

# Clarity

Prototype: [Student](#), [Professor](#) ([ReadMe](#))

Akaash, Ari, David, and Sherman

## I. Problem and solution overview

Clarity encourages more communication between students and professors in large lecture environments. Professors frequently lecture without knowing if students are understanding and questions about material typically come at the end of lecture or not at all. Part of the reason why a teacher's lecturing style seldom improves is because all the feedback they receive is delayed and somewhat "after-the-fact." Like with anything else that requires trial & error to improve, teaching, too, should be rapidly iterative. Some students are afraid to ask questions while others are too overwhelmed by their confusion to even considering interrupting the professor. A live mobile/web-based interface allowing students to anonymously indicate when material presented by the teacher is unclear and ask questions can alleviate confusion and encourage teachers to spend more time/effort on unclear material on the fly. Equipped with this information, a teacher can answer questions, particularly the questions that are causing the most student confusion.

## II. Tasks

### 1. Student indication general understanding of material (Simple)

Clarity enables instructors to better gauge student understanding of material presented during lecture. As such, our first task aims to gauge student understanding with a finer comb. The professor ought to know how much of the class is understanding and how helpful answers to a question are. We want students to reflect and express understanding of material to professors. One change we made from our previous task is that the understanding button gauges overall understanding of material, and is not dependent on topics or specific material. This task is completed by updating the Clarity logo and lightbulb at the top of the screen. When they are "lit," the student is understanding, as soon as they are not understanding, they turn it off.

### 2. Student asking a specific question (Complex)

We intend to build out the forum for students to perform the task of asking questions of the instructor. We classify this as a complex task because of the competing agendas. On one hand, we need this interface to be clean and minimalist: we would not want a student to lose attention to the lecture because he or she has a question about material presented a few minutes ago. Students should be able to ask questions in real time with the material currently being presented. For this, students can easily type their question in on the home screen and submit it to the question feed.

### 3. Professor addressing a student question (Moderate):

Ultimately, we want the professor to be able to act on the questions being asked by students in the audience on our forum. In line with this vision, we hope to keep the

interface clean and legible, even to an instructor whose primary task may not be combing through the question feed or reading at all. Professors eliminate the question once they have answered it by double clicking on the question itself.

### 3. Revised interface design

We made three key changes as a result of the low-fi testing: add a question button, “got it” button, and the design of the professor interface. Below is a before (Figure 3.1) and after (Figure 3.2) picture of our homepage design.

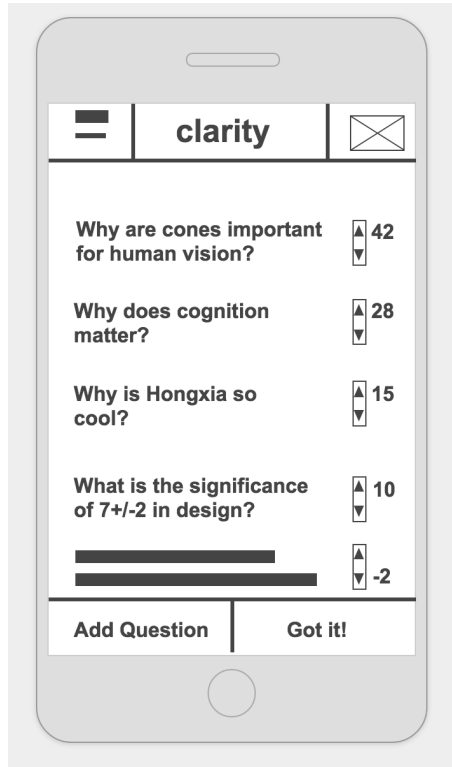


Figure 3.1 Low-fi Clarity homepage

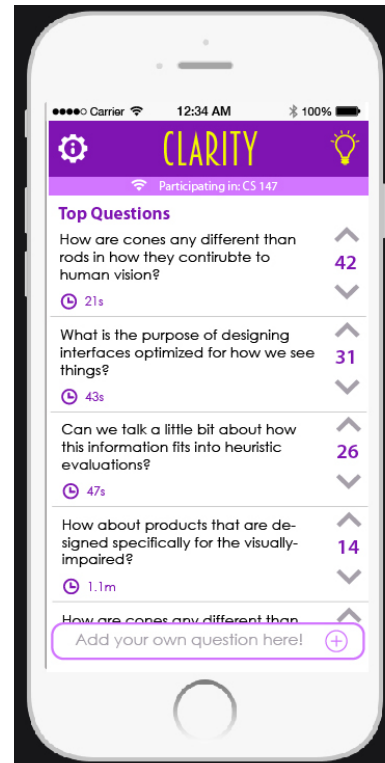


Figure 3.2 Medium-fi Clarity homepage

First, we knew we needed to simplify the “add question” feature. One issue is that we had two ways to add a question (depicted in Figure 3.3): pressing the add question button in the bottom left or pressing the envelope in the top right corner of the screen. We wanted to simplify this process by keeping the ability to add the question on the home screen. We redesigned it so that students could simply begin typing in a text box in the bottom of the screen to ask a question. Instead of going to another screen and having two ways of getting there, students can immediately ask a question so that they are less distracted and less confused. Our new design is shown in Figure 3.4 after the student has tapped on the text box.

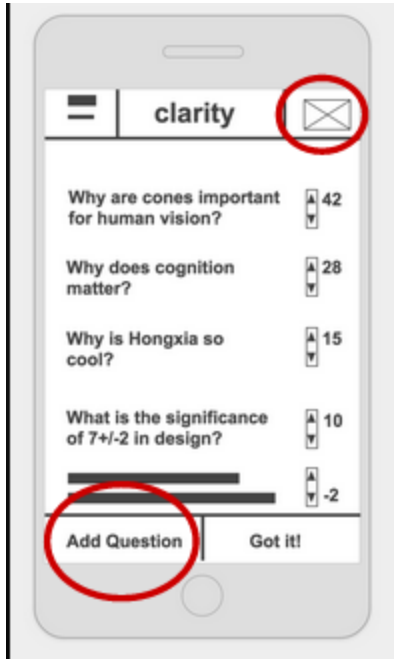


Figure 3.3 Low-fi “Add Question”

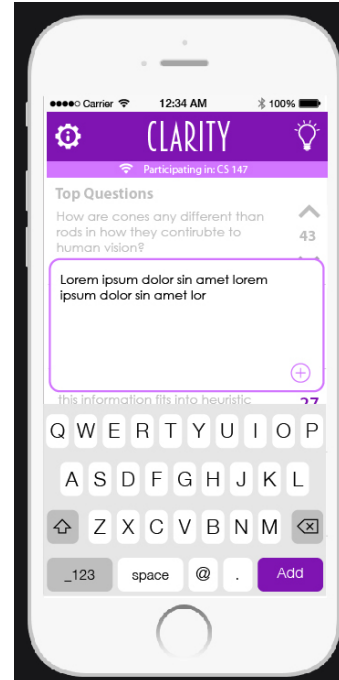


Figure 3.4 Revised “Add Question”

The second feature we focused on redesigning is the “Got it” button (shown in Figure 3.5). None of the participants in our testing understood the purpose of this button. We intended it to be the way that students showed understanding to the professor. However, when they pressed it, it brought the student to another screen, they had to find their way back to the home screen, and then there was no indication that they were still understanding. We wanted to keep it simple for students to show they were understanding the material. We needed to have better feedback that they had indicated their understanding. We are now thinking of it like a lightbulb (a cliché of intelligence) that is kept on while the students are understanding, and turned off while they are not. We decided to focus it at the top of screen with the Clarity logo (Figure 3.6). The student will press the logo and it lights up until they press it again.

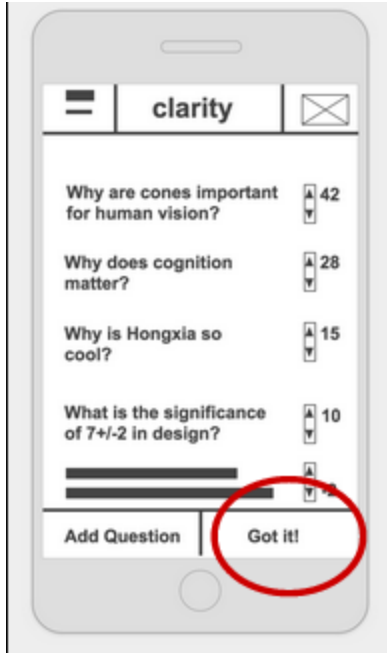


Figure 3.5 Low-fi “Got it” button

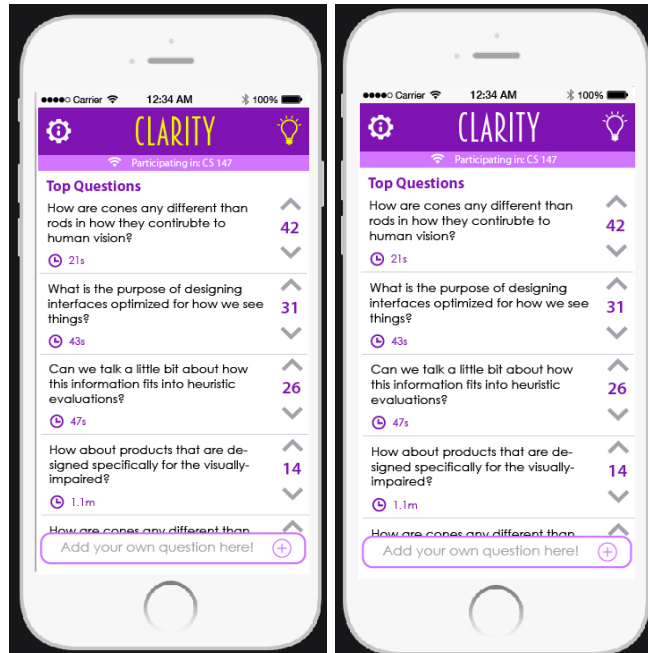


Figure 3.6 Med-fi Clarity “lit” up and turned off

Finally, we made key revisions to the professor’s screen. We removed the timer figure because it proved confusing. We wanted the most focus to be on the questions the students want answered. The professor does not need to see the upvote and downvote buttons; the only necessary information is the question and how pressing they are for the students. These changes are depicted in Figures 3.7 and 3.8.



Figure 3.7 Low-Fi Professor design

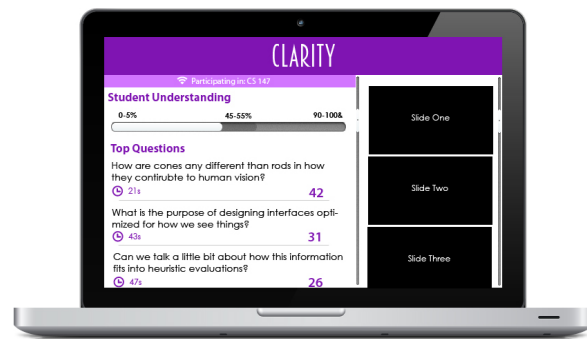


Figure 3.8 Med-fi Professor Design

To see annotated scenarios for each task, please see the Prototype Screenshots section.

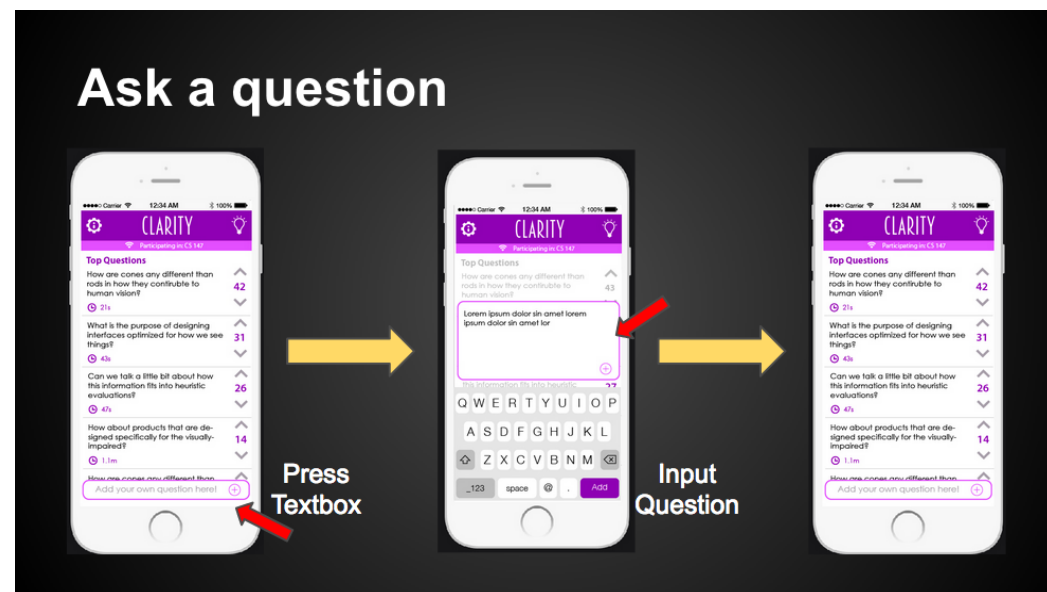
#### 4. Prototype overview

Marvel is a well-designed prototyping tool built by a startup out of London. It allows the ability to link sections of images to develop and design a pageflow of a mobile/tablet application. The

tool in this project helped us to design and fully envision fine visual and user interaction details with the application without the need to implement the actual code behind the storage. Though the interface was intuitive and simple to use, we would have liked to see scroll functionality in the prototyping tool.

Marvel natively supports touch, pinch, and swiping gestures but because one large portion of our mobile application involves scrolling - we would have liked to see that feature built-in to the prototyping tool. This means we are currently not able to scroll through questions on the main page. However, this will be achievable in the actual development of the application.

## 5. Prototype screenshots



# Supporting a Question



# Professor Answers Question

