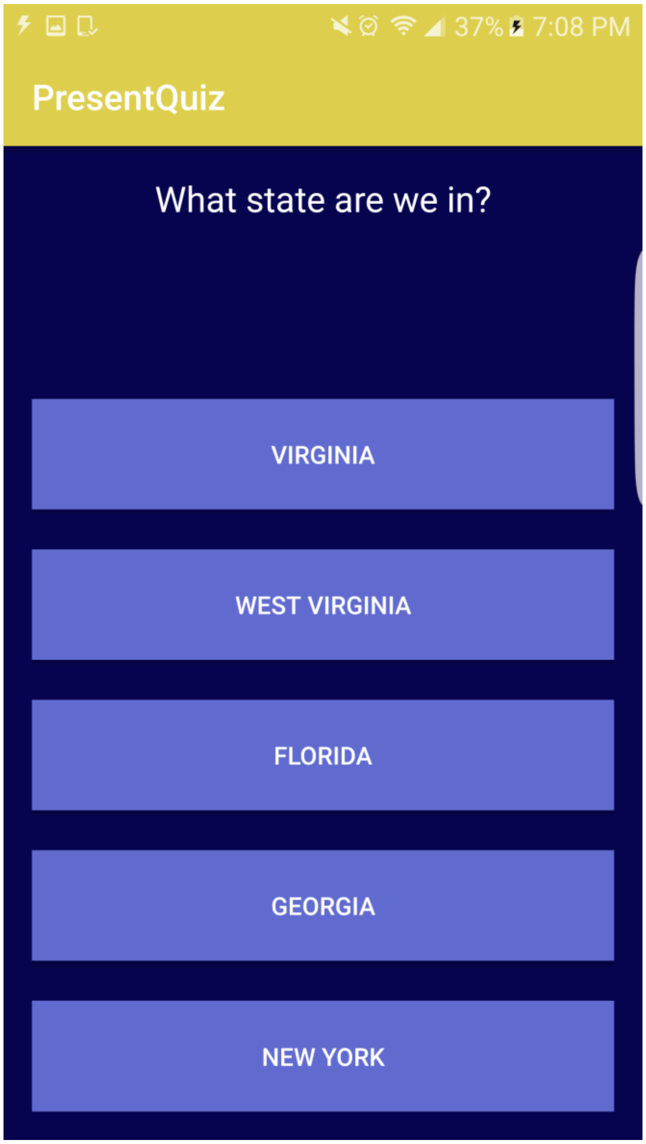
Navigation Paths

Our application path is as follows:

1. Login Screen- Student must enter class identifier and then proceed to the next screen by scan the image.
2. Scan Screen- The student must take an image of a custom QR code that is decrypted to return information containing server connection parameters.
   a. After the parameters are returned it automatically connects to the server in an AsyncTask and retrieves the first quiz question.
3. Question Screen- The question will be populated on the screen and once the user answers the question the program will check to either proceed to the next question or if all of the questions have been answered. If all the questions have been answered then another AsyncTask is started where the mobile reconnects to the server and then sends out the students' responses along with their student identifier.
4. Results Screen- Once here the results are displayed on how well the student did. The student cannot start another quiz unless the professor posts a new quiz and the user logs back in.

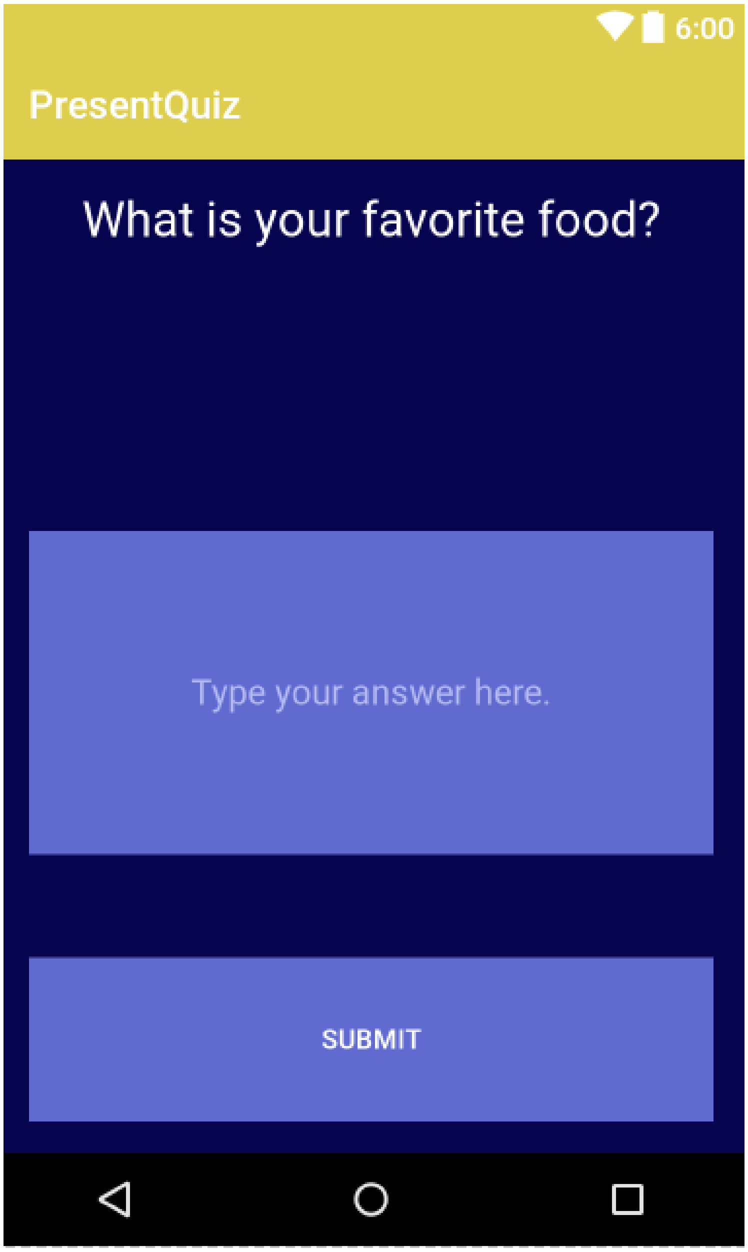


Screen Shot I: Login Screen (Main Screen)



What state are we in?

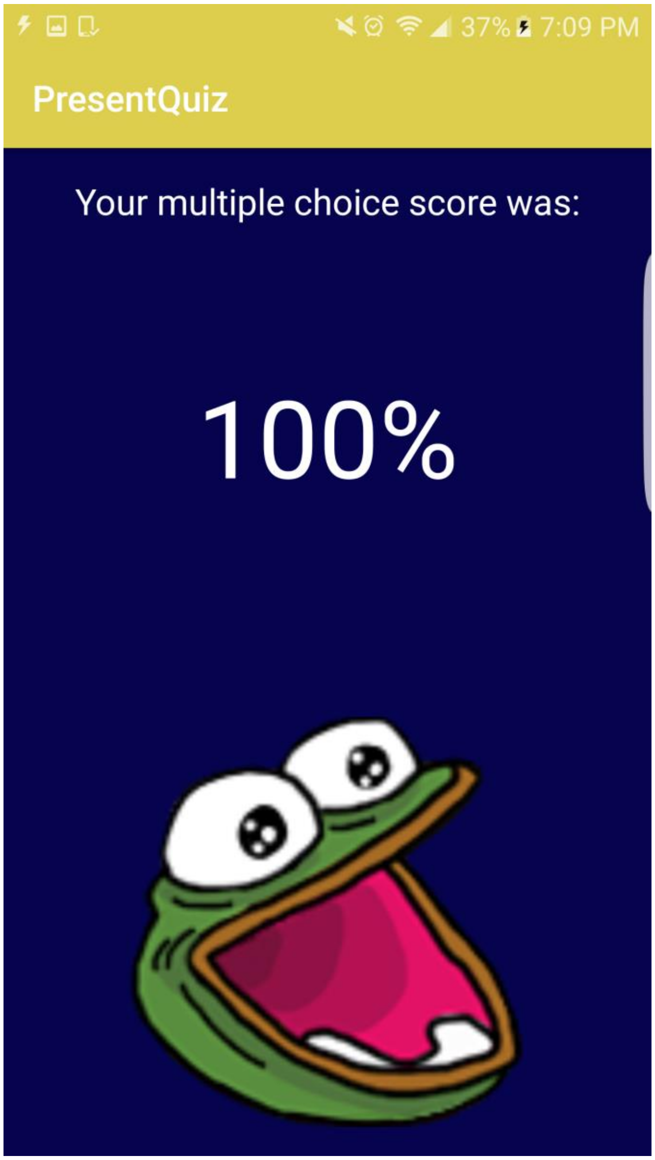
Screen Shot II: Multiple Choice Question Screen



What is your favorite food?

Type your answer here.

Screen Shot III: Short Answer Question Screen



PresentQuiz

Your multiple choice score was:

100%

Screen Shot IV: Results Screen