

Speak↑: Low-Fi Prototype

“Better lectures, powered by real-time student data”

Team



Name	Email	Role
Brad Reyes	breyes28@	Developer
Karen Gomez	kgomez@	User Testing
Nick Akiona	nakiona@	Design
Reid Watson	rawatson@	Manager and Documentation

Introduction

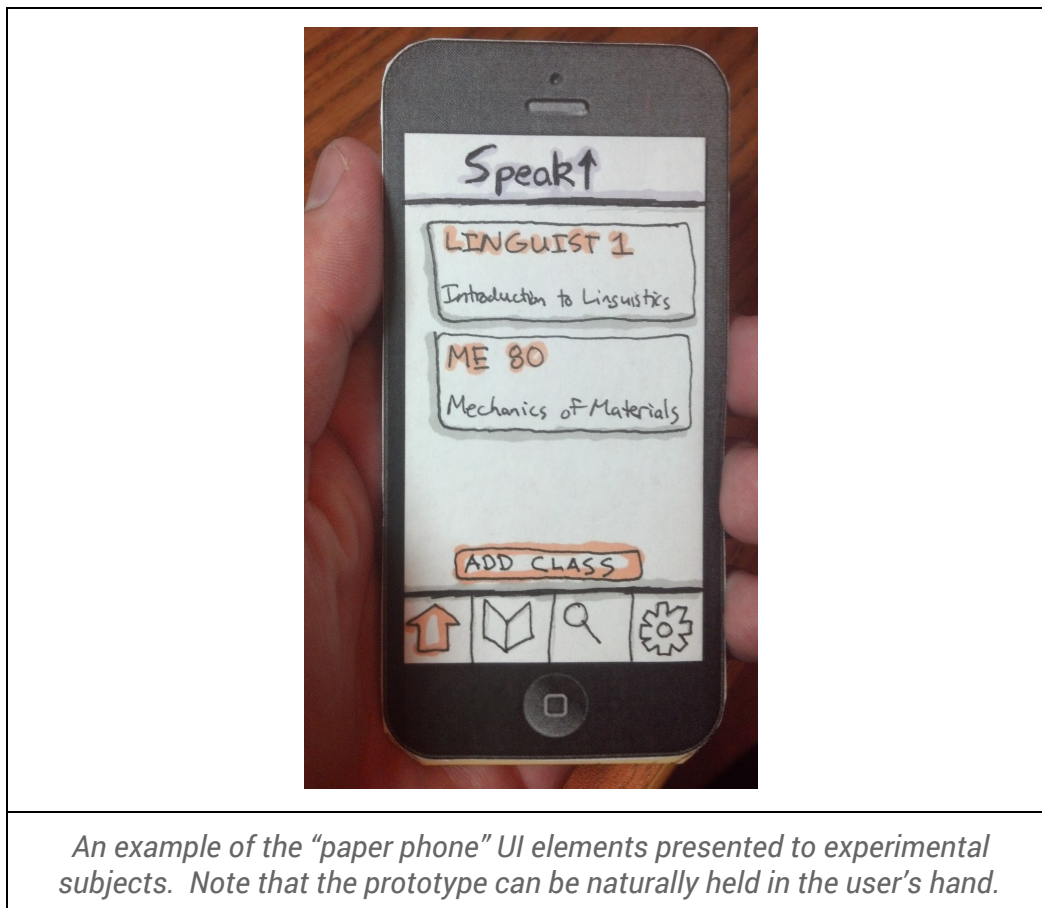
This document describes a usability experiment conducted by the SpeakUp team in the Fall 2014 offering of CS147 at Stanford. Team members constructed a “paper-prototype” of a mobile app for enabling feedback and communication in large lectures. UI flow was tested for both students and professors, revealing a number of possible design changes.

Mission Statement

SpeakUp will facilitate communication between the instructor and students in lectures with over 50 participants. Students will be able to submit honest feedback, and lecturers will be able to see granular data about student understanding.

Low-Fi Prototype Description

Our interface was created using a series of 21 “paper phone” UI cutouts. Each UI cutout measured 2.26 x 4.70 inches (including bezel). The usable screen area measured 1.93 x 3.43 inches. Each cutout was placed onto a piece of foam board cutout. This allowed users to hold the paper phone just like they would a normal phone, without having to worry about bending their phone. Pictures of all UI elements are in the appendix.



Main Functionality

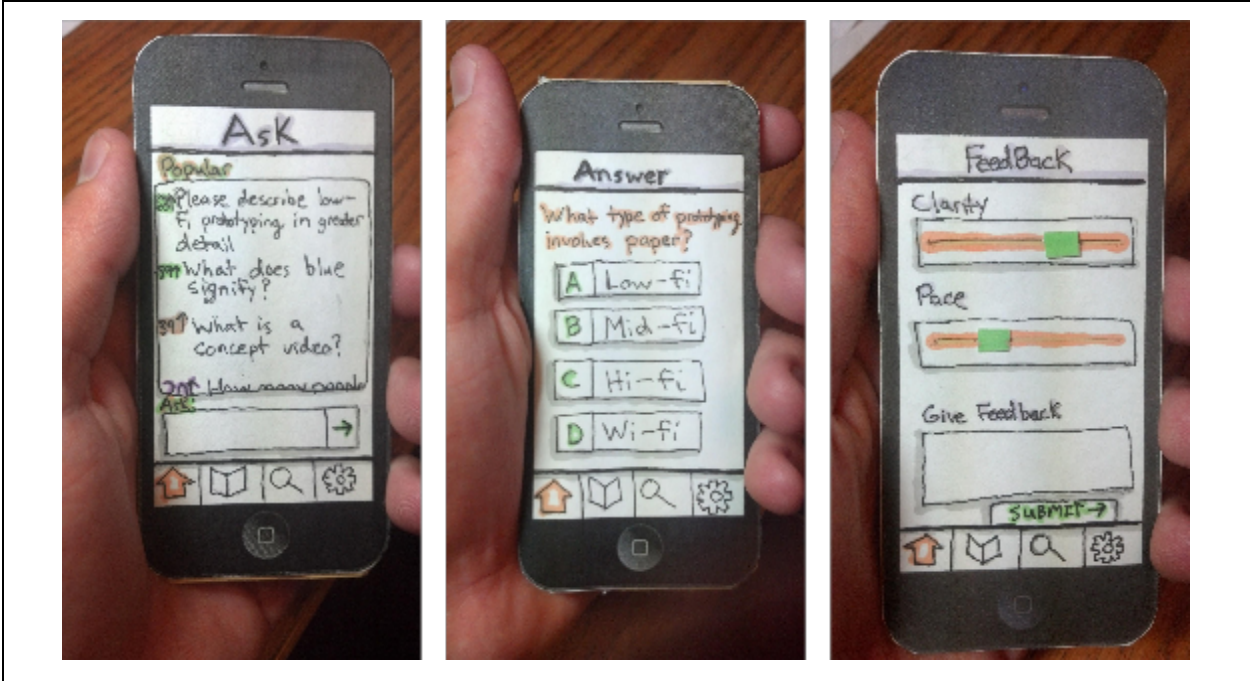
The SpeakUp app is focused around enabling three main interactions:

- Enabling students to submit real-time feedback in large lectures.
- Allowing students to ask questions, and view the questions that others have.
- Giving instructors the ability to ask questions to their class (like iClicker)

Interaction Ideas

All interaction with the speakup app is done through:

- Taps
- Swipes
- Text input with the phone keyboard

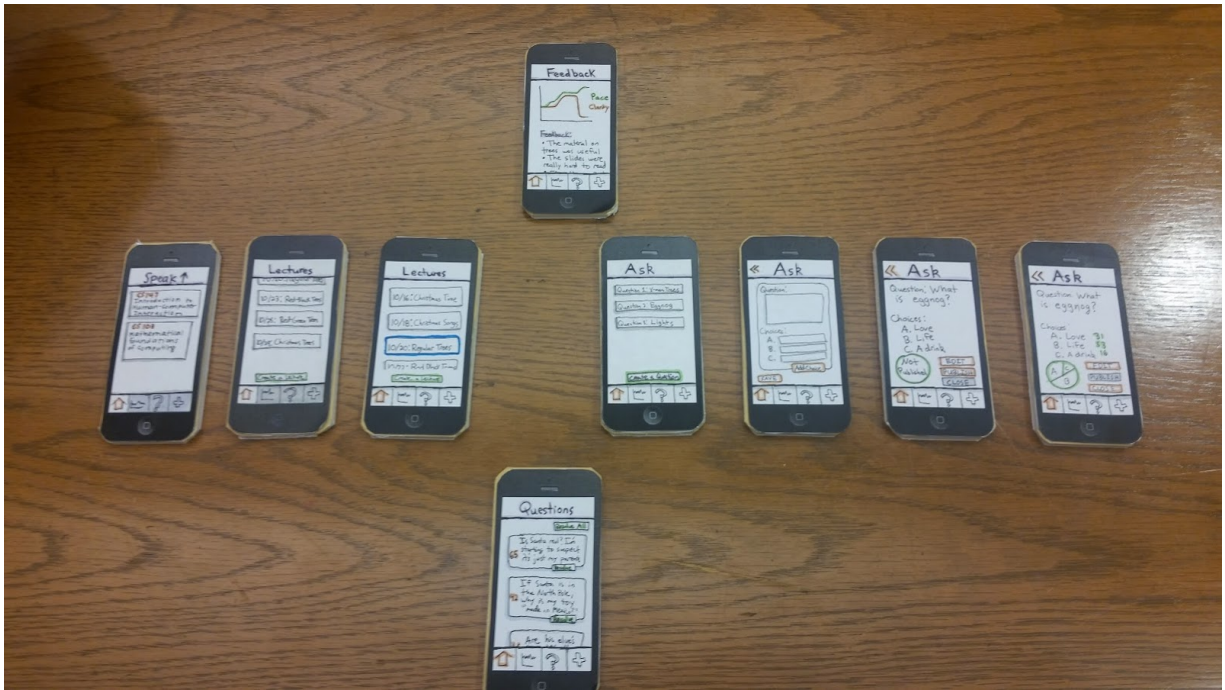


The student UI for asking questions to the instructor, answering questions from the instructor, and providing feedback. Students swipe left and right to navigate between screens

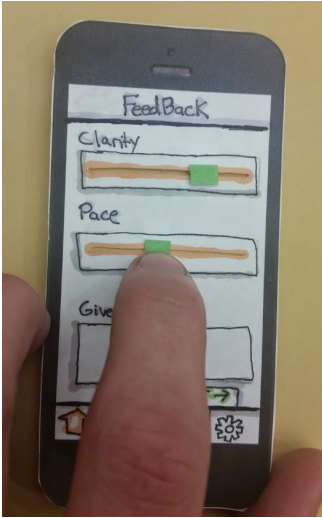
Complete System



The complete student UI



The complete instructor UI



Our favorite interface: a feedback screen with sliders which actually move.

In order to keep this paper at a reasonable length, we've omitted a formal specification of all possible actions in our UI. However, the actions required to complete the tasks we chose are specified in detail in the [testing procedure](#) section.

Testing Method

We assumed the same roles for all of the experiments conducted in this project. In some of the experiments we had only one note-taker, but ours setup was otherwise unchanged.

Facilitator	Karen Gomez
Computer	Brad Reyes
Notes	Nick Akiona, Reid Watson

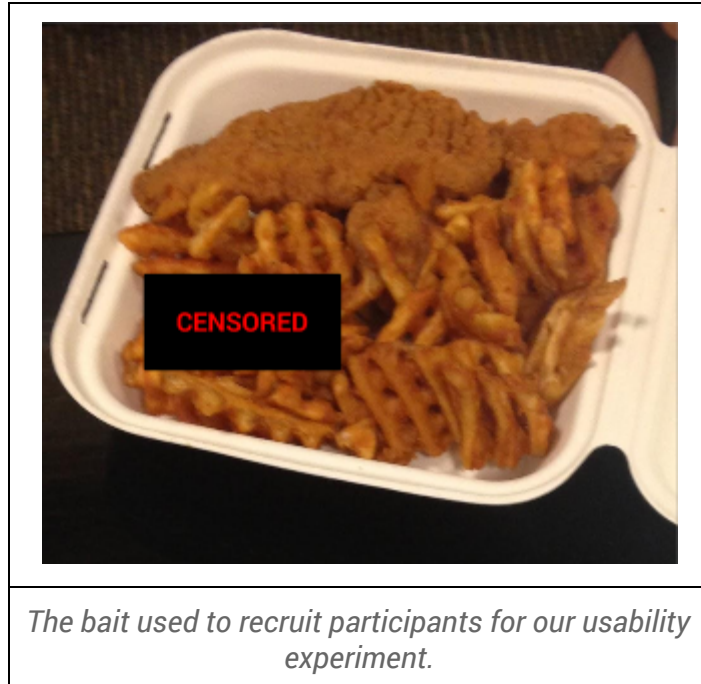
Student Testing Environment

All tests with students were conducted in The Axe and Palm, a cafe located in the central area of the Stanford campus. We reserved room 113 and conducted all of our student usability tests inside. Karen sat across from the participant, while Brad and Reid were seated adjacent to the participant, affording each of them a good view of the participants actions

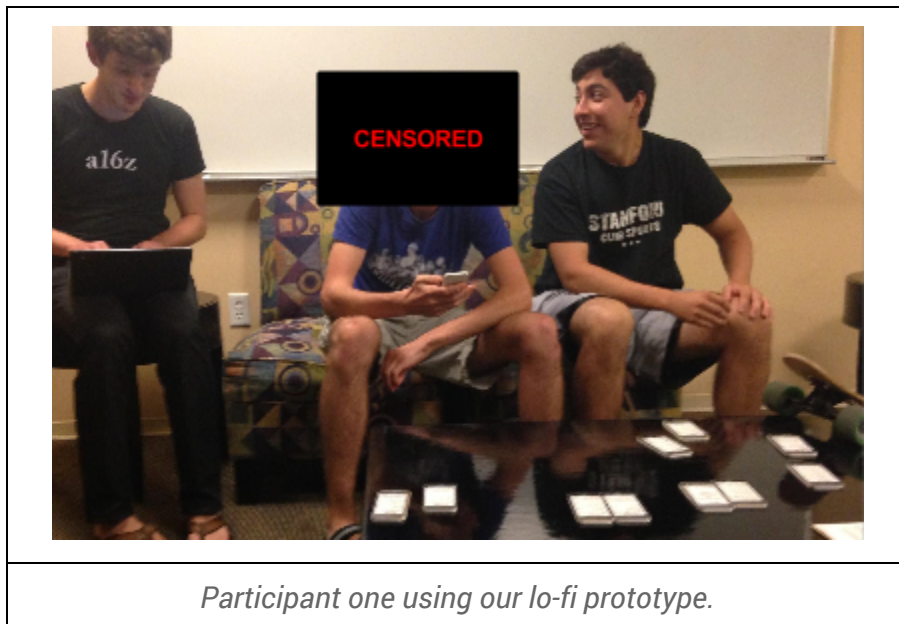


A photo of room 113 in the Axe and Palm, where our student experiments took place. Participants were seated on the couch in the right side of the room.

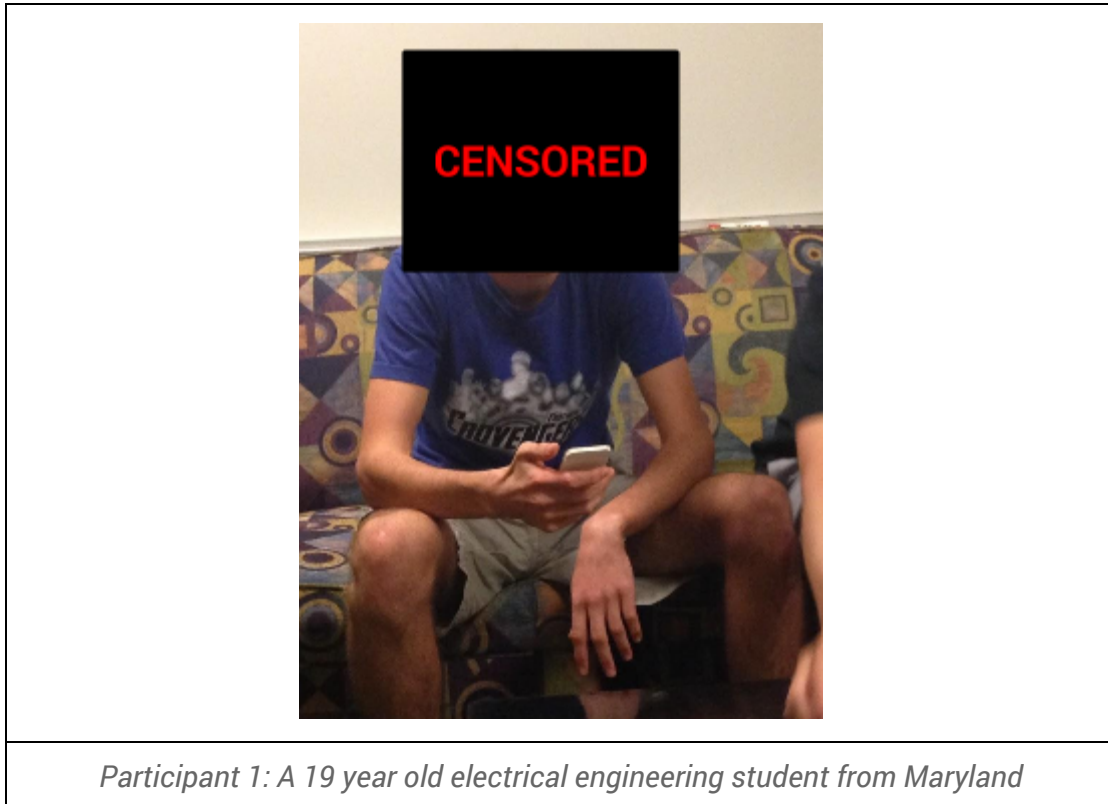
In order to entice students to participate in our usability experiment, we offered them delicious food, in the form of the ever-popular chicken tenders and waffle fries combo from The Axe and Palm (see below).



As an example, the following picture was taken by Karen during our first usability experiment with participant one.



Student Participants



Name	Participant 1
Age	19
Occupation	Student
Education	Studying for a BS in Electrical Engineering
Primary Smartphone	iPhone
Other Info	Participant one has a bit of a background in CS, and is from Maryland
Recruitment	Found him in the Axe and Palm
Compensation	Offered delicious food (he declined)
Testing Environment	Room 113 of the Axe and Palm(see above)



Participant 2 (left): A 19 year old electrical engineering student from Maryland

Name	Participant 2
Age	21
Occupation	Student
Education	Studying for a BS in Science, Technology, and Society (STS)
Primary Smartphone	Android
Other Info	Speaks four languages, and grew up in the Washington DC area.
Recruitment	Found him in the Axe and Palm
Compensation	Offered delicious food
Testing Environment	Room 113 of the Axe and Palm(see above)

Lecturer Testing Environment

All tests involving lecturers were carried out in room 200 of the Gates Computer Science building at Stanford. We used a setup similar to the one we had at the Axe and Palm, with Karen on one side of a table, and Brad, the participant, and the note taker on the other side.



A photo of our lecturer testing environment in Gates 200

Lecturer Participants



Participant 4 (left): A current grad student / lecturer at Stanford

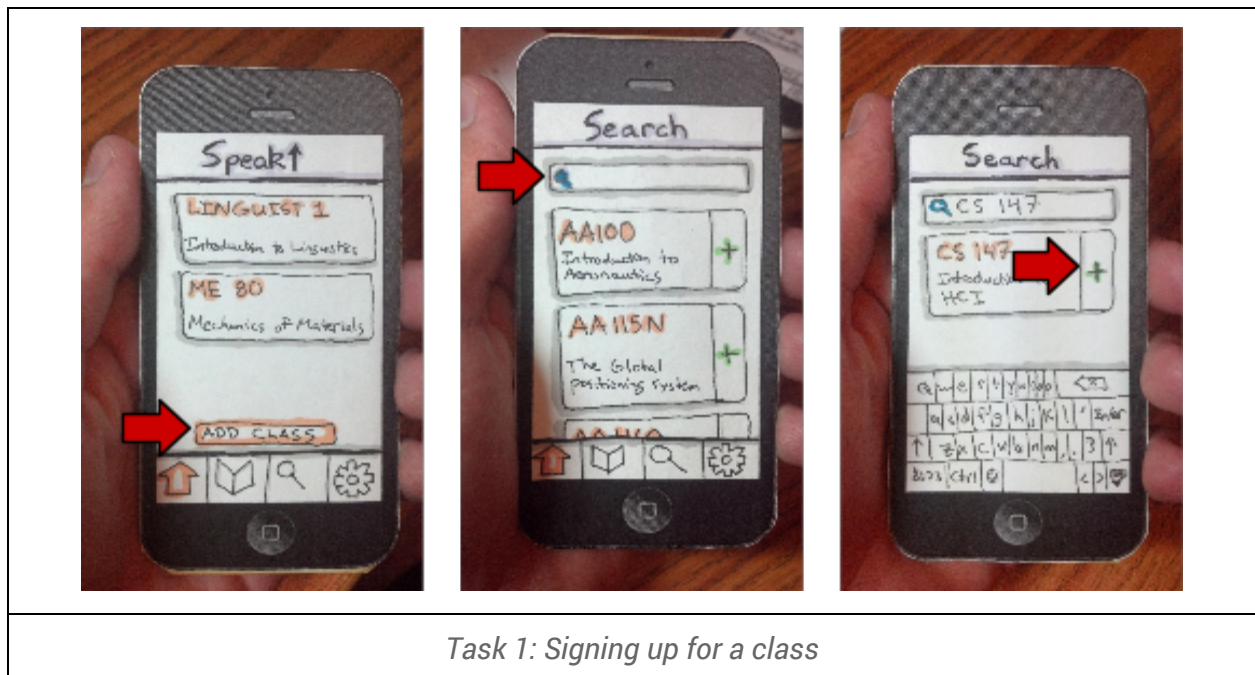
Name	Participant 4
Occupation	Student
Education	BS Computer Science (Stanford)
Primary Smartphone	Android (Nexus 5)
Other Info	Currently a grad student in Stanford CS, lectured for a course on Haskell
Recruitment	Frantic searching around Gates after our original interview subject cancelled last minute
Compensation	Our eternal gratitude
Testing Environment	Gates 200

Testing Procedure (Student)

[See page 5 for experimental roles of team]

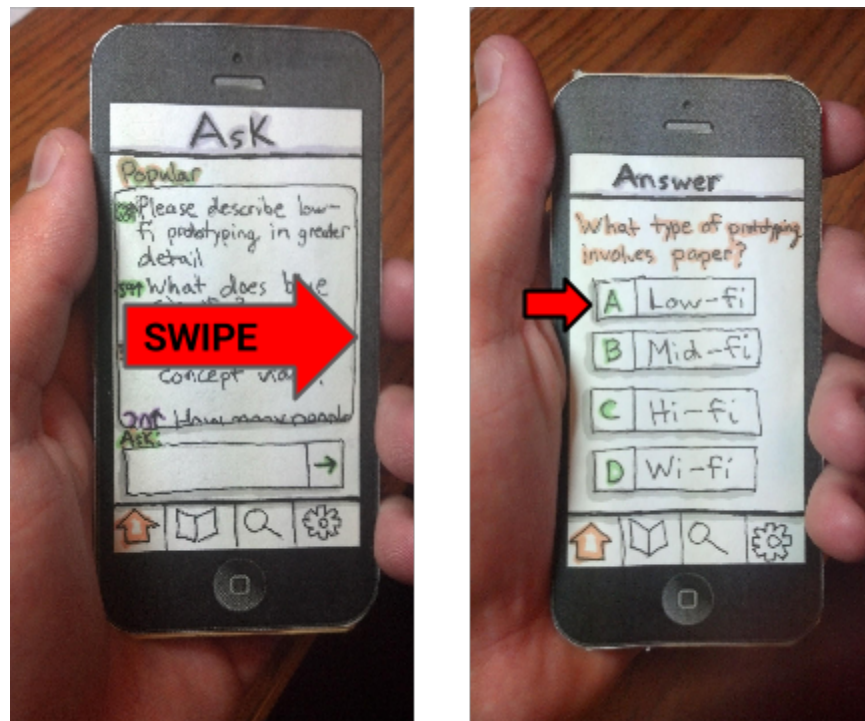
Students were required to complete four tasks, three of which we designated as our “primary” tasks (see [the appendix](#) for the exact phrasing read to participants):

1. Sign up for a class
2. Indicate to the instructor that they have a question about a particular concept
3. Answer a question posed by the instructor
4. Submit feedback on the lecture

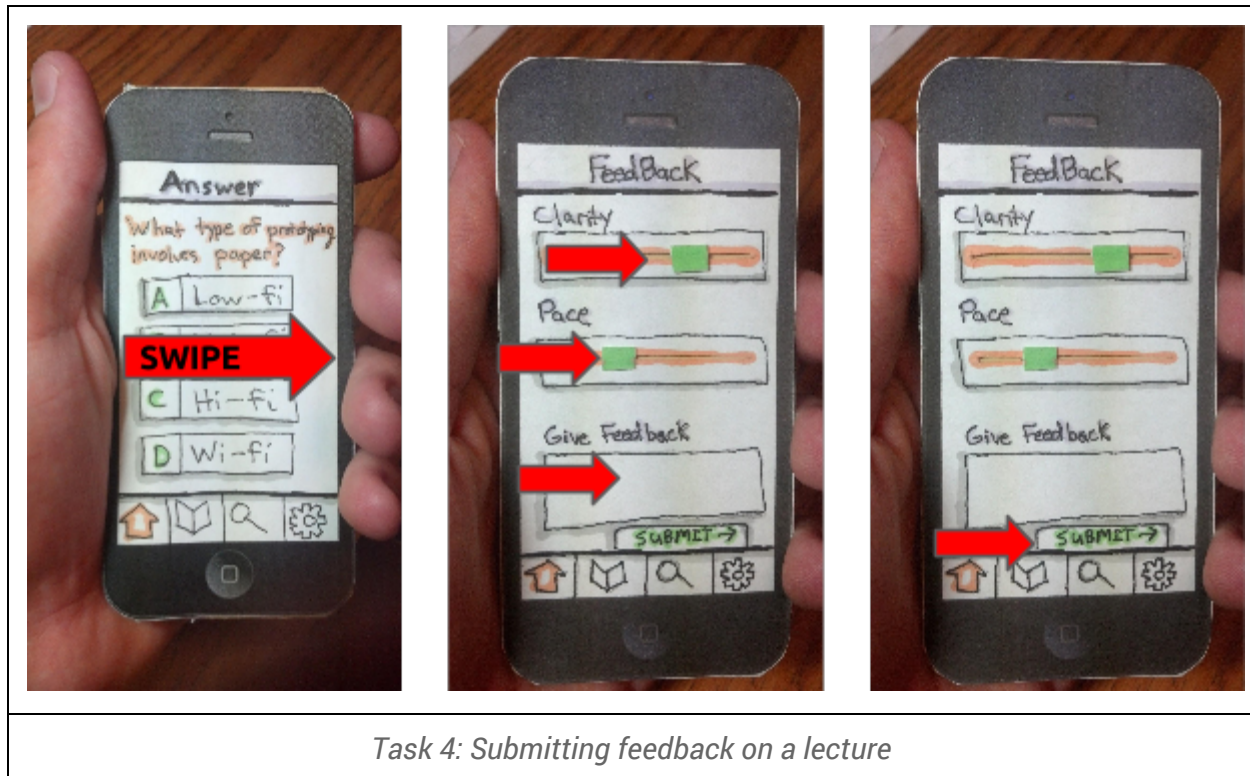




Task 2: Voting up a question asked by another student.



Task 3: Answering a question posed by the lecturer



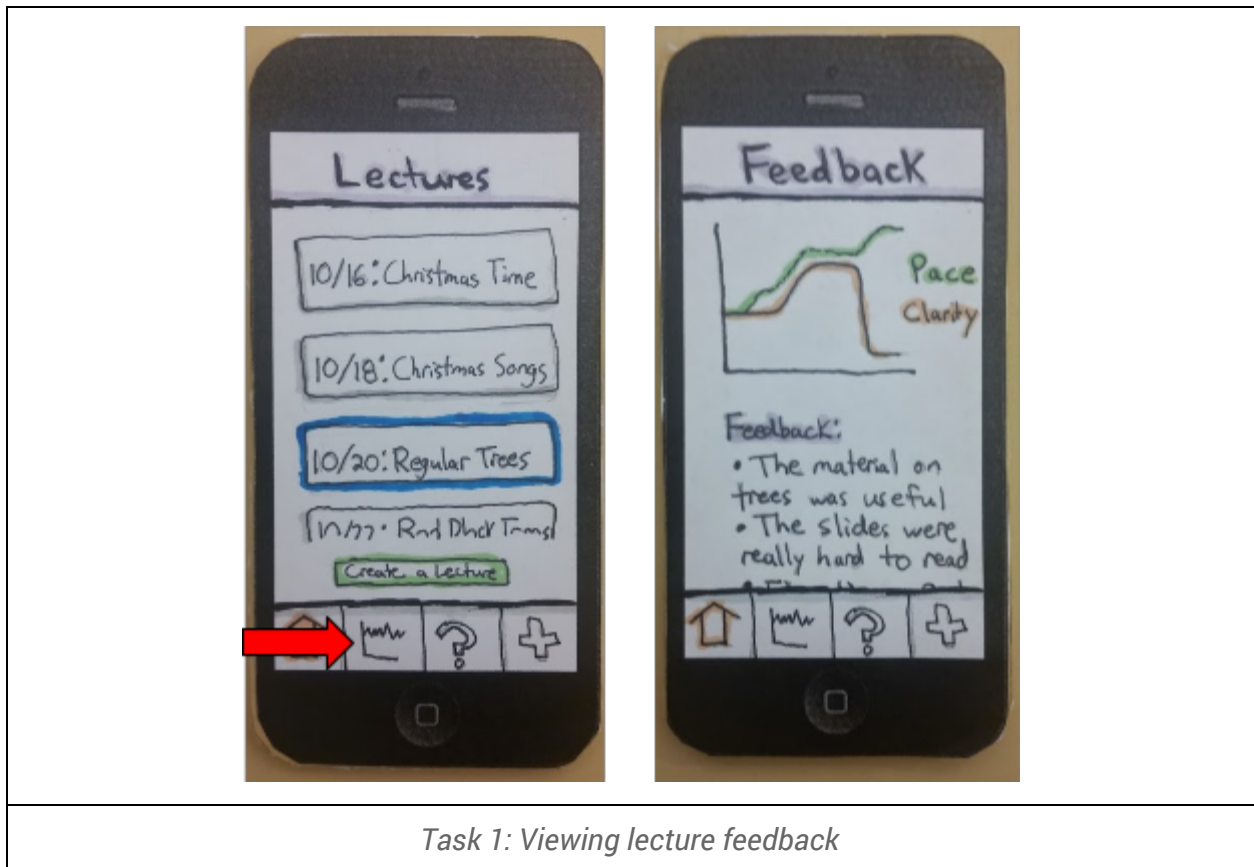
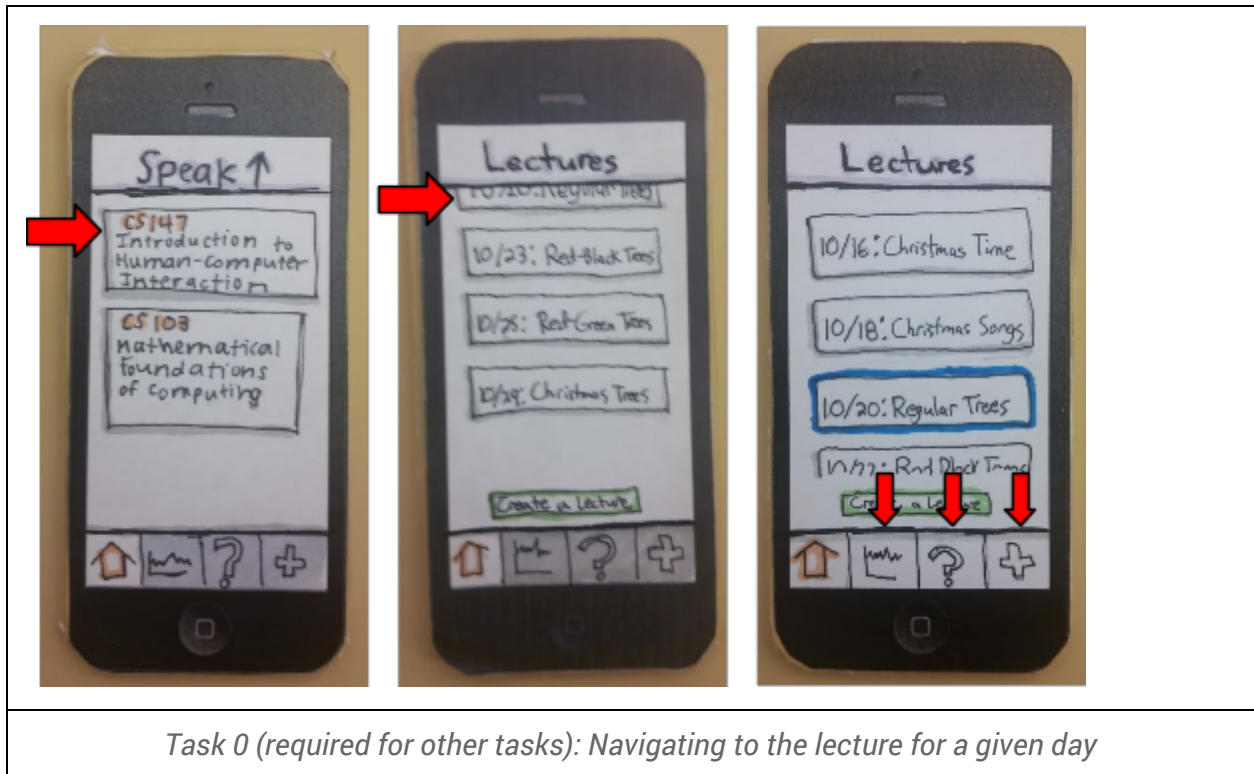
Task 4: Submitting feedback on a lecture

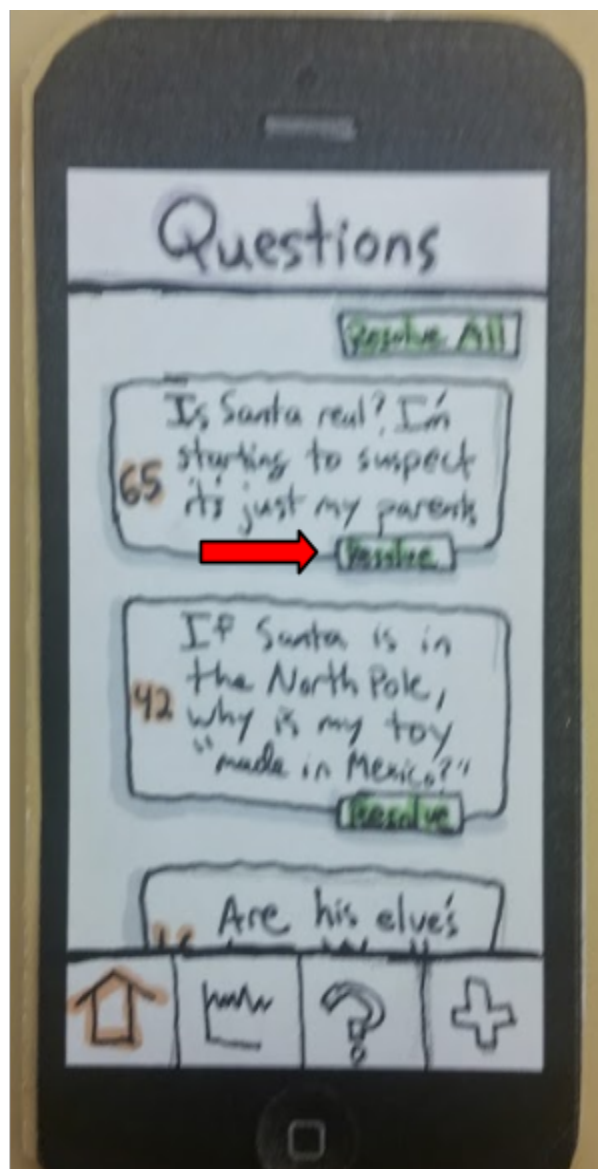
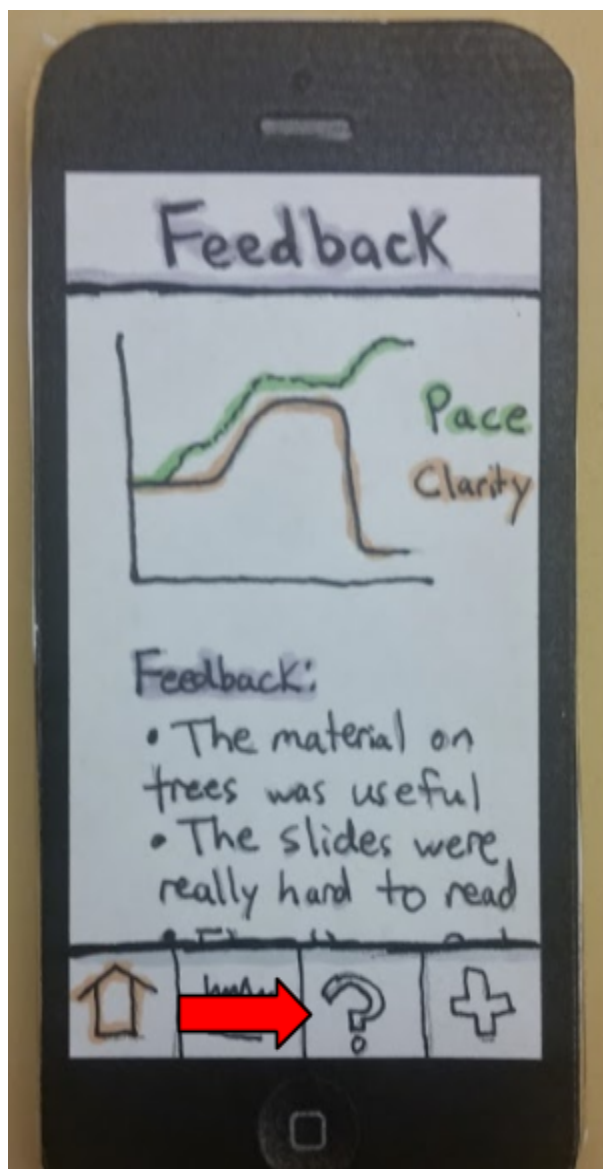
Testing Procedure (Lecturer)

[See page 5 for experimental roles of team]

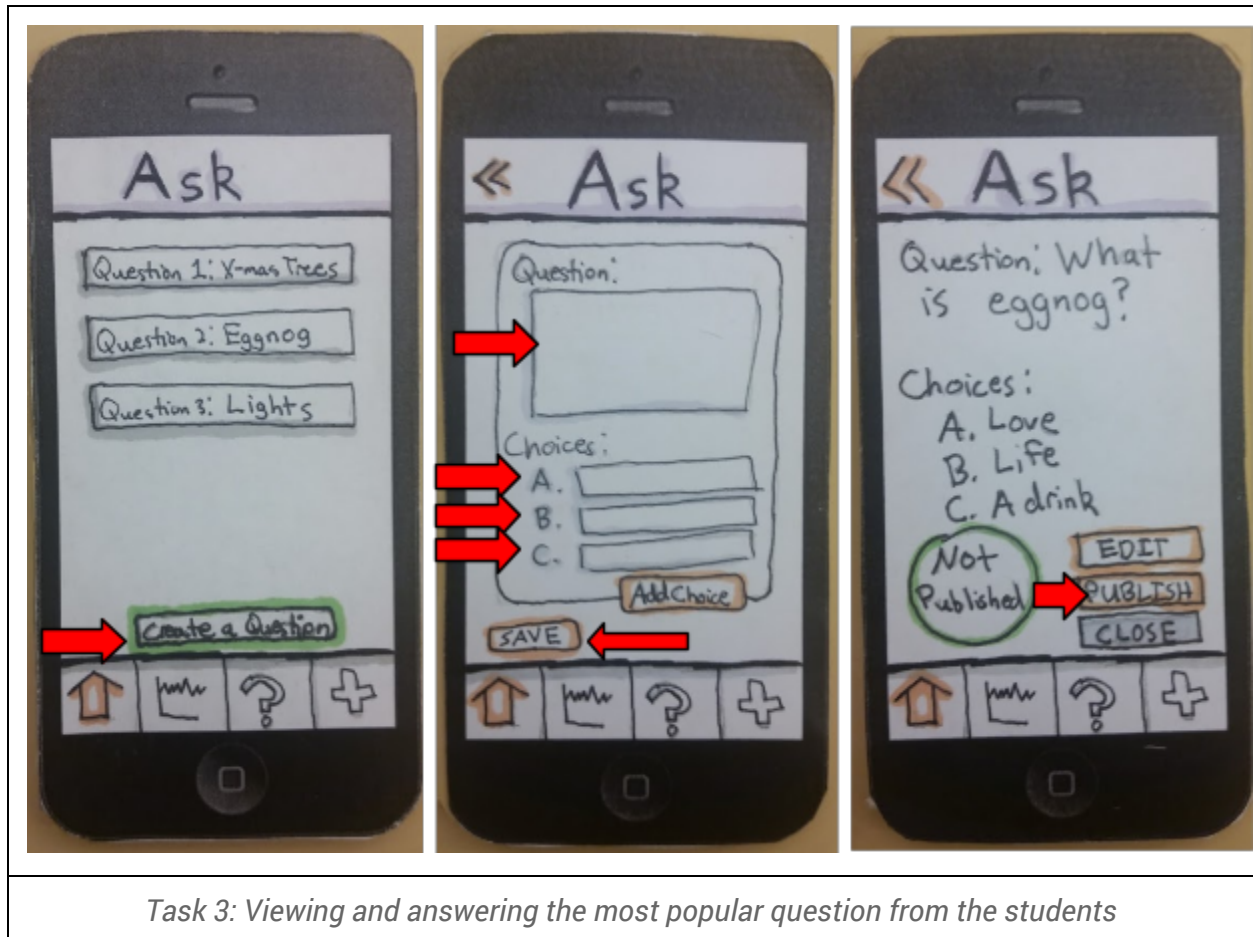
Lecturers were required to complete three tasks (see [the appendix](#) for the exact phrasing read to participants):

1. **View feedback on their lecture**
2. **Read and answer the most popular question from students**
3. **Create a question for their students and open the question for answering**





Task 2: Viewing and answering the most popular question from the students



Data Collected

We recorded the actions a user took as they interacted with our application, specifically, any mis-steps which deviated from the “easiest” way to complete a task. We decided not to log time to task completion, since such data would be inconsistent due to the inherent difficulties of the paper prototype system (delays due to finding UI screens).

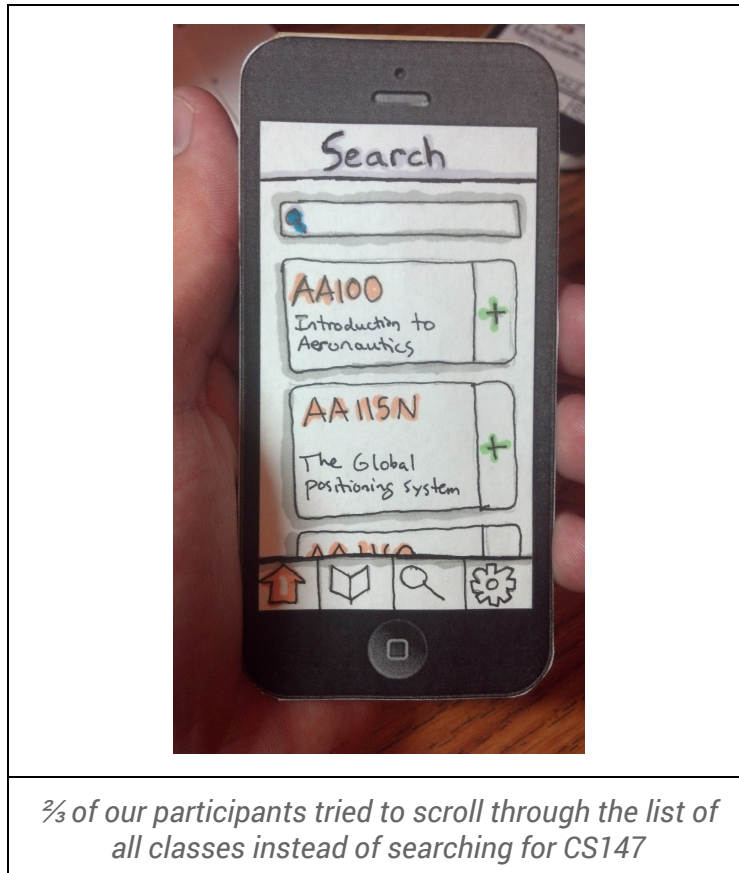
We tried to make notes about a user’s general reaction to our prototype as well as recording the very granular data about user actions.

Testing Results

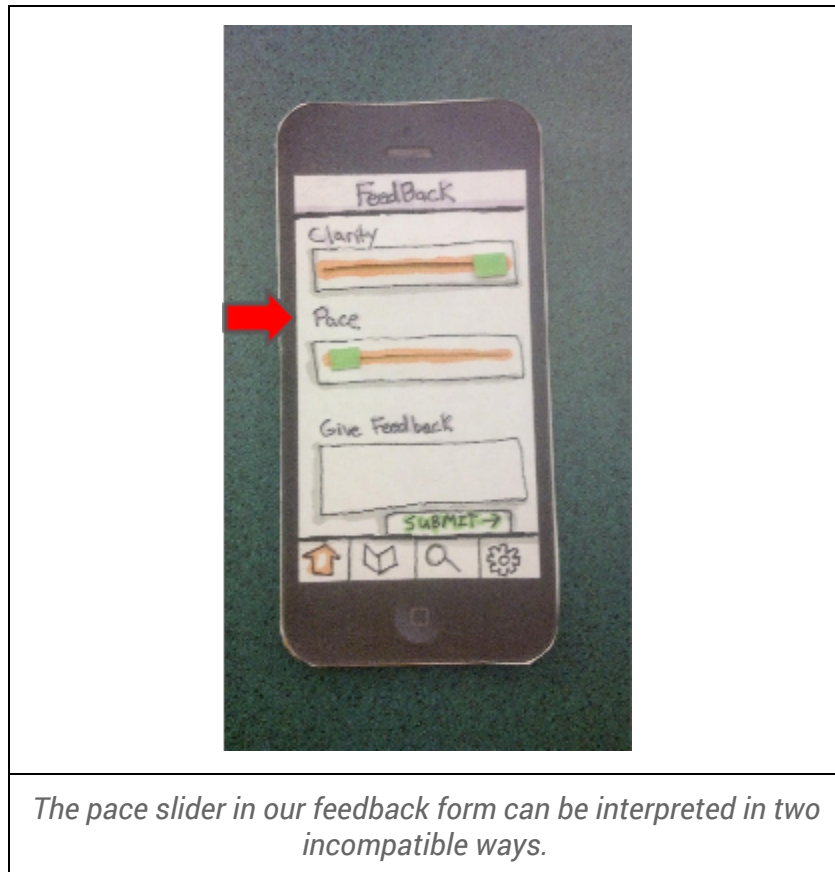
A more complete log of our results is available in [the appendix](#), but we’ve highlighted a few key takeaways in this section:

- **Swipe gestures are difficult to discover:** All 3 of our student participants were unable to find the feedback and response section of the app. We had to suggest the idea of a swipe gesture for the students to complete the task. As participant 3 put it: “I just didn’t know it existed”
- **Users will scroll if given the option:** When students add a class, they are given the option to scroll through a list of all of the classes in our system. Users can also start

typing into a search box to jump to a specific class. Despite the fact that there were an obviously large number of classes listed in our app, two participants first tried to find CS147 by scrolling through a list of classes. This would take a very long time in the real app.



- **Ask and feedback seem too similar:** Our model of the app includes separate functionality for asking a question and providing feedback to the lecturer, but participant three tried asking a question saying “your pace is too fast” to provide feedback.
- **No one tried to vote up existing questions:** The question we told students to ask was actually already listed as an “asked question”. We were hoping that one of our student participants would try and interact with that question, but none did.
- **Newly enabled buttons should be made obvious:** When lecturers select their class, we ungray the three buttons at the bottom of the lecturer UI. However, participant four didn’t notice this, and wasn’t sure what he could do next.
- **“Pace” is a tricky term:** Student participants were told to submit feedback that a lecturers “pace was too fast”. Some students lowered the “pace” slider to it’s minimum value, indicating that the pace was bad. Some students raised the pace slider to its maximum value, indicating the pace was too fast.



Discussion of Results

Some obvious changes follow from the issues mentioned in the [testing results](#) section. Additionally:

- **The student icon bar needs to be changed:** We designed the student icon bar (book, magnifying glass, settings icon) before we fleshed out other parts of the app. Unfortunately, the most critical functionality was handled with the swipe gesture instead of the bottom menu. We should change this in future iterations.
- **If we keep the swipe gesture, we need to make it discoverable:** None of our student participants discovered the swipe gesture on their own -- we'll need to tweak that somehow.
- **We need to clarify our ask / feedback messaging:** The current naming, icons, and overall design of the ask questions and give feedback UIs are too confusing. Not only did participant three have issues submitting feedback, but other participants seemed surprised that there was a separate feature for submitting feedback.

Our experiment was done with a low-fi prototype, so it's definitely not perfect. For example, lecturers may have had an easier time noticing when icons become ungrayed if they didn't have to refocus on a new cutout UI. However, we were overall very satisfied with the lessons we learned from our paper prototype.

Appendix

Consent Forms

As per [the Piazza post](#), we're including only [our consent form](#) without any signatures on it. Scans of filled out forms are available if needed.

Script for Participants (General, read to both students and lecturers)

Welcome to our usability test! We're working on a product called SpeakUp, which is designed to facilitate communication in large lecture settings. Students can use SpeakUp to ask questions, answer questions from the professor, and offer feedback on lecture. Lecturers can pose questions to students, answer questions written by students, and read live feedback about their lecture.

First, we'd like you to take some time to familiarize yourself with our app. Take a look at the paper phone in front of you. You can interact with this paper phone in the same manner that you would interact with a normal smartphone. You can tap, swipe, and press any buttons you like. We'll "handle" your actions by showing you a different prototype, representing a different component of the UI.

Once you're finished with that, we'd like you to complete a series of four tasks using our prototype system. To accomplish these tasks, you'll have to explore the UI of our product. If you get stuck, you can ask us for help, but we're going to avoid dictating the process to you.

Script for Participants (Student)

Let's say that you were sitting in the first lecture for a class you were taking: CS147. The professor instructed you to download SpeakUp, and you did. Now, using our app, try and sign up as a student in CS147.

Great, now let's say that you're in CS147 lecture. The professor is describing the requirements for an upcoming assignment. You're required to interview a certain number of people, but he wasn't very clear about how many. Using our app, try and indicate to the professor that you have a question about how many people to interview.

Fantastic. But what's this, the lecturer has set up a question for you to answer using the SpeakUp app! Find the portion of the SpeakUp app which allows you to electronically answer his question. Due to the fact that you pay such good attention during lecture, you know that the correct answer is "low-fi"

Almost done! You've been in lecture for a while, and you'd like to submit some feedback on the lecturer's teaching. Specifically, you think he's moving too quickly and you're having trouble understanding what he's saying, especially when he started talking about the human eye.

Script for Participants (Lecturer)

One of the features offered by SpeakUp is the ability for students to easily submit anonymous feedback about your lecture. Take a second now to see what students have said about your lecture.

Additionally, SpeakUp allows students to anonymously submit questions for you to answer. Take a moment now to view the questions asked by students, and answer the most popular question.

Finally, SpeakUp allows you to ask questions of the students in your class. Use our question functionality to create a new question asking students “what is eggnog”. Possible answers should be “love”, “life”, “a drink”. Once you’ve finished creating the question, publish it for your students to answer.

Raw Data

Participant Number	Task	Severity	Event
1	1	1	Looking at the history screen: “What’s this for?”
1	1	0	Able to add a class very quickly. No mis-steps or misses
1	2	2	Asks his question without checking whether or not it’s already been asked. Otherwise no issues with task 2.
1	3	2	Confused to see that search and add class lead to the same point.
1	3	4	Has no idea how to get to the part for answering -- the swipe gesture isn’t very intuitive
1	3	0	Answers easily once he finds the correct section
1	4	3	His instinct for submitting feedback is to ask it as a question.
1	4	1	Cranks up clarity to max, pace to min, types in some feedback, then hits submit

Participant Number	Task	Severity	Event
2	1	2	When asked to add the class, he starts scrolling through our ridiculously long list of classes. Eventually decided to search, but only because we "wanted him to".
2	1	0	Able to add a class very quickly. No mis-steps or misses
2	2	2	Asks his question without checking whether or not it's already been asked. Otherwise no issues with task 2.
2	3	4	Navigates to the ask page for CS147 first, then to the search page, then to the home page, then to the ask page, then to the history page, then to the home page, then to the ask page again.
2	3	3	We had to tell him to try swiping, at which point he found the answer page and submitted without issue
2	3	0	Answers easily once he finds the correct section
2	4	2	Starting from the answer page, he first goes to settings, then to home, then to cs147
2	4	2	He puts pace up to the maximum value, then leaves the clarity bit right around the middle.

Participant Number	Task	Severity	Event
3	1	1	"Is there a back button"?
3	1	1	Finds the add a class interface, but then tries to scroll through the list of all of the classes
3	2	2	Asks the question, but doesn't realize that the question has already been asked. Otherwise no issues with task 2.
3	3	4	Tried to swipe a specific item once we told her to swipe. Didn't know what to do to be able to answer the question before we told her to swipe.
3	3	0	Answered without issue once the answer screen was located

3	4	2	Hits home button, class button, then asks a question instead of offering feedback
3	4	0	Now that she's seen the swipe gesture she has no problem figuring out the swipe to the other direction
3	4	2	Lowers both pace and clarity
3	4	2	Doesn't seem to know that she has to click submit.
3	4	0	Really liked the paper slider

Participant Number	Task	Severity	Event
4	1	3	Taps around, doesn't understand how to navigate from selected lecture to content for each lecture, goes from lecture to home, doesn't understand to press the three buttons at the bottom of the screen
4	1	2	Wanted to go back to the lectures screen on the feedback screen, but wasn't able to
4	1	0	Understood the feedback just fine once he found it
4	2	0	Tapped on the correct icon for students asking questions
4	2	0	Tried to scroll, but we don't have enough content
4	2	2	Presses on the top question itself, didn't press the resolve button until it became clear we wanted him to click resolve
4	3	0	Types sideways!
4	4	0	Fills out all of our text fields well, understands to press save.
4	4	1	Thought the question mark was help for the current screen.

Pictures of System

To decrease filesize, a complete photo catalog is hosted externally at <http://stanford.edu/~rawatson/CS147Photos.zip>