

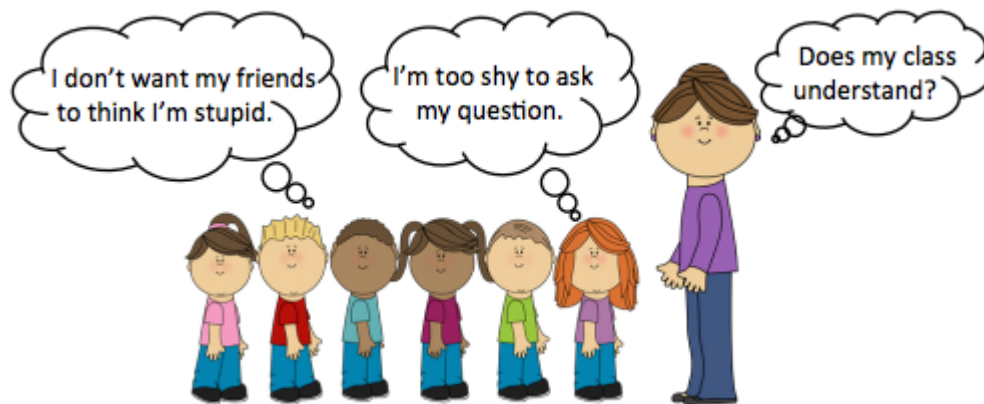
clicked

See the Questions Behind Their Masks

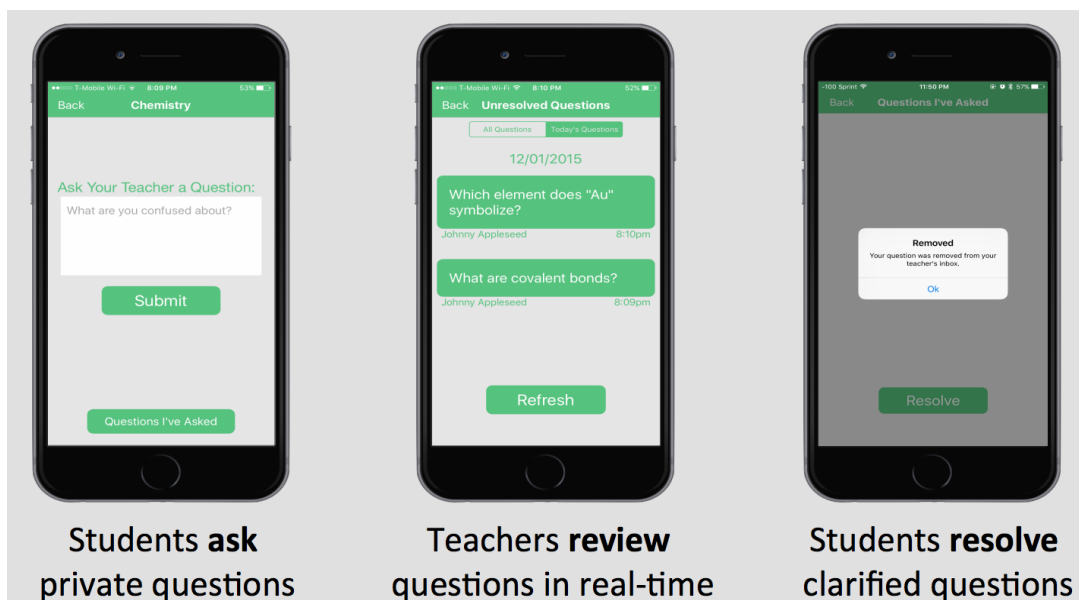
Gordon Dean (Product Management), Jake McKinnon (Product Design), Krister Johnson (Development), Michael Kim (Community Outreach)

Problem and Solution Overview

Problem: Students don't want to publicly admit that they don't understand class material, so teachers don't have a reliable way to determine their students' comprehension until it's too late (e.g. at the end-of-unit test).



Solution: Remove this barrier by creating direct communication channels between students and teachers, enabling teachers to revisit topics that students don't understand either during or after class and to revise their lesson plans based on the feedback implicit in this communication.



Tasks & Final Interface Scenarios

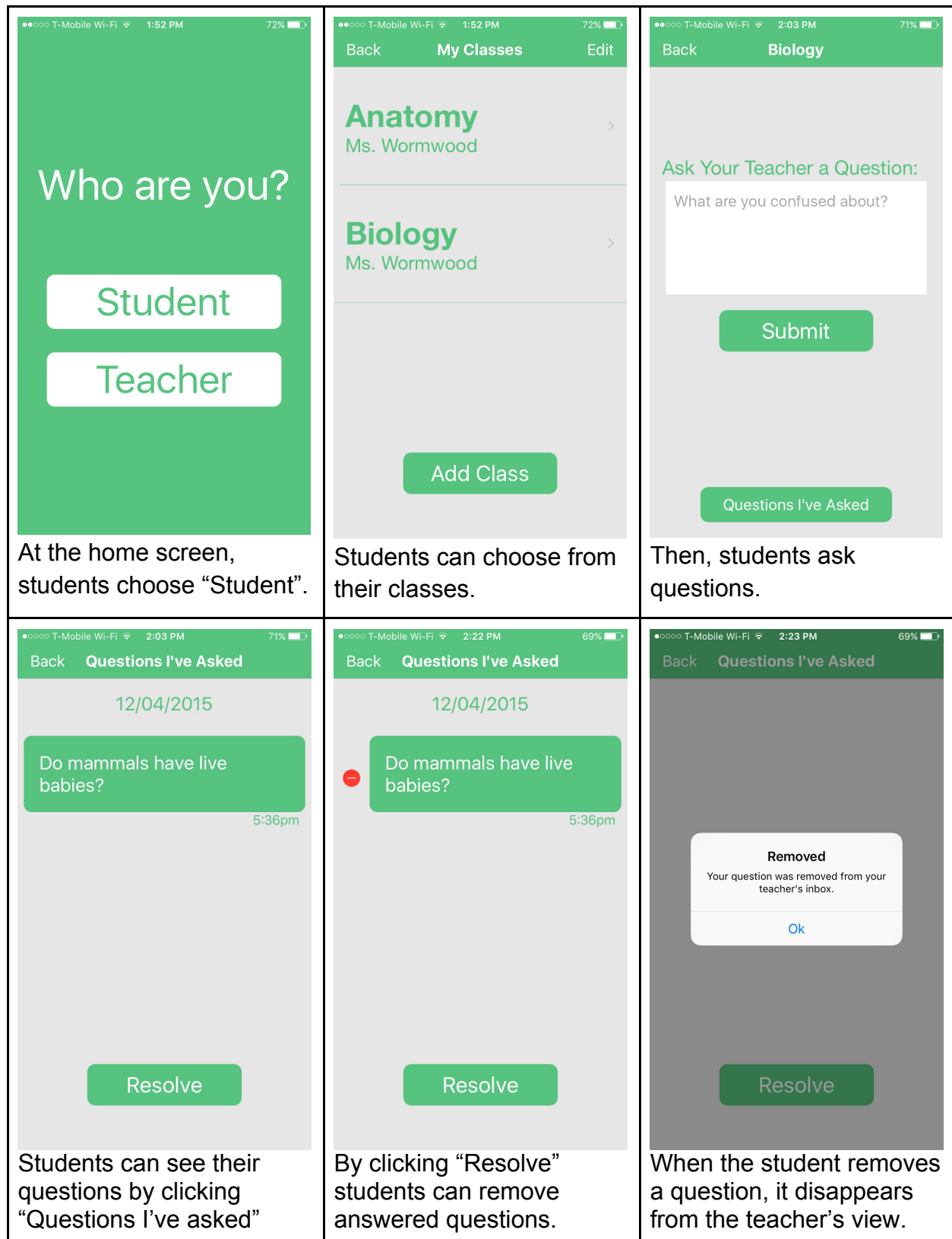
Simple-Complexity Task: Students need to ask questions when they are confused.

We chose this task - which is arguably the most important task and our core insight - because it came up repeatedly in our needfinding talking to students and especially teachers. Students often mentioned that when they didn't know topics especially well, they would just shrug it off because they didn't feel like being "that kid" who holds up the class or doesn't understand something they should. But more significantly, teachers knew that there were many students who weren't understanding the topics or who weren't engaged, and the teachers wanted to help. But it was hard to help them without knowing who they were or especially what they were confused about - particularly when they have the responsibility of doing this for over 100 students every semester.

Our solution to this task gives students a clear means of resolving their confusion, with a very low barrier to entry (previously, the perceived social cost and extra effort required to ask teachers during work time or after class prevented many students from resolving confusion). For a similar situation, where removal of small perceived barriers boosts student participation, compare usage of piazza to usage of email. Even though they both seem relatively low effort, piazza has utilization maybe 10x email. We hope this will perform a similar role.

Note: We altered this task to specifically reference "student questions." Originally, we had leaned towards emphasizing the speed of student responses or additional context. However, based on talking to students and teachers, they wanted specific questions answered.

Simple Task Storyboard:

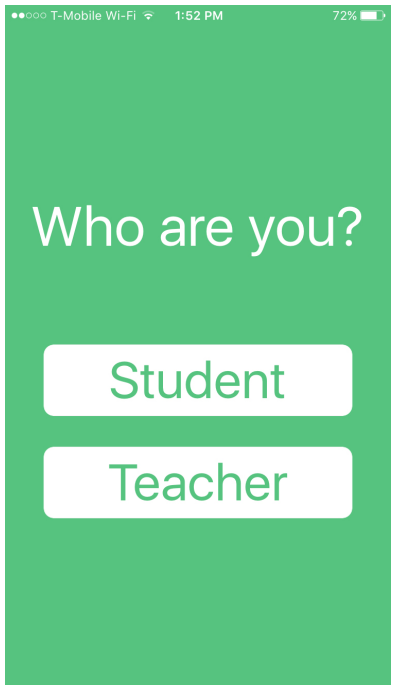
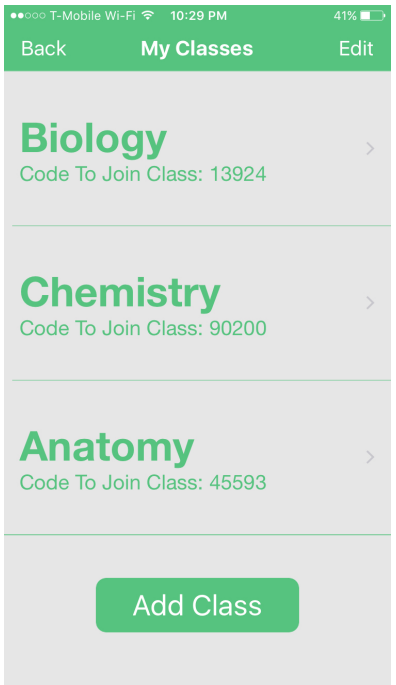
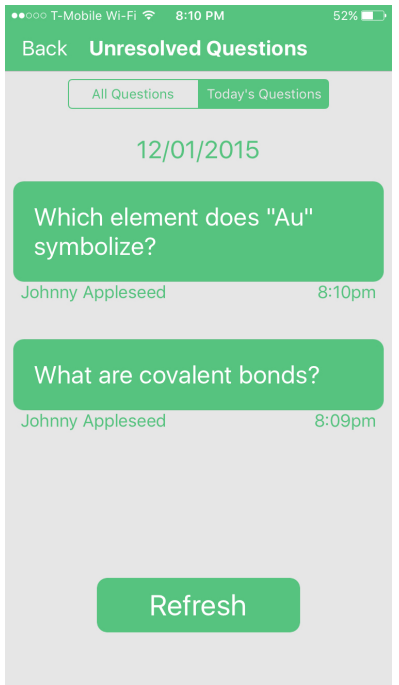


Moderate-Complexity Task: Teachers need to effectively judge whether or not students are understanding the current lesson while in class.

This task was chosen largely as corollary to the aforementioned task, from the teacher's perspective. Nearly every teacher said they struggled to judge whether students were 1) understanding and 2) engaged. Though all of them said they tried to use whatever coarse-grained signals they had (when they really lost everyone, they would know), if their best students were understanding everything, often everyone else would just follow along even if a large portion of them were confused.

Our solution for this task fell out of what teachers said they already try to do: use coarse-grained signals. By making it much easier for students to give feedback, we hope to give teachers a stronger signal with which to “read” the class and judge student understanding, by essentially giving them a newsfeed of students' thoughts about class material.

Moderate Task Storyboard:

 <p>The home screen has a green background. At the top, it says 'Who are you?'. Below this are two white buttons with green text: 'Student' and 'Teacher'.</p>	 <p>The 'My Classes' screen has a green header with 'Back', 'My Classes', and 'Edit'. Below is a list of three classes: 'Biology' (Code To Join Class: 13924), 'Chemistry' (Code To Join Class: 90200), and 'Anatomy' (Code To Join Class: 45593). At the bottom is a green button labeled 'Add Class'.</p>	 <p>The 'Unresolved Questions' screen has a green header with 'Back' and 'Unresolved Questions'. Below is a tab bar with 'All Questions' and 'Today's Questions'. The date '12/01/2015' is displayed. There are two green boxes containing questions: 'Which element does "Au" symbolize?' and 'What are covalent bonds?'. Each question is attributed to 'Johnny Appleseed' with a timestamp (8:10pm and 8:09pm respectively). At the bottom is a green button labeled 'Refresh'.</p>
<p>At the home screen, teachers choose “Teacher”</p>	<p>They are taken to their classes, which have codes to let new students join. Choose the class that would like to view the questions for.</p>	<p>For each class, teachers can see “Today’s Questions” to see how the class is doing.</p>

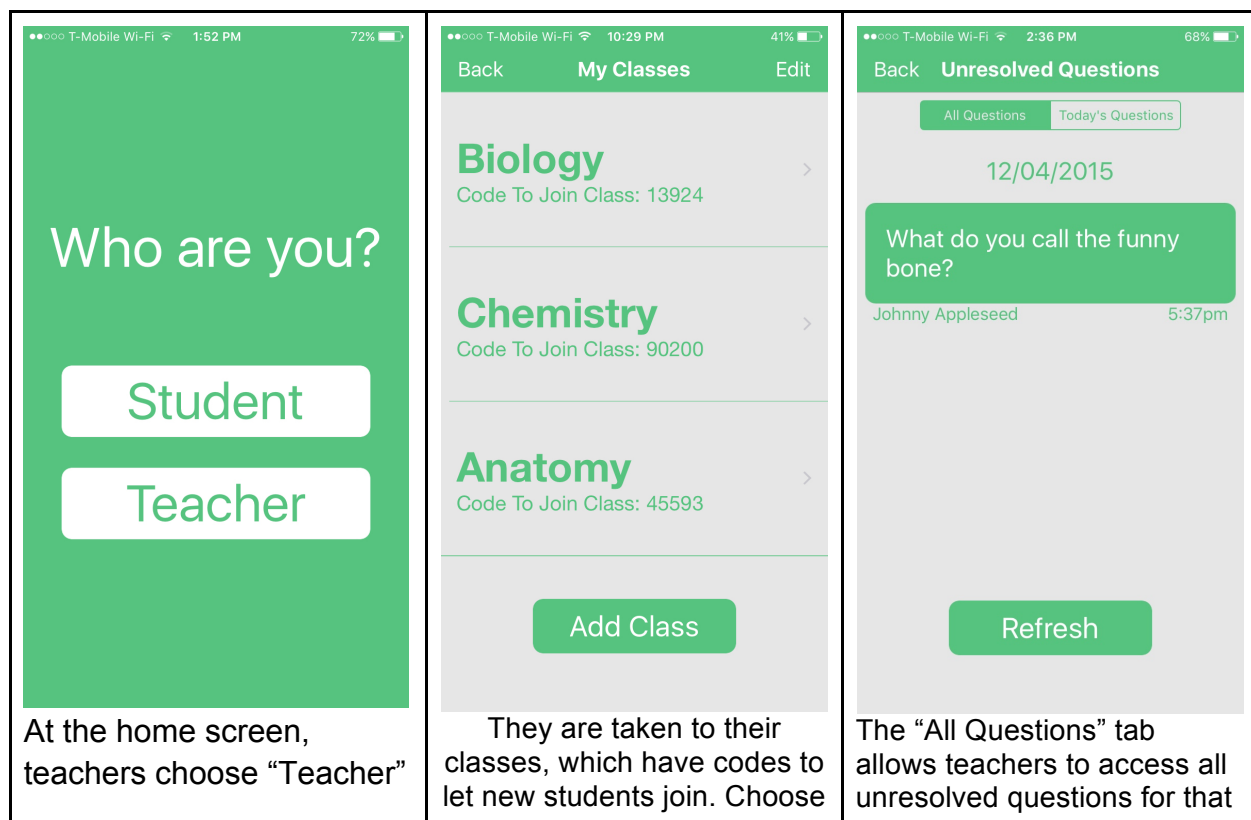
High-Complexity Task: Teachers need to address unresolved questions after class or in the next lesson.

In order to address unresolved questions after class, teachers can either follow-up privately with students or update future lessons. It's hard to pinpoint which students understand which questions, so teachers generally struggle to give this level of personal attention. We found that teachers are constantly looking for data to inspire lesson changes that better suit their students and that they don't want to single out struggling students in class.

Our solution here - which we will further develop in cs194H - is to give teachers a list of specific questions attached to individual students that teachers can use to appropriately follow-up. Teachers can look at "Today's Questions" and reach out to those students privately to try and immediately improve their understanding. The "All Questions" tab enables teachers to look back over the whole history of their class at all unresolved questions that they can use to make adjustments for future lessons. We plan on taking our app to the next level by enabling teachers to directly message students through the app to facilitate these conversations.

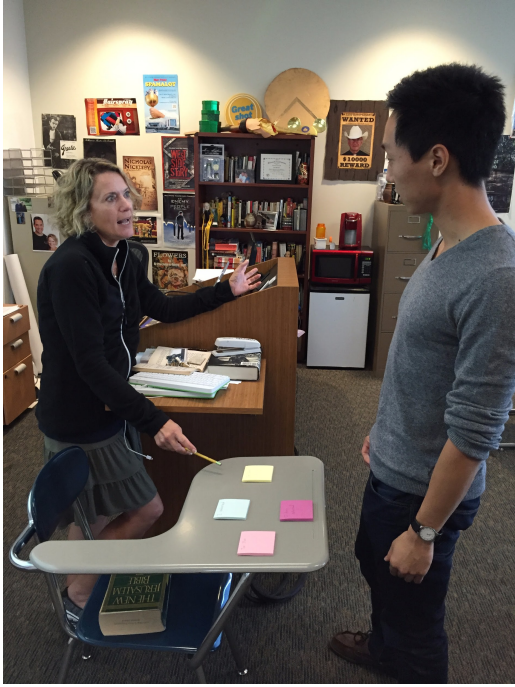
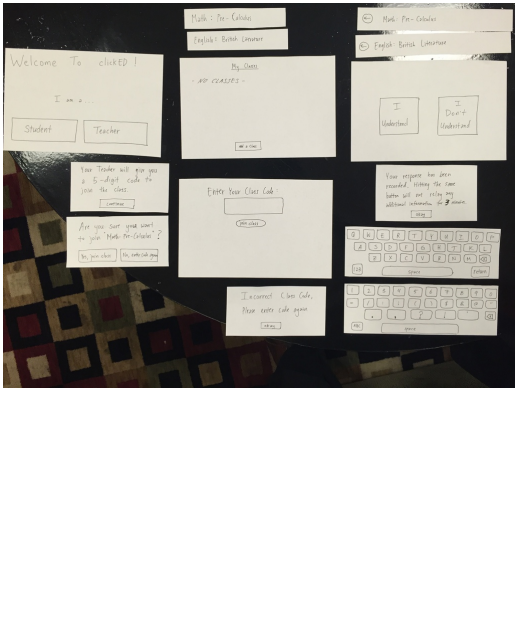
Note: We made this task more general because updating lesson plans is not the only way to use the data that we are providing. Our task was always focused on helping teachers to address student confusion, and now we are stating that more plainly.

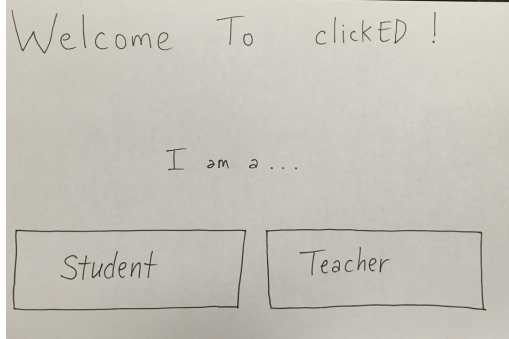
High Task Storyboard:



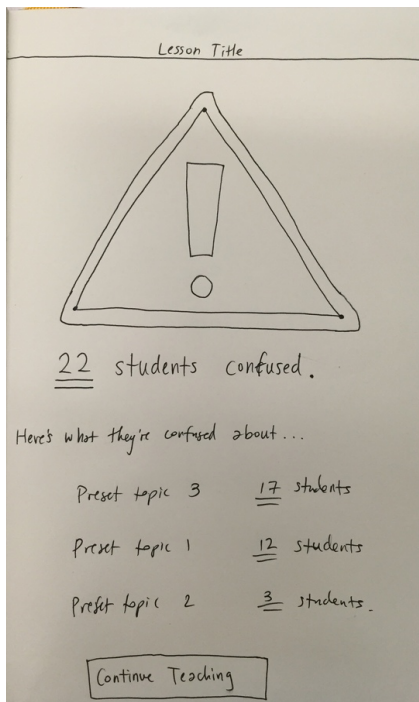
	the class that would like to view the questions for.	course in order to follow-up with students about specific problems or to update future lessons.
--	------------------------------------------------------	-------------------------------------------------------------------------------------------------

Design Evolution

	<p>Our most successful Experience Prototype modeled a system that gave teachers real time feedback from the class about their engagement and understanding.</p> <p>Key Takeaways:</p> <ul style="list-style-type: none"> • Frequent pinging is distracting to teachers. <ul style="list-style-type: none"> ◦ Solution: Use a threshold and only notify the teacher if enough students are confused. • Negative feedback can often be detrimental to a teacher's confidence <ul style="list-style-type: none"> ◦ Solution: Offer a "I Understand" button so students can affirm.
	<p>Our first paper prototype was done horizontally because we intended to develop the app for iPads that are used at Bellarmine.</p> <p>Key Takeaways:</p> <ul style="list-style-type: none"> • Designing for iPad was difficult as only 1 member in our group had an iPad. <ul style="list-style-type: none"> ◦ Solution: We decided to switch to iPhone but make the app responsive. • Data presented to teachers had to be sophisticated and ideally anonymous. <ul style="list-style-type: none"> ◦ Solution: We made it so the in class feedback was anonymous until the end of class, when the teacher could see which students had responded and follow up.



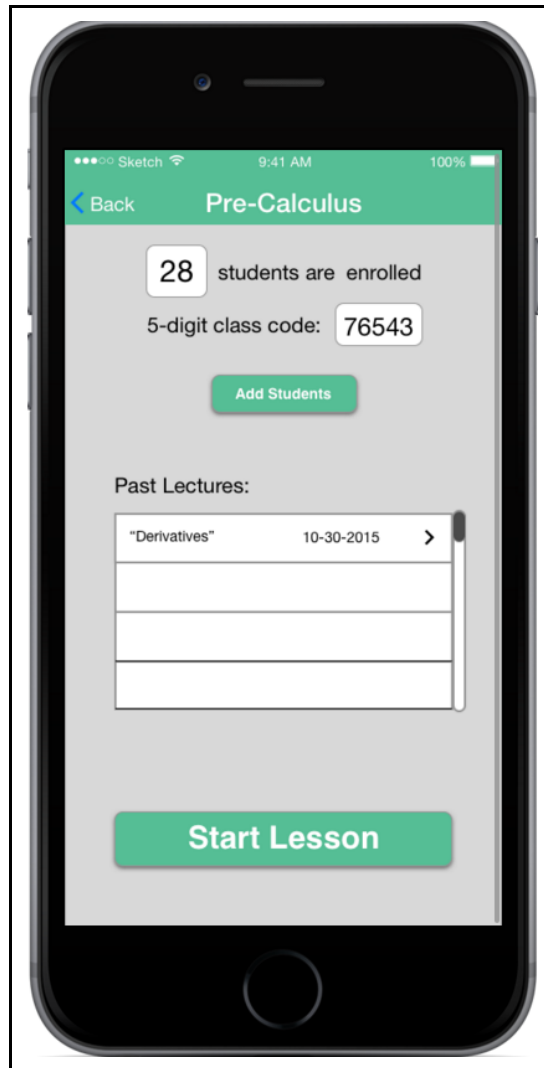
- The main screen was too simplified, to the point where it was sometimes unclear how to proceed
 - Solution: We added pop ups and tutorials to guide people through the app.



Our revised paper prototype included elements from the first, like having a “threshold exceeded” warning and giving students 2 buttons to signal understanding.

Key Takeaways:

- Teachers felt that just knowing the number of those who hit “I don’t understand” wasn’t very helpful. Students also wanted to specify their misunderstandings/confusions.
 - Solution: We decided to give teachers the option to input “Preset topics” so that students could specify what they were confused about.



Our medium-fi prototype was done on Marvel and made a lot of the visual elements more important. We chose to use a simple color scheme and keep all the screens relatively hassle free.

Key Takeaways:

- We used a lot of popups with the intention of guiding users through the app and reducing the number of screens to navigate, but sometimes the popups were excessive.
 - Solution: We modified a few of our popups into new screens.
- We considered adding a login option because it was brought to our attention that we'd need to know the identity of the person using the app.
 - We decided not to add that implementation for this class because it didn't actually add to the app's functionality - but it is high on our list of things to add as soon as we move from functionality fixes to real life testing.

Major Usability Problems Addressed

The majority of the feedback from our Heuristic Evaluation yielded major functionality changes that did not lend themselves well to simple visual change comparisons. There were also two severity 3 issues that overlapped with existing issues, so we removed them from the discussion to avoid redundancy.

1) [H2-1 Visibility of System Status] [Severity: 3]

“I Understand” button does not seem to do anything from the student perspective. No output is displayed when students tap “I Understand” and it seems that students would be less inclined to tap “I Understand” than “I’m Confused”. Provide feedback for users when they tap “I Understand” or consider removing the button.”

Solution:

We removed both the “I understand” and “I’m confused” buttons (Figure 1a). It didn’t make sense to keep the “I understand” button, and we couldn’t find a way to make the single button look good. We shifted towards students directly submitting questions as their only action during class (Figure 1b). We thought this would greatly simplify the interface and the student’s ability to complete the task.

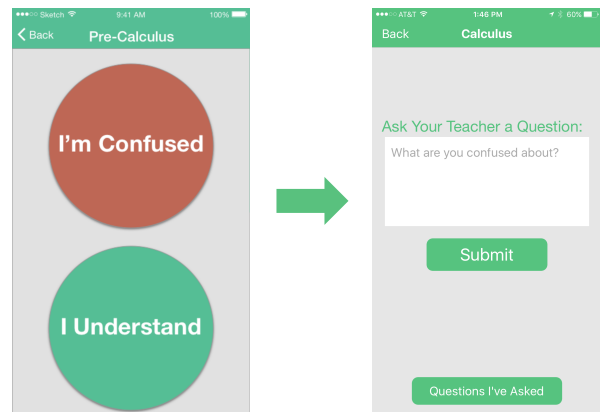


Figure 1a

Figure 1b

2) [H2-1: Visibility of System Status] [Severity: 3]

After you submit a question or confusion report, there is no acknowledgement of the receipt of this information, nor is there any sense of next steps. This could be something as simple as “Thanks for asking your question! The teacher will get to it soon.” This constitutes lack of visibility of status because this interface could be misconstrued as simply not working (ie the question just isn’t submitting, for instance).

Solution:

Before, students received no confirmation after submitting a response (Figure 2a), so we added a notification that provides confirmation upon submitting a question (Figure 2b). Because the teacher is not actively pinged with every incoming question, it would be easy for students to think that their questions hadn’t gone through. Any extra time spent re-typing the question is unnecessarily distracting for the students.

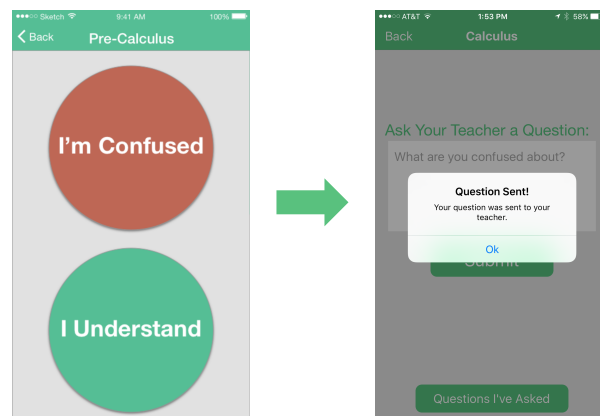


Figure 2a

Figure 2b

3) [H2-1: Visibility of System Status] [Severity: 3]

No record of questions that have already been asked exists in the student view as per this medium-fi prototype. Perhaps previously asked questions should be shown in the student view as well so that (1) students can know that their questions have been received and (2) duplicate questions can just be upvoted rather than repeated.

Solution:

Before, students submitted responses and had no way of tracking what they had asked (Figure 3a). We added a “Questions I’ve asked” section where students could reference and resolve outstanding questions (Figure 3b). We decided against allowing students to see other students’ questions because we feared they would abuse that ability and disrupt the class. We felt that preserving the private channel between students and teachers gave the users the most benefit.



Figure 3a

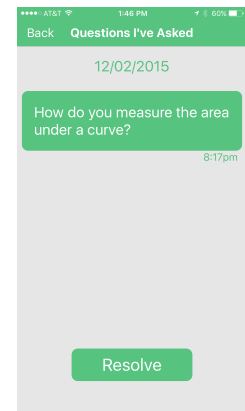


Figure 3b

4) [H2-3: User Control & Freedom] [Severity: 3]

Users (teachers) are confined to entering just 3 topics, which then limits the number of things that students can be confused about. In addition, if they add no topics, it is unclear what will be displayed for the student, and if they can even choose any topics. Could more fields be added? This also limits the format of questions asked – perhaps the “do you have any other questions” feature can be emphasized? This format should be highlighted.

Solution:

We experimented with enabling teachers to add topics before class that students could pick from when submitting responses (Figure 4a). This approach ended up limiting both students’ and teachers’ ability to submit and receive helpful feedback. We were also worried about adding extra work for teachers, which was a large concern from our initial need finding. We shifted towards students asking questions in order to give them the freedom to ask what they wanted (Figure 4b).

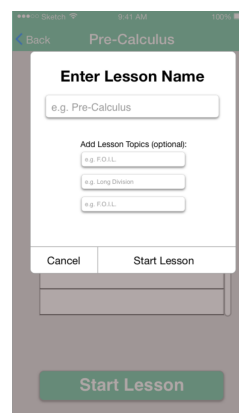


Figure 4a

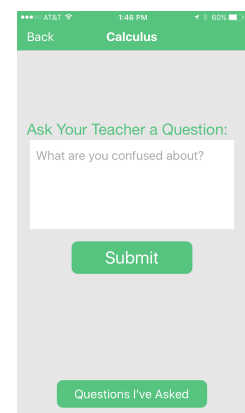


Figure 4b

5) [H2-5: Error Prevention] [Severity: 3]

On the “radar” screen, the back/end button is very accessible. Although there is a more info button that explains when to press the end button, there is no guarantee the teacher will click this button before clicking the end button. If they accidentally click the end button before finishing the lecture, there is no way to restart the lecture. Thus, I’d suggest having a small popup when the end button is clicked with wording such as, “This will end the current lecture. Are you sure you want to end?” The options could be “Continue Teaching” and “End Lecture” with an additional checkbox to “Don’t show me this message again.”

Solution:

We decided to remove class sessions completely from the app (Figure 5a). These only added complexity and wasted teacher’s time. We are also shifting the focus of the app from just an in-class app to work outside of class as well. The feed of unresolved questions organized by date works well to encapsulate both the in-class (“Today’s Questions” in Figure 5a), as well as resolving questions out of class (“All Questions” in Figure 5b).

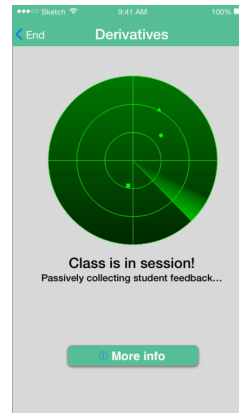


Figure 5a

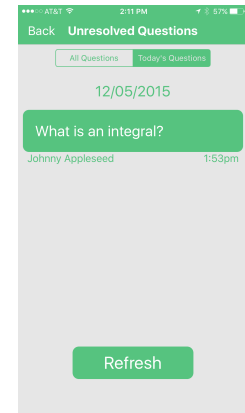


Figure 5b

6) [H2-7: Flexibility & Efficiency of Use] [Severity: 3]

The classroom setting is unpredictable and requires flexibility to be built into to any app that is used in that context. Teachers should have the option to pause lecture, tell the app when a concept is being repeated, or summon questions on the fly during the lecture if that is something they would like to do. All this could be facilitated by adding relevant buttons on the “lecturing now” screen.

Solution:

We tried to simplify the interface by limiting the app to passively poll in the background and alert the teacher when the threshold had been exceeded (Figure 6a). However, this forced the teacher to interrupt their class when the app said to and not during their natural pauses. We found that teachers won’t use software if it doesn’t work with their current style. We shifted to a feed of unresolved questions so that teachers could access them whenever they wanted (Figure 6b).

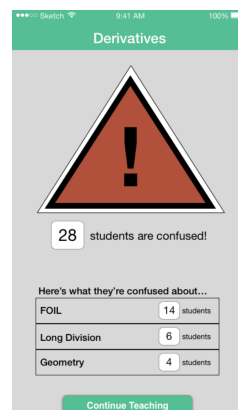


Figure 6a

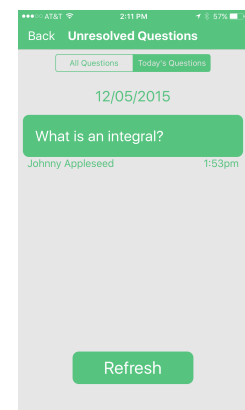


Figure 6b

7) [H2-8 Aesthetic & Minimalist design] [Severity: 3]

On the "What are you confused about?" popup, there are several different things going on at once. When I see one of these types of popups in a mobile app, I generally am expecting to click once, or perhaps type a small amount of text (i.e. entering a wifi network password on iOS). I'd suggesting limiting the popup to either subject selection or question entry, or just make this a separate page.

+1 for "Enter Lesson Name" popup in teacher mode.

Solution:

In an effort to quicken response submission for students without limiting their ability to express themselves, we had a very busy pop-up that ended up making the whole process more complex (Figure 7a). We simplified it down so that students only had one way to submit feedback (Figure 7b). By reducing the number of steps, we hope to gain back the time that students spend typing out their questions.

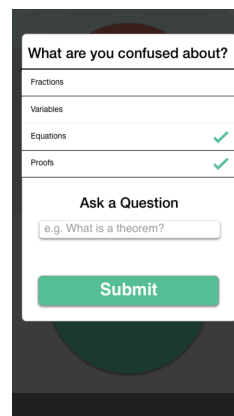


Figure 7a

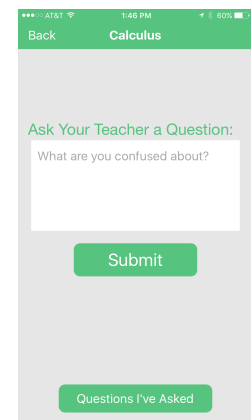


Figure 7b

8) Additional Fixes

In addition to the changes above, we made two additional improvements throughout the app that addressed many of the remaining issues regarding user's ability to recover from error and ease of navigation.

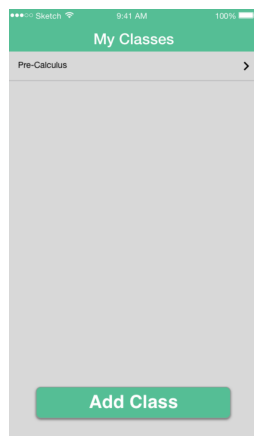


Figure 8a

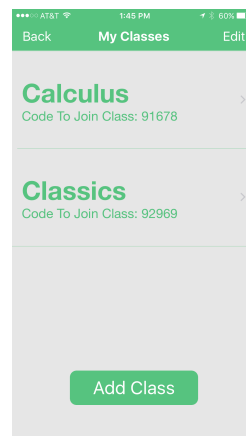


Figure 8b

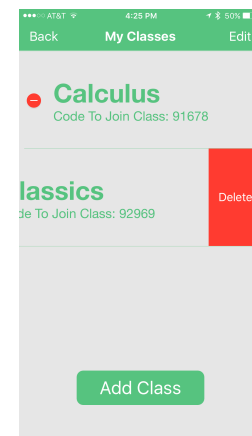


Figure 8c

Once a user had created or joined a class, there was no way to remove or edit that class (Figure 8a). Additionally, if you accidentally selected the wrong role, you couldn't go back. We added "Back" and "Edit" buttons for both students and teachers, so they could recover from

error or leave a class after it had ended (Figure 8c). We also made it so that if a teacher removes a class at the end of the year, the student is automatically un-enrolled. We also made all of the text and sections larger and more clickable so that it was easier to navigate using your fingers (Figure 8b).

Prototype Implementation

Tools (How you built the prototype)

To build our high fidelity prototype, we had a single team member use Xcode to write an iOS app in Swift. We used TestFlight to distribute builds to other team members for feedback, and we used Slack to communicate about design decisions.

How the tools helped

Xcode's "Storyboard" option allowed us to visualize the app as we created it. Apple's "TestFlight" option allowed us to immediately share builds of what we'd currently built to the entire team – all four of us could instantly download the up-to-date version of the app, without having to deal with Apple's App Store review process.

How the tools did not help

MYSQL is not easily implementable in Swift without third-party add-ons. As such, we could not easily create a central database for our app to communicate with. We opted to store our data locally, with a "student" view and a "teacher" view to simulate the experience of sending data between student and teacher.

Wizard of Oz (Any Wizard of Oz techniques required to make it work?)

We were not able to create a central MYSQL database. As such, our current version stores all data locally. This means that a tester can log into a student account, ask a question, and then log into a teacher account and see the question. However, this means that a tester cannot ask a question on one device, and then see it appear on another device.

Hard-coded Data (which pieces use are hard-coded data?)

We hard-coded a single student profile (Johnny Appleseed) and a single teacher profile (Ms. Wormwood). Users are automatically logged into either of these profiles when they choose between Student and Teacher from the home screen. We figured that it would be easier to demonstrate the functionality of the app without testers needing to constantly log out and log back into profiles.

What is missing and what might you add in the future?

In CS194H, we intend to create a MYSQL database so that a student can submit a question on one device, and a teacher can see that question appear on another device. We will also implement a login screen with individual user accounts, so that students can only see the questions that they ask, and teachers can see all questions submitted from their students. Finally, we will add the ability for a teacher to respond to students in the app. We imagine the following use-case: the student asks a question in the app during class; the teacher sees the question and verbally responds during class; after class, the student is still confused, so they send a follow-up question, and the teacher is able to respond to the follow-up question by text in the app.

Summary

Clicked is a tool for high school classrooms that opens a direct channel of communication between the teacher and student. It offers an embarrassment-free way to ask questions, and will give struggling students a chance to catch up. Teachers will also have valuable information about the efficacy of their teaching, and can use the information to adjust their lesson during and after class. Though we knew from the beginning that we wanted to facilitate communication, we shifted from "I understand / I don't understand" to specific questions. We learned that students were self-aware that any app that allows for student to student communication would be distracting in class. By only focusing on the teacher-student relationship, we sought to make it easier for teachers to manage their class and provide individualized attention. The design process allowed us to focus in on the key functionalities and integrate feedback from potential users into our work.