

HelpList

Andres C., Santos H., Minmin H.

Value Proposition

Anonymous in-class asking, ranking and answering questions for college lectures

Problem/Solution Overview

College students can't always ask questions when they get confused in lecture, which may lead to not being able to understand the remainder of the lecture. Furthermore, it is not always apparent which questions are the ones that the majority of students are confused about to faculty, or to students who are willing to help. We are building an unintrusive mobile platform where college students can stay engaged and understand lectures by anonymously asking questions, getting their questions answered, and helping others learn by answering questions.

UI Sketches



Figure 1. Rough sketches of UIs of different modalities

We sketched multiple UIs of different modalities, including mobile apps, wearable apps, voice interaction, force touch, and web apps (Figure 1). The two UIs that we chose were a mobile app (Figure 2) and a wearable app (Figure 3). The mobile app emphasizes on simplicity. The home screen features buttons for functions that students use most frequently: ask a question and answer a question. The wearable app running on smart watches can be controlled by voice as well as touch and scrolling. Students can interact with it through voice and tapping.

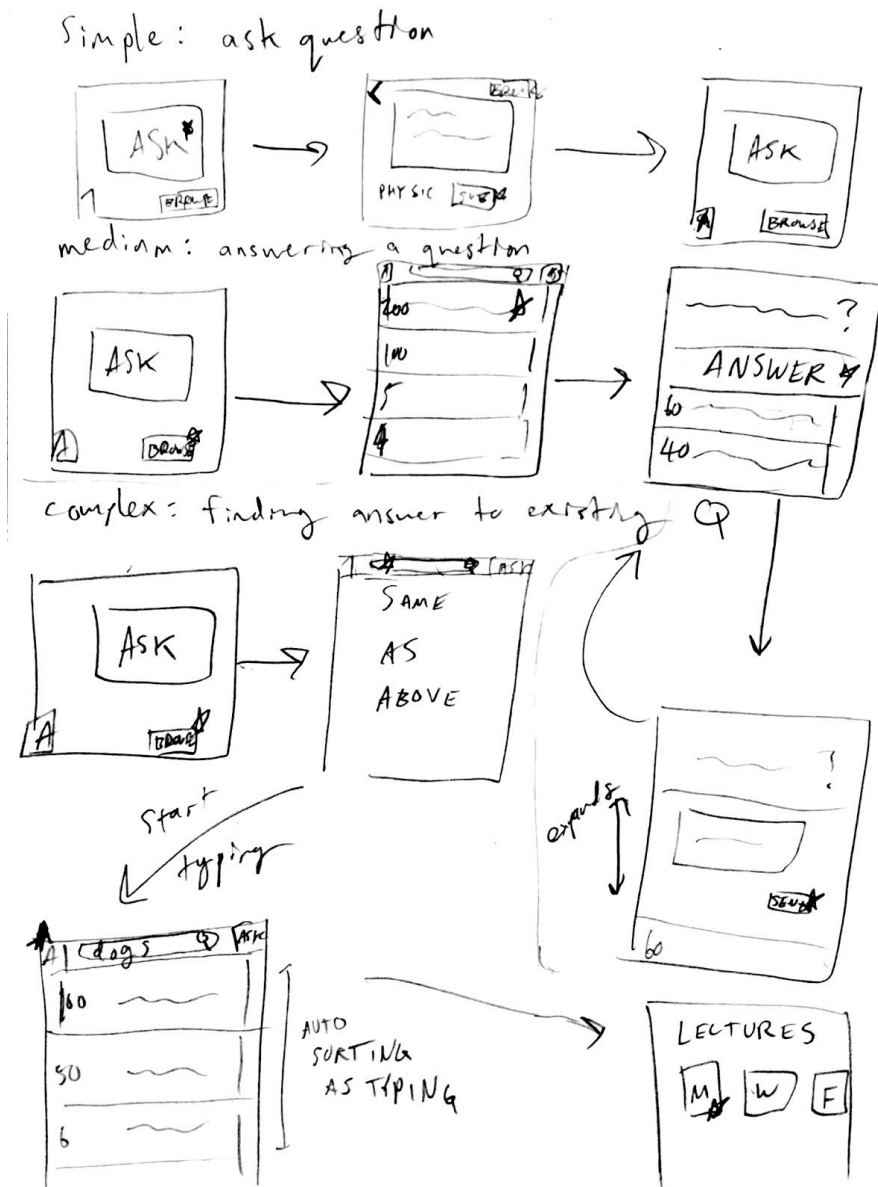


Figure 2. UI design pick 1: mobile app

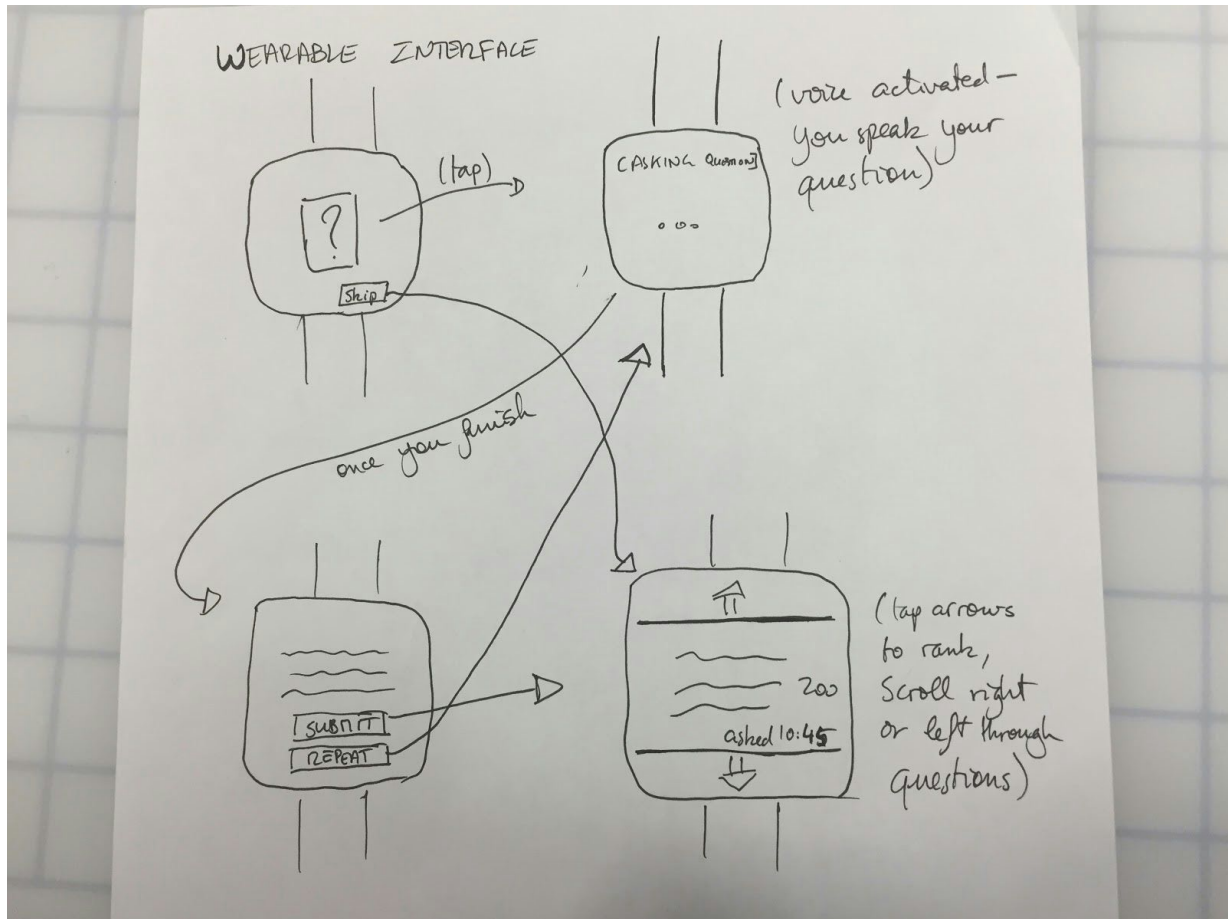


Figure 3. Design Pick 2: Wearable App Interface

Selected Interface Design

We picked the mobile app UI as our final interface design because almost every college student has a smartphone while smartwatches are still fairly uncommon. Also, due to the limited surface area of watches, tasks need to be done through voice input, which could be disrupting in lectures.

Table 1. Summary of functionalities

<p>Simple Task: Ask a question</p>	<p>Touch HELP button to open text input. Touch SUBMIT to return to initial screen.</p>
<p>Medium Task: Answer a question</p>	<p>Touch BROWSE to look at questions.</p> <p>Tap on questions to go to the next screen with the full text of the question and the GIVE HELP button.</p>

	Touch on GIVE HELP button to type in answer. Touch GIVE HELP again to submit answer.
Complex Task: Find a previously asked question	Touch ARCHIVE button to go to archives of questions organized by lectures. Touch BROWSE button to go to the question lists for the current lecture, then type key words in SEARCH BAR to search questions

Simple task: ask a question.

In this task, a student gets confused in lecture and wants to submit a question anonymously. He/she taps on the icon of the app, which brings him/her into the home screen of the app. He/she then taps the big “Help” button on the screen, which will takes him/her to a screen with a text box. He/she can then type in the question, and hits submit button. Once he/she hits submit, he/she will be taken back to the home screen

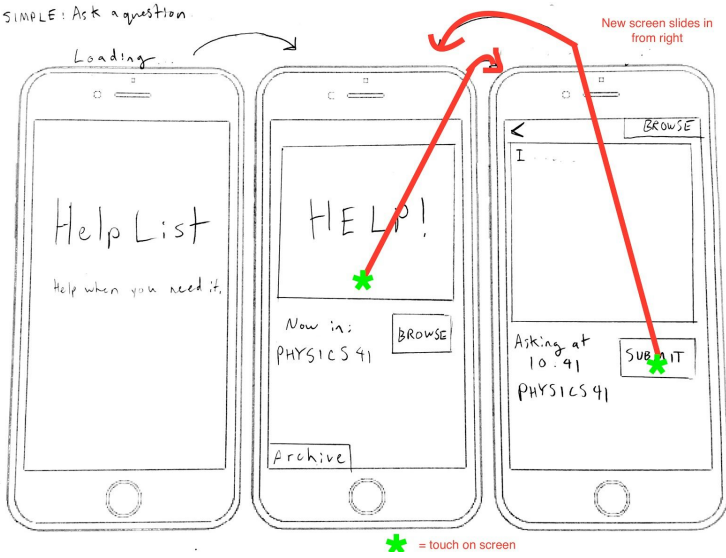


Figure 4. Storyboard for the simple task, touch represented by green star (only one touch at a time)

Medium task: Answer a question

In this task, a student or teacher is going to answer questions that have been asked. They will touch the “browse” button, which will take them to a screen with the questions asked listing in the descending order of the number of votes that the questions have received. They can then tap on the question that they want to answer. The question will then enlarge to show the full text, and other people’s answers will also appear. They can then tap “Give help”, which will pull down a text box for them to enter their answer. They can then tap “Submit” to submit their answer.

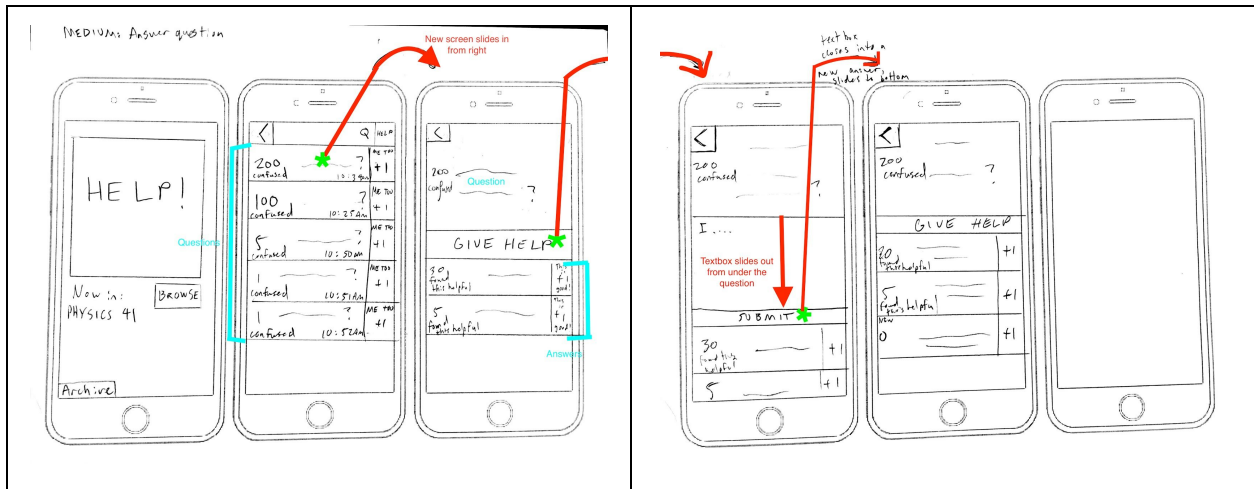


Figure 5. Story board for medium task

Complex task: Find an existing question

A student can look for an existing question regarding the most recent lecture by clicking “browse” on the home screen, which will take them to a screen with a list of asked questions. They can type keywords in the search bar to filter questions containing those keywords. If the question couldn't be found, they can tap back to go back to the home screen, where they can click on “Archive” to go to a screen where one can select previous lectures. They can then click on each lecture to browse the questions asked in that specific lecture. Alternatively, they can click “search all” to search among all lectures.

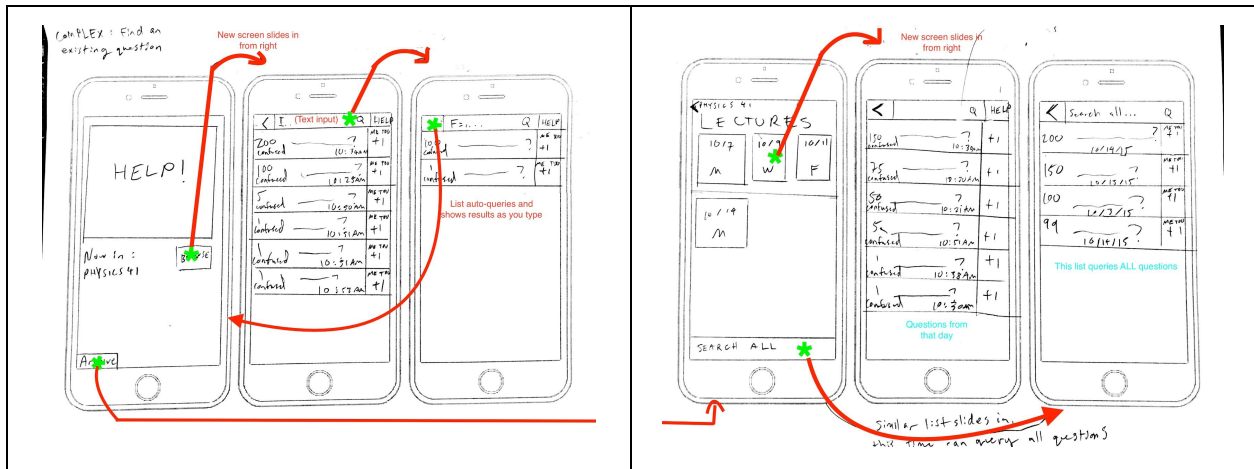


Figure 6. Storyboard for the complex task

Low-Fi Prototype

Prototype components for 1st Task: ask a question in class.

When User click on “Help” on the first paper, they will be presented the 2nd paper. When they try to type in their question, they will be presented a sticky note where they can write their question on. And the sticky note will be paste on the paper screen (see Figure 7). And they can then tap submit to submit their question.

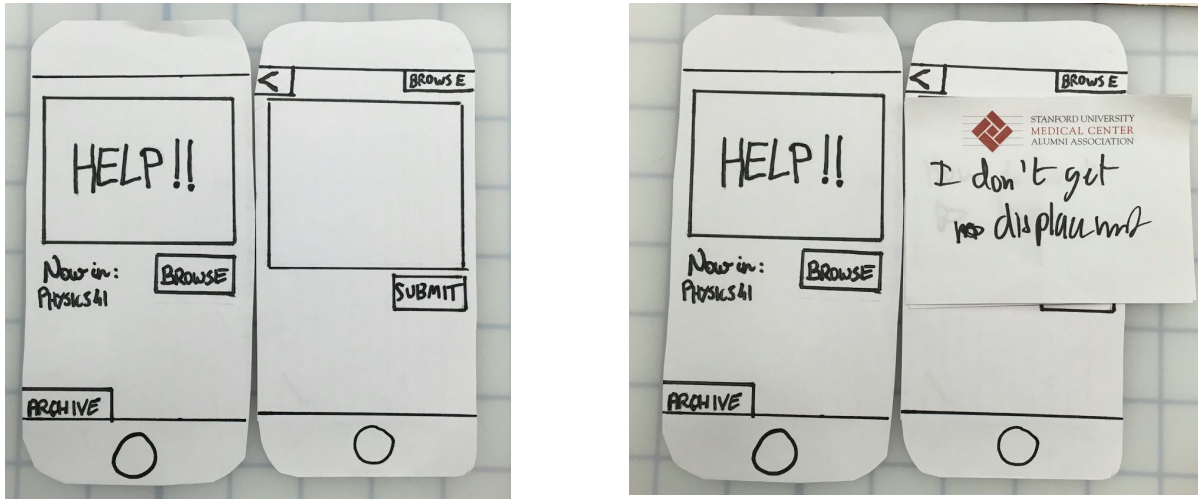
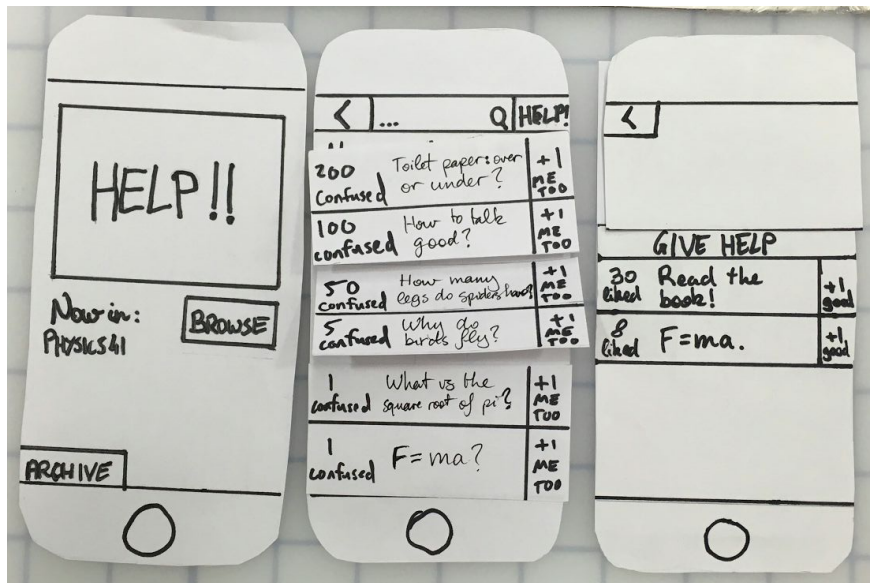


Figure 7 Prototype components for Task 1 Ask a question

Prototype components for Task 2: Answer a question

We made a cascaded question list, which can be changed based on the participant and the lecture. When user taps on “Browse”, they will be presented with the 2nd paper screen with the cascaded question list on it. When user taps on a question, they will be presented with the 3rd paper screen with the full text of the question. When the user taps “Give help”, the “computer” will then pull down the originally folded paper to display the the text box where the user can enter their answer.



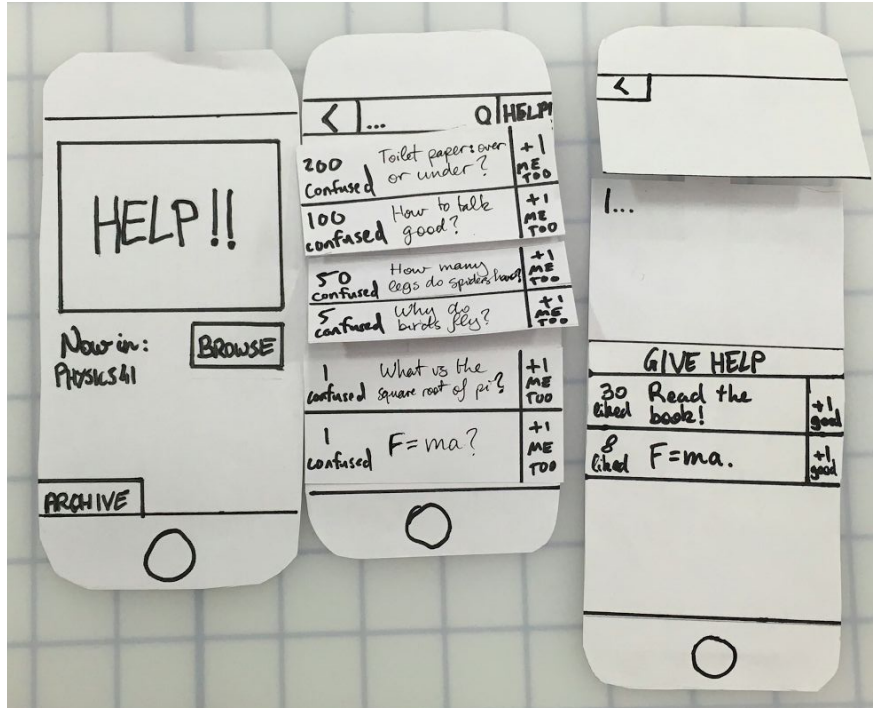


Figure 8. Prototype components for Task 2: Answer a question

Prototype components for Task 3: Find a question

On the home screen, the user tap on “Archive”, and they will be presented with the paper with lectures listed on it. They can they tap on one of the lectures to see the questions for the selected lecture or they can search all.

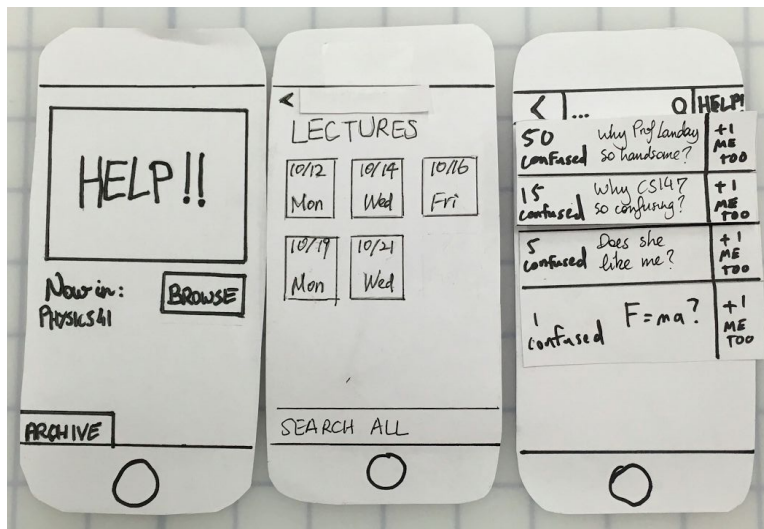


Figure 9. Prototype components for Task 3: find a question

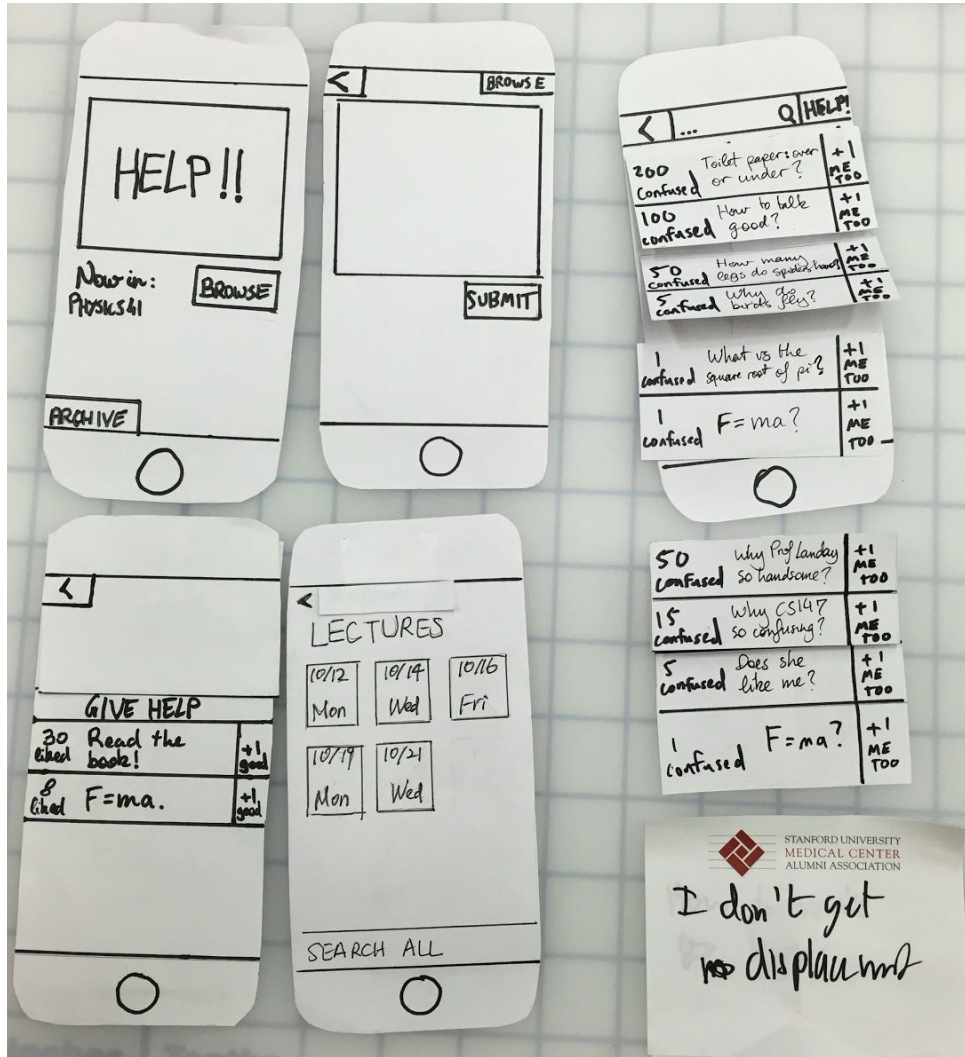


Figure 10. Low-Fi Prototype: all components

Method

Participants:

We went to two big lectures on campus, and asked for volunteers before the class started. The volunteers agreed to help with the testing without compensation. They are all undergraduate students. We asked them to sign the consent form to protect their identities.

Environment:

The testings were all conducted during lectures, where no/few questions were to be asked aloud.

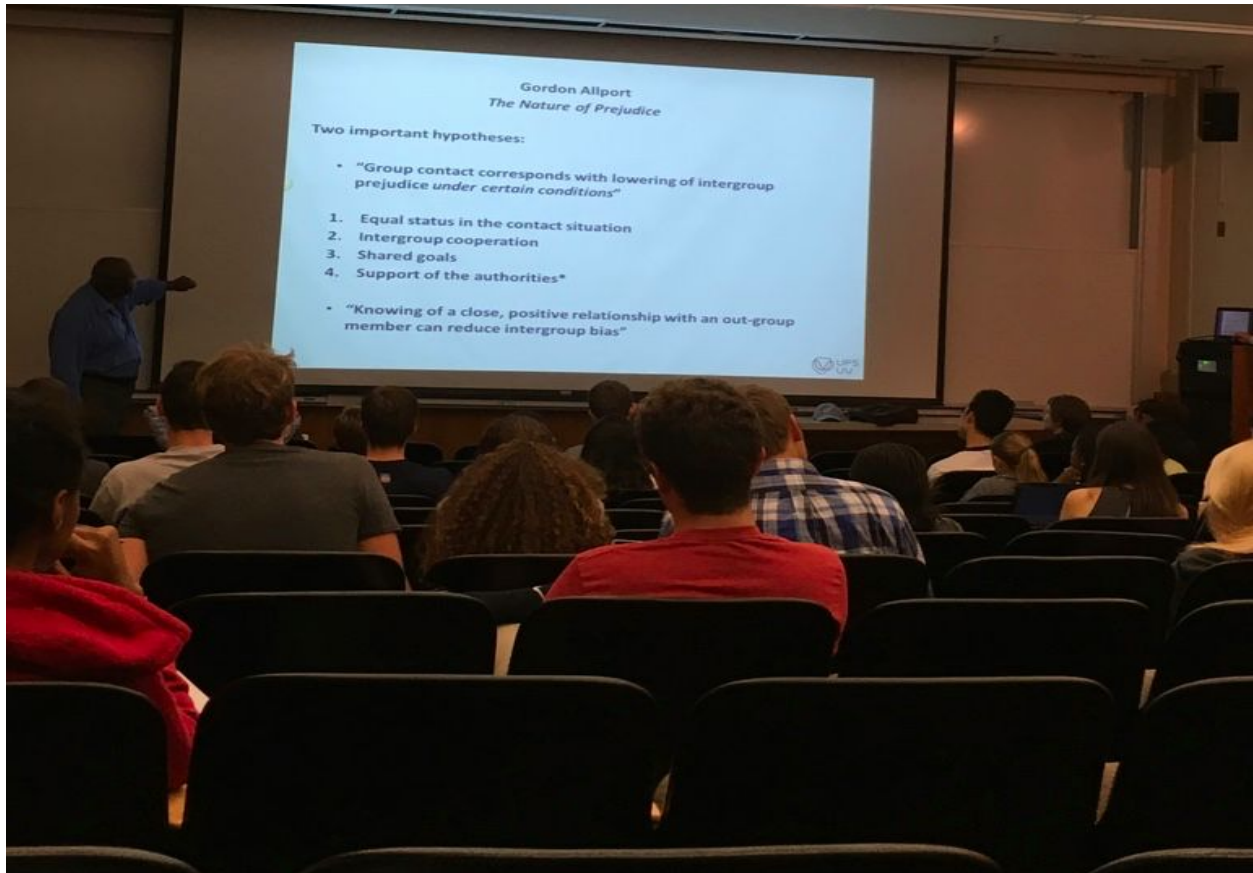


Figure 11. One of the lectures where the testing was conducted

Tasks:

We asked them to perform all the three tasks (ask a question, answer a question, find a question) in the lecture.

Testing procedure:

The full testing script is in Appendix. We first introduced ourselves, what we were trying to do, a brief overview of our mobile app. And then the facilitator asked the participants to perform the three tasks in sequence, and the “computer” switched the paper screens according to the participant’s reactions. After one task is finished, the “facilitator” would give the participant a thumb up to indicate that the task is finished. And after the testing is done, the facilitator would ask the participant for his/her feedback.

Roles of each team member:

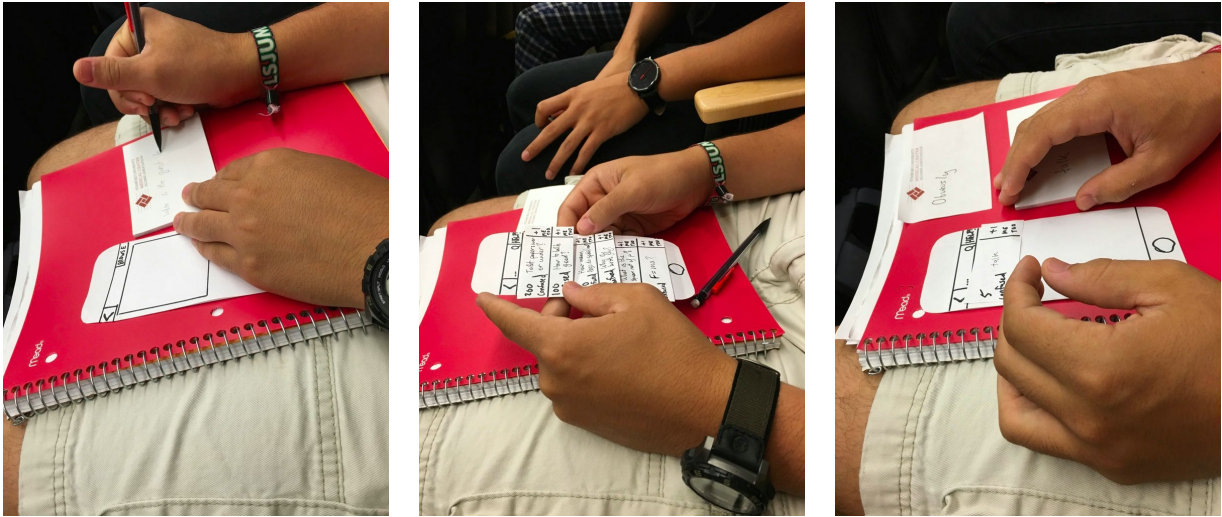
Santos acted as the computer, Andres was the facilitator, and Minmin was the observer.

Test measures:

In the testing, we were observing the participants’ speed and how they interact with our app while listening to the lecturer.

Results

Participant 1:

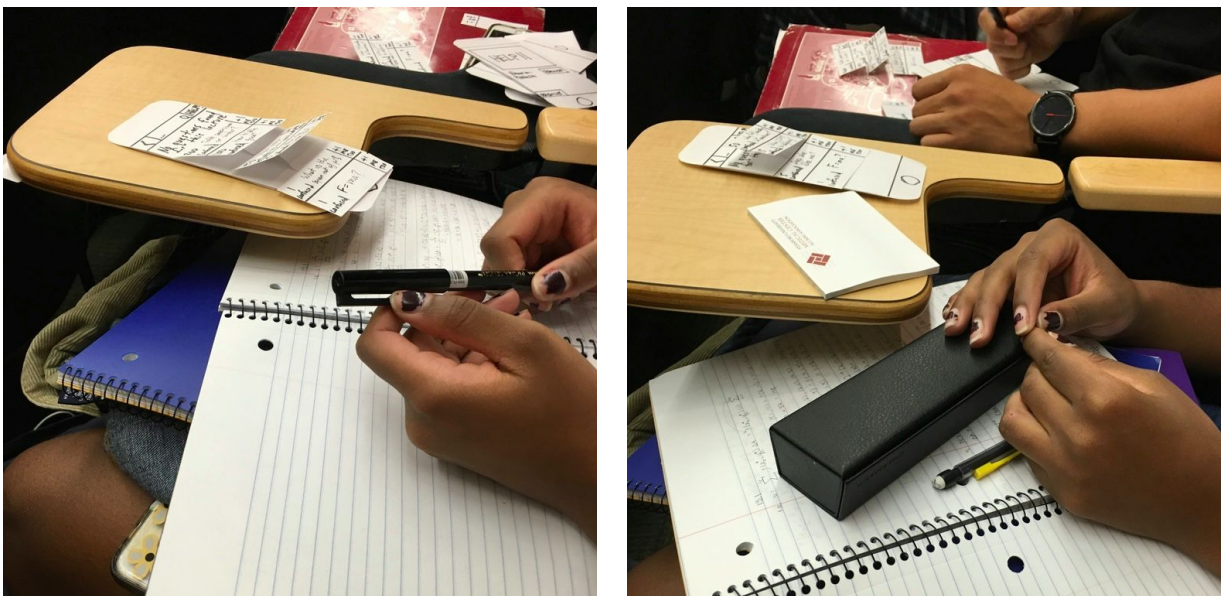


Task 1 was very easy for him. He quickly figured out how to use the interface to ask a question -- immediately touched the help button.

Task 2: He was quick to figure out he needs to tap "Browse" on the 1st screen, and then he directly tap on one of the questions.

Task 3: he clicked on "back" several times, and he went through most of the lectures in "Archive", and then finally clicked on search all and found the question he was looking for. This is a minor usability problem (level 2) because in the end he managed to finish the task. From this incident, we realize that we need to make the search process easier.

Participant 2:

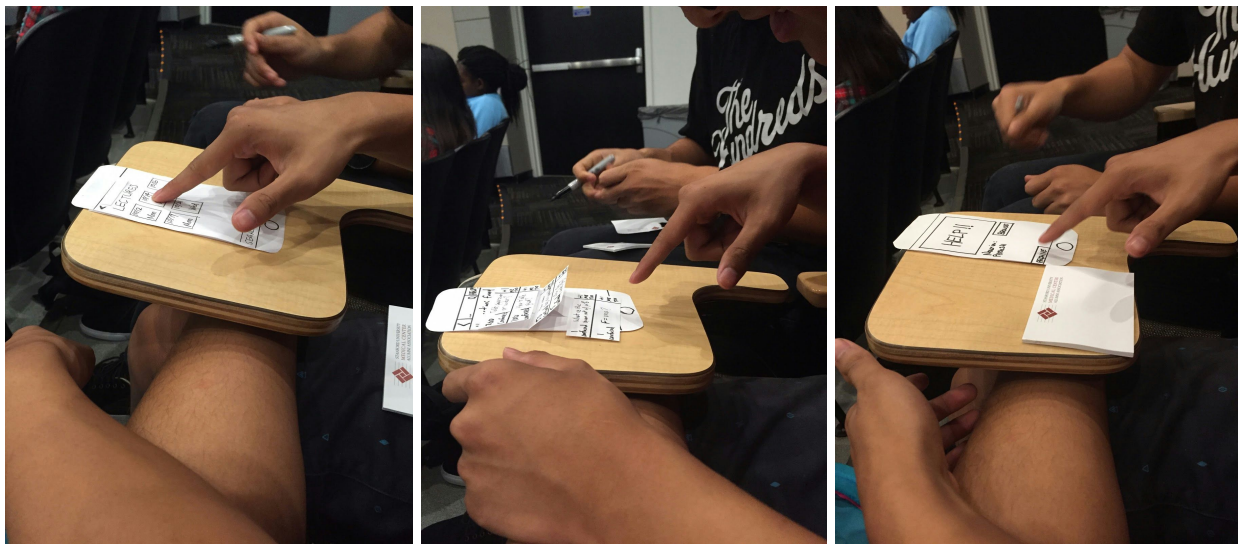


Task 1: She tapped on “Browse” instead of “Help” -- this was not a major usability problem (level 2), because she hit the smaller help button on the next screen, but later said she didn't realize the big help button the first screen was a button due to the low-fidelity nature of the prototype. This is a minor usability problem (level 2).

Task 2: She tapped on “Browse” as we designed, and it was easy for her to understand that she needs to tap on the question and enter her answer. However, she didn't realize at first how to submit the question (by tapping "give help" again) making it a minor usability problem (level 2).

Task 3: She did not notice the “Archive” button, and she went back and forth several times but found it eventually and used the search function/archives of questions very easily (level 2).

Participant 3:



Task 1: No problems.

Task 2: No problems.

Task 3: No problems. But he mentioned that the search process has too many steps.

The tasks seemed most intuitive for him: he breezed through them and did not seem to struggle with anything.

Discussions

Problems:

We intentionally wanted to test the “search all” function when the participants were searching through the archives, so we forced that by not showing them the results they expected when they simply searched through one old lecture. However, from this we found out that we need to make “search all” more obvious and/or optimize the search process, especially in the settings of lectures. The search process right now is a bit too complicated to use while listening to lecture. This would be a minor usability problem (level 2), and we think that the way to solve it would be to make the process more intuitive and/or make the “search all” bar more obvious, by making it a bright color.

Two participants separately had trouble initially finding the search bar on the browse screen. This is most likely because it is smaller and tucked away on the top of the screen, and not immediately visible to them. We had to point out that there was a search bar there, and then they understood how to perform the tasks. This would be a minor usability problem (level 2). We do not really want to make the search bar more prominent, however, because it would distract from the rest of the screen (the questions themselves), so a way to fix this would be to add grey text inside the search bar that reads “search” and that immediately disappears when you click on it.

Also, two of the participants had trouble realizing that once you’ve typed an answer in the “give help” section, you have to tap “give help” again to record your answer to the question. We are only labeling this a cosmetic problem because we foresee an easy fix: once you tap on the “give help” button, and the dialogue box appears, the “give help” text will change to “submit,” so it’s obvious that you have to click it again to send in your answer.

Finally, all three of the participants did not immediately see the “Archive” button, to search through older lectures. This would qualify as a minor usability issue (level 2), because to fix it we would have to make that button more prominent, or add an archive button on other screens to make the process of searching through old lectures more intuitive.

Good parts:

Two users found Help button very intuitive. The “+1” and the number of how many people are confused were intuitive to the participants. They liked the fact that they can like and rank the questions, giving them an active role in the application.

Overall, we did not have any major usability problems and the minor usability problems could be fixed by making certain buttons more visible.

Appendix

Test script:

We are conducting an experiment to evaluate our user interface for our mobile application, HelpList. Santos will act as the phone, he will be switching the paper screens based on your interactions with the prototype. And he will be silent throughout the testing. If there is some lag in computer time, please be patient, as this is our first round of testing. Minmin will be the observer, she will be taking notes of what happens during the testing, including what you said and your reactions. Your personal info will be kept confidential, and we have this form for you to sign.

What is our app? Our app is called HelpList. It is a mobile app where you can post questions anonymously in class when you get confused and need help. Other students can up-vote a question if they are confused with it too. And the most popular question will appear on the top of the list. You can also help other students by answering their questions. You can also search questions to see if other students have asked that question and if the question has been answered yet.

The app will know which lecture you are in based on the location and time info, and will automatically sign you into the lecture, so you don't need to worry about signing in during lectures.

Task Description #1: I will now ask you to submit a question in the lecture. (user attempts to perform first task).

Task Description #2: At this point I'd like you to help other students by answering one of their questions.

Task Description #3: Now suppose you have a question in mind, and you want to see if it has been asked or not. I would like you to do the search in the app.

Consent forms

Consent Form

The HelpList application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of HelpList. Data will be collected by interview, observation and questionnaire.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Santos, Andres, Minmin) or with Professor James Landay, the instructor of CS 147:

James A. Landay
CS Department
Stanford University
650-498-8215

landay at cs.stanford.edu

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the [TEAM NAME HERE] experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time

Name Alex Ramsey

Participant Number #1

Date 10/22/15

Signature Alex Ramsey

Witness name _____

Witness signature _____

Consent Form

The HelpList application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of HelpList. Data will be collected by interview, observation and questionnaire.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Santos, Andres, Minmin) or with Professor James Landay, the instructor of CS 147:

James A. Landay
CS Department
Stanford University
650-498-8215
landay at cs.stanford.edu

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the [TEAM NAME HERE] experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time

Name Thu My

Participant Number _____

Date 10/22/15

Signature [Signature]

Witness name _____

Witness signature _____

Consent Form

The HelpList application is being produced as part of the coursework for Computer Science course CS 147 at Stanford University. Participants in experimental evaluation of the application provide data that is used to evaluate and modify the interface of HelpList. Data will be collected by interview, observation and questionnaire.

Participation in this experiment is voluntary. Participants may withdraw themselves and their data at any time without fear of consequences. Concerns about the experiment may be discussed with the researchers (Santos, Andres, Minmin) or with Professor James Landay, the instructor of CS 147:

James A. Landay
CS Department
Stanford University
650-498-8215

landay at cs.stanford.edu

Participant anonymity will be provided by the separate storage of names from data. Data will only be identified by participant number. No identifying information about the participants will be available to anyone except the student researchers and their supervisors/teaching staff.

I hereby acknowledge that I have been given an opportunity to ask questions about the nature of the experiment and my participation in it. I give my consent to have data collected on my behavior and opinions in relation to the [TEAM NAME HERE] experiment. I also give permission for images/video of me using the application to be used in presentations or publications as long as I am not personally identifiable in the images/video. I understand I may withdraw my permission at any time

Name Charlotta Reed

Participant Number _____

Date _____

Signature Charlotta Reed

Witness name _____

Witness signature _____