University of Arkansas – CSCE Department
Capstone I – Preliminary Proposal – Spring 2019

Lecture+

Joseph Gorman, Vladimir Sabado, Cristhian Espinosa, Jagmeet Wander, Brett Hustead

Abstract

Students' questions play an important role in meaningful learning and scientific inquiry. They are a potential resource for both teaching and learning science. There are many reasons students may be resistant to speaking in class. For example, introversion, shyness, English as second language, culture differences and previous bad/embarrassing experience etc. The main objective of our Lecture+ app is to enable students to ask questions anonymously during the lecture.

1.0 Problem

Most students have been in a classroom setting where they wish they hadn't asked a question or added to the discussion, maybe the answer was obvious or the question seemed unrelated - "out in left field", so to speak. It could also be that they wish they had asked a question about something they didn't understand but did not because of reasons listed below. It takes strength to raise your hand in class and ask question. Let's examine reasons why students do not ask questions in class.

Shyness: shyness can be bitterly difficult for many students.

Fear of appearing uninformed: Asking a question shows that one does not understand a concept. No one wants to look foolish or be judged for how much they may or may not know.

Difficulty forming the question: Other times a student may struggle to formulate the question. Students in the process of learning English may be hesitant and unclear about how to structure the sentence grammatically.

Disabilities: A student may be deaf and providing a way for such a student to ask a question without attempting to speak can be of great benefit, that is assuming the professor does not know sign language or an interpreter for the student is not present.

2.0 Objective

The objectives of this project are:

- To create a way for students to ask a lecturer a question anonymously through technology.
- To Encourage classroom participation.
- To Allow more interaction between the speaker (instructor) and the audience (students).
- To make communication easier in the classroom.

3.0 Background

3.1 Key Concepts

Client: A *client* is a program that uses services that other programs provide. Client programs typically handle user interactions and often request data or initiate some data modification on behalf of a user.

Server: A *server* provide services to the clients. Server functions often require some resource management, in which a server synchronizes and manages access to the resource and responds to client requests with either data or status update.

Node.js: Node.js is an open-source runtime environment for building fast and scalable applications using JavaScript. Node.js works as the runtime and npm works as the package manager for Node.js.

Bootstrap: open source toolkit and powerful front-end framework for developing with HTML, CSS, and JS. We can use it to quickly prototype ideas or build entire app with Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

JavaScript: interpreted programming language that follows ECMAScript specification. *JavaScript* is the most widely deployed language in the world.

3.2 Related Work

Piazza is a platform designed to create communication between professors and students more efficient. It is designed to reduce email clutter and provides several ways to poll students and allows students to ask questions anonymously [1]. Our product is different in that it is designed for communication during a lecture.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

This project requires to develop a web application that will be used by students and teachers to help determine how concepts are being understood by the class during and after a lecture.

In order to make this possible, the following will be required:

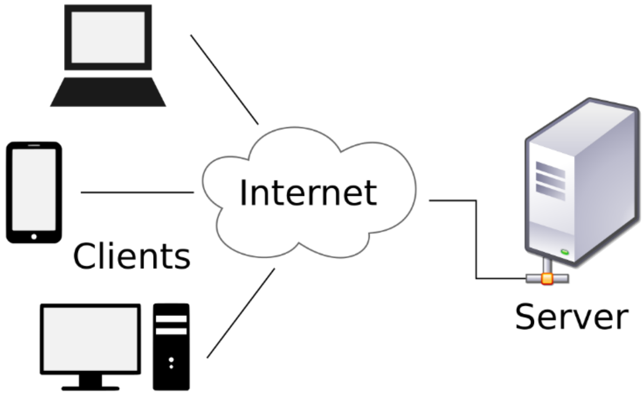
1) A server

2) The clients

3) A connection between the server and the clients

4) Computing devices

4.2 Client-Server Architecture



A client server architecture will be used to make this all possible. The server will host a session for the lecturer and the students to connect to. This will likely be implemented using Node.js, any other cross-platform environment, or language that would allow us to accomplish the main objectives.

4.3 Risks

Risk	Risk Reduction
Abusive language	Create a content filter.
Spam messaging	Lecturer can choose when to accept messages.
Outside invasion	Restrict access to the server to anyone not in the class/session.

4.4 Tasks

1. Plan designs and understand client-server architecture.
2. Begin implementation of architecture.
3. Design the user interface.
4. Begin implementation of user interface.
5. Test product for any risks or errors.
6. Document the results.

4.5 Schedule

Tasks	Dates
1. Plan designs and understand client-server architecture.	08/26/19 - 09/09/19
2. Begin implementation of architecture.	09/09/19 - 09/23/19
3. Design the user interface.	09/23/19 - 10/07/19
4. Begin implementation of user interface.	10/07/19 - 10/21/19
5. Test product for any risks or errors.	10/21/19 - 11/04/19
6. Document the results	11/04/19 - 11/18/19

4.6 Deliverables – Give a thorough listing and description of each item which will be submitted with your final, working project. Each major component should be described. The below is just an example list which should be replaced with your own.

- Design Document: Information about the client-server design
- Source Code: All source code used to create the product
- Final Documentation

5.0 Key Personnel

Jagmeet Wander – Wander is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has experience using HTML/CSS/jQuery/Bootstrap he also completed Software Engineering, Algorithms, Programming paradigms, Computer networks and Applied Cryptography. Wander has experience with UX design as he has taken few online courses and will be working at JB HUNT as UX intern in summer. His task will be Front End development and helping with Backend Programming.

Vladimir Sabado – Vladimir holds a BSc in Biology, but he is currently pursuing his second BSc which is in Computer Science. He is currently a software engineer intern for Affirma Consulting where he's working with his team to develop front-end and back-end applications to meet their client's needs. While his language of choice is Java, he has experience working with

C#, Python, HTML, JavaScript, C++, SQL, and Ruby. For this project, he will be working on the back-end.

Cristhian G. Espinosa S. – Cristhian is a Senior in Computer Engineering and a minor in Mathematics. Cristhian spent a semester in Jönköping, Sweden working on Internet and cloud services and developing backend applications connecting with local Swedish companies like IKEA and Scania. Current member of IEEE Eta Kappa Nu (HKN), Gamma Phi Chapter. He has worked for the Network Enterprise Systems Team for two years which involves working with Cisco, Juniper, and Aruba network infrastructure devices. He has worked with Laravel, HTML, PHP, Node.js, bootstrap, and some other web development frameworks. Cristhian has also worked with C++, C#, Python, Java. He has studied hardware descriptive languages including Verilog and VHDL. He has developed 2D and 3D videogames using the cross-platform engine Unity. Cristhian is currently working on his Honors Thesis in deep learning, neural networks and facial recognition.

Joseph Gorman – Joseph is a Senior pursuing a Bachelor of Science degree in Computer Science at the University of Arkansas. He has experience using several programing languages including C++, C#, Java, Python, and PL/SQL. He is the team leader for this project and will be guiding his team through the development process.

6.0 Facilities and Equipment

Computer hardware and software.

Git Hub

Slack

IntelliJ, PyCharm, sublime text etc.

Facilities: Google, Library, CSCE Labs etc.

7.0 References

[1] Piazza, (https://piazza.com/)