Contextual Inquiry and Task Analysis

Thuy Ny Le - Team Manager & Design
Chris Michael Ponce de Leon - Design & User Testing
Howon Lee - Development

Value Proposition

kindergarten will provide instructors with a simple and quick method for gathering student feedback and streamlining office hours.

Problem & Solution Overview

Often instructors set office hours and determine the course flow and workload at the beginning of the quarter. However, the workload for students and instructors varies from week to week depending on the difficulty of conceptual topic or assignment covered that week. Office hours, especially, can become a torture. Before deadlines, office hours are packed, the queues are long and students are unable to receive the personal help that they need. In other times, they are empty and TAs waste their time.

We found that the most pressing problem, therefore, is the lack of feedback between instructors and students on what is working for the students' understanding and what is not. **kindergarten** is on a quest to make the distribution of office hours and the quick review of concepts in college **as easy as kindergarten**. We want to facilitate student and instructor communication and gather feedback to improve the learning experience for everyone involved.

We are creating a mobile app to equalize the distribution of office hour attendance over the course of a quarter and and to ease the collection of in and out-of-class student feedback and short "clicker"-style quizzes directly on a phone. The quizzes will test whether students understand the concepts that professors try to present, office hours surveys will provide feedback about how much help students need for an assignment or concept, and polls will enable professors to learn about the disposition and learning goals of their students.

Task

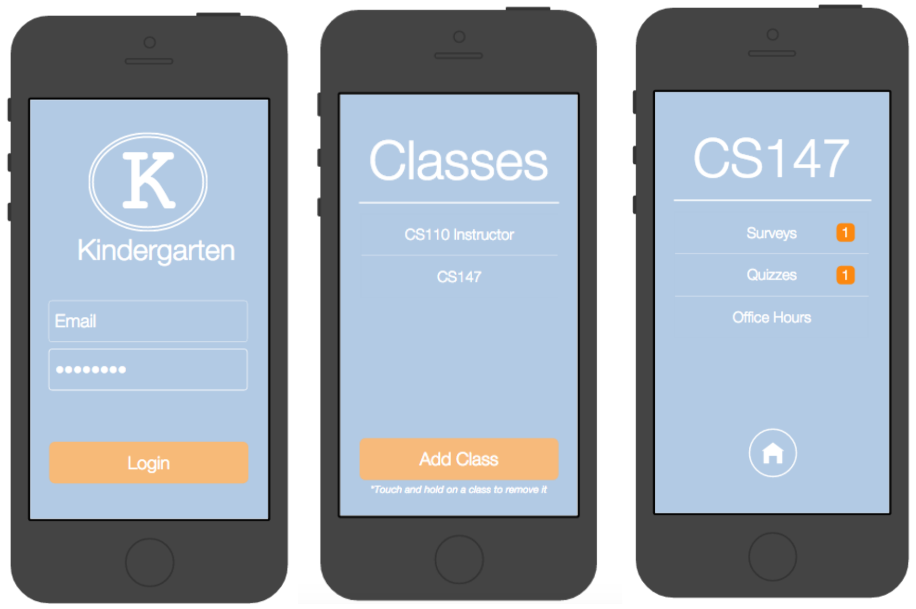
We imagine the user using Kindergarten through three tasks.

1. **The most simple task** is adding a class because this is something a student would do or an instructor would do by the first class: they would be informed by an instructor that the class will use Kindergarten, they will bring up the app, add the class to the class listing and then take a poll about learning goals (simply adding the class would not be a real task).
2. **A medium task** would be being prompted in class to fill out a poll or a quiz. We need students to be able to complete this task in order to gather and present the feedback to professors.
3. **A more complex task yet** would be signing up for office hours from the student perspective. This task is import to test whether kindergarten improves and facilitates student and instructor interactions by streamlining office hours.

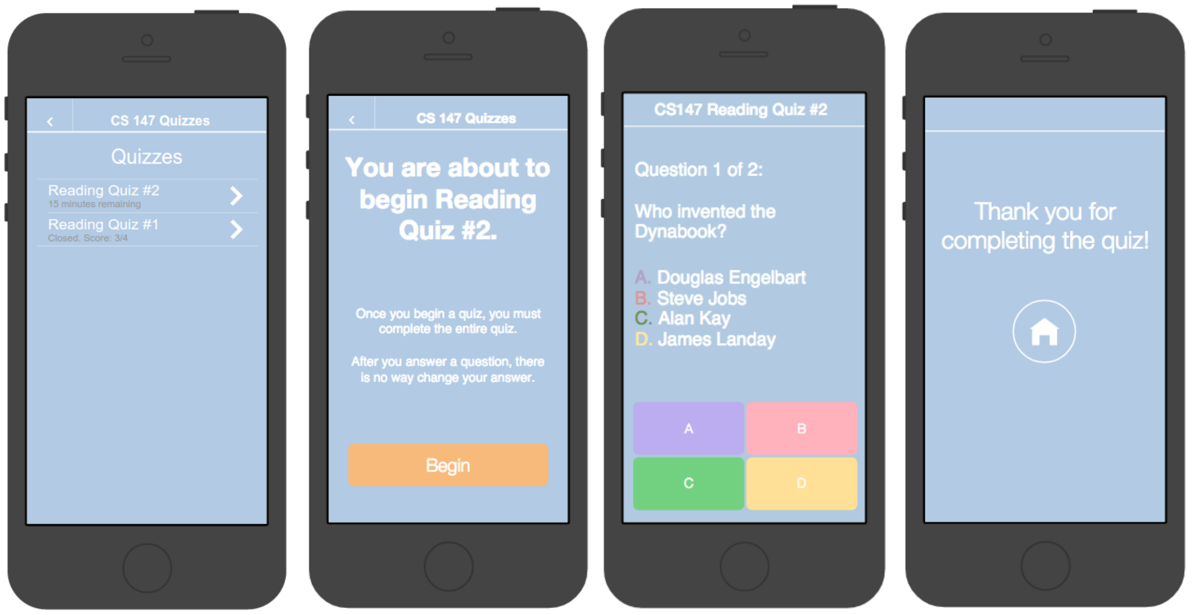
Final Interface Scenarios

1:Take a quiz

Imagine that you are student in CS147 and, when you enter class, you notice that Professor Landay has put up a reading quiz covering last night's reading. Instead of a long URL or an expensive iClicker, however, this class is using Kindergarten.



When you open the app, click the "login" button (please, not that this function is wizarded in our app with a set email and password). This will take you to your homescreen, which lists all the classes you are enrolled in as a student and/or as an instructor. To get to the quiz, you will need to got to you CS147 class screen by clicking on "CS147". You will notice a notification indicating that Professor Landay's quiz is live. Clicking on "Quizzes" will allow you to see all the quizzes for this class.



Once you begin a quiz, you must complete the entire quiz.

After you answer a question, there is no way change your answer.

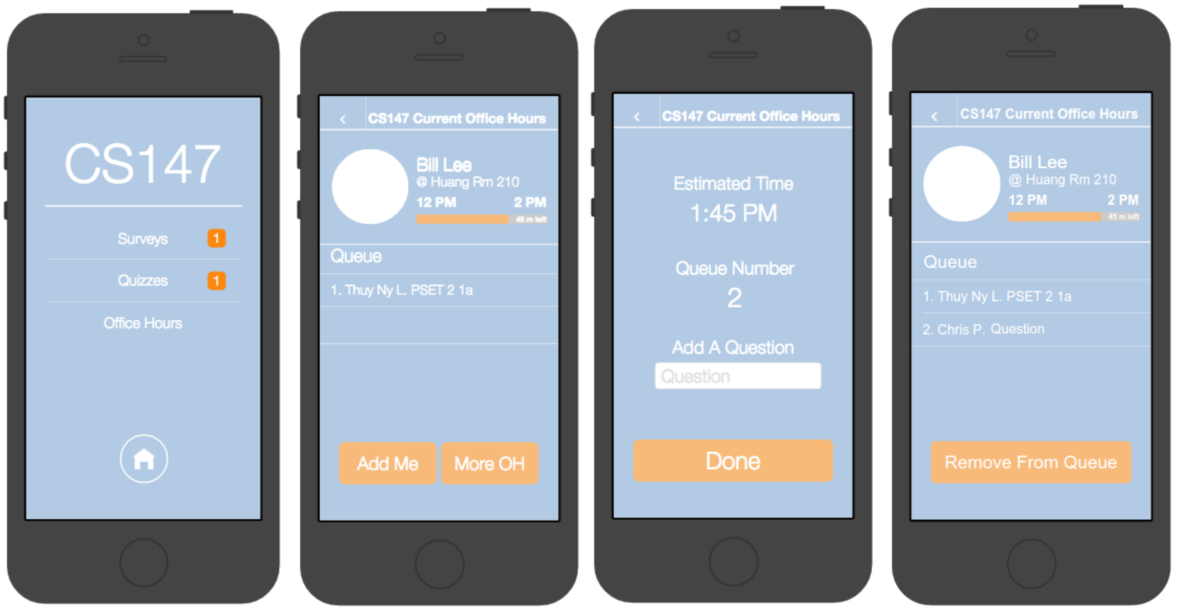
Who invented the Dynabook?

A. Douglas Engelbart
B. Steve Jobs
C. Alan Kay
D. James Landay

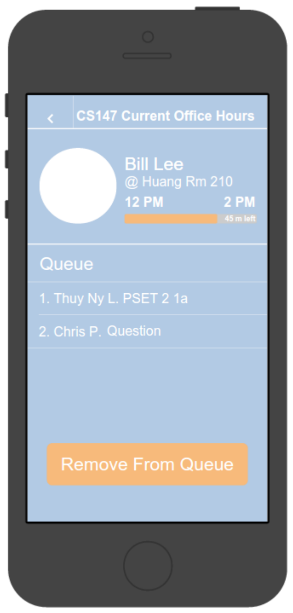
Clicking on the appropriate quiz, then click “Begin” to begin the reading quiz when you’re ready. Once you select the final answer, you will be directed to the Quiz Completion screen. Once there, you will be redirected back to the class’s home screen.

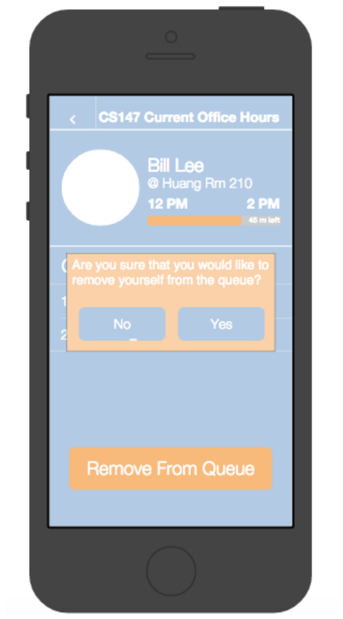
2:Student sign up for office hours while the instructor manages office hours
Later that week, you are working on an assignment for CS147 and decided that you have a question that you want to bring to office hours. The problem with how office hours are currently handled is that I have no way of knowing how crowded or empty office hours will be until I show up there. With Kindergarten, however, you can view and add myself to the queue without having to be physically present.

Student Screens

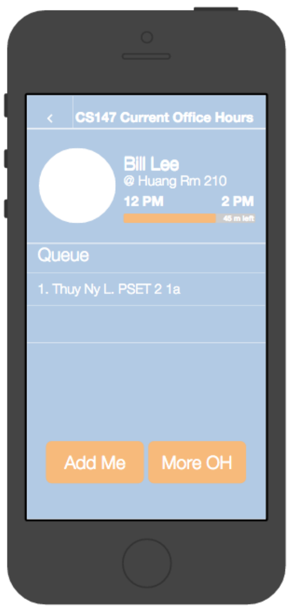


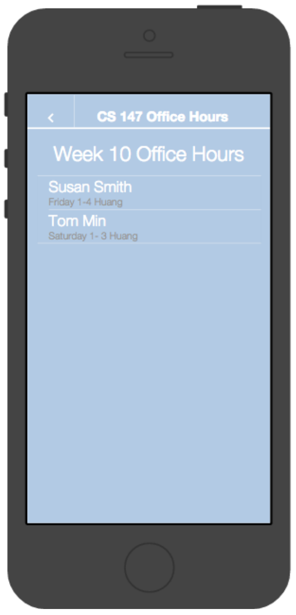
You are in the CS147 home screen. To sign up for office hours, you click on "Office Hours", which displays the current (or upcoming office hour). Let's say you decide to attend office hour. To add yourself to the queue, click "Add Me". The app will calculate an estimated time based on how the number of people on the queue as well as give you a queue number. To finish adding yourself to the queue, type in a question into the "Add a Question" and click "Done". On the office hour screen, you will see your name added.




CS147 Current Office Hours
Bill Lee
@ Huang Rm 210
12 PM
2 PM
45 m left
Are you sure that you would like to remove yourself from the queue?
No
Yes
Remove From Queue

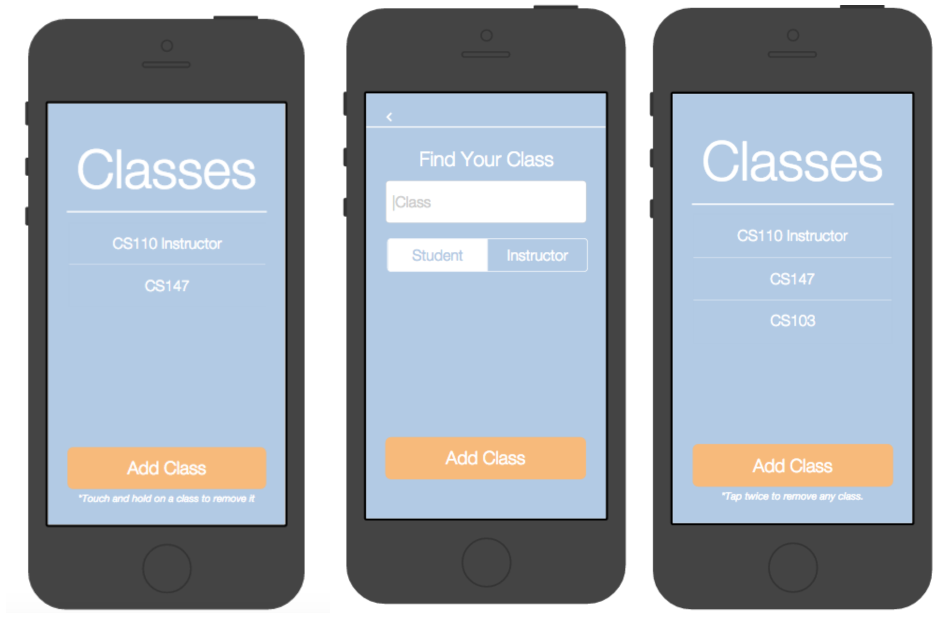
Let's say you figure out your question and want to remove yourself from the queue. To do that just click on "Remove From Queue" and then select "Yes" on the notification screen that pops up.





Let's say that you decide to not attend this office hour, then you can view other office hours by clicking "More OH" on the current office hour screen. This will take you to a screen that list the office hours for the current week.

3: Add a new Class
Imagine that it is now the next quarter and your CS103 also has decided to use Kindergarten.



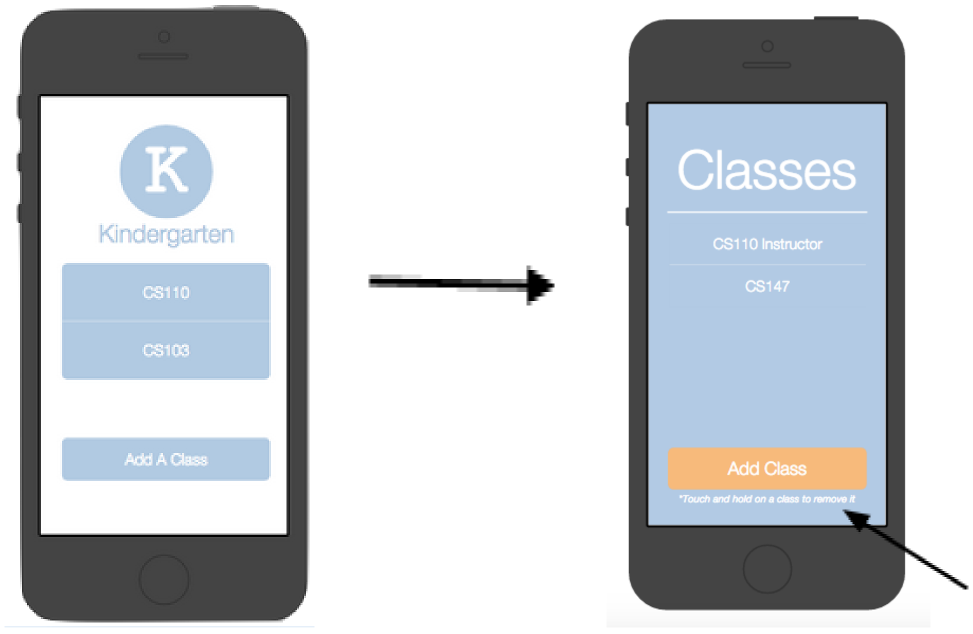
From your home screen, Click "Add Class" and type in "CS103" into "Find Your Class", and then click "Add Class". The app will redirect you back to your home screen, where you will see CS103 added.

Major Usability Problems Addressed

Level 3 or 4 heuristic violations:

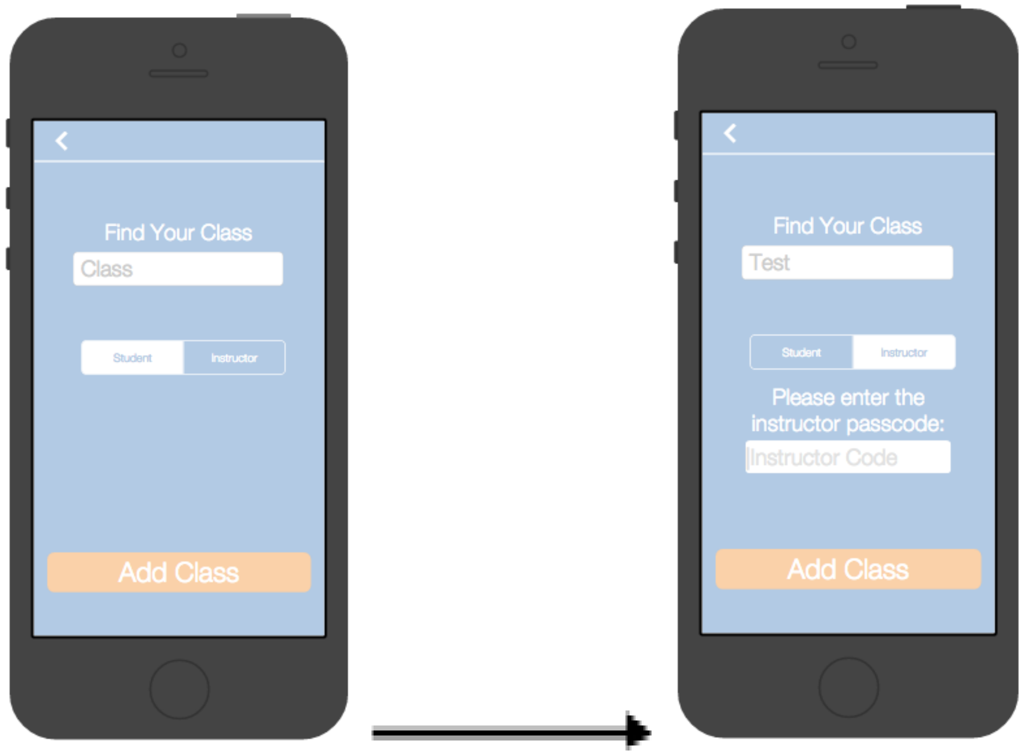
1. **[H2-3: User Control & Freedom][Severity 3]** There is no way to undo joining a class. Thus, if you joined a class by accident there was no way to undo this error.

2. **[H2-3: User Control & Freedom][Severity 3]** If you sign up for a class as a student by accident when you are an instructor, there is no way to change this mistake. Both of these violations had the same fix for our Hi-fidelity prototype. We created a Remove Class functionality and added it to the screen that lists a user's classes.

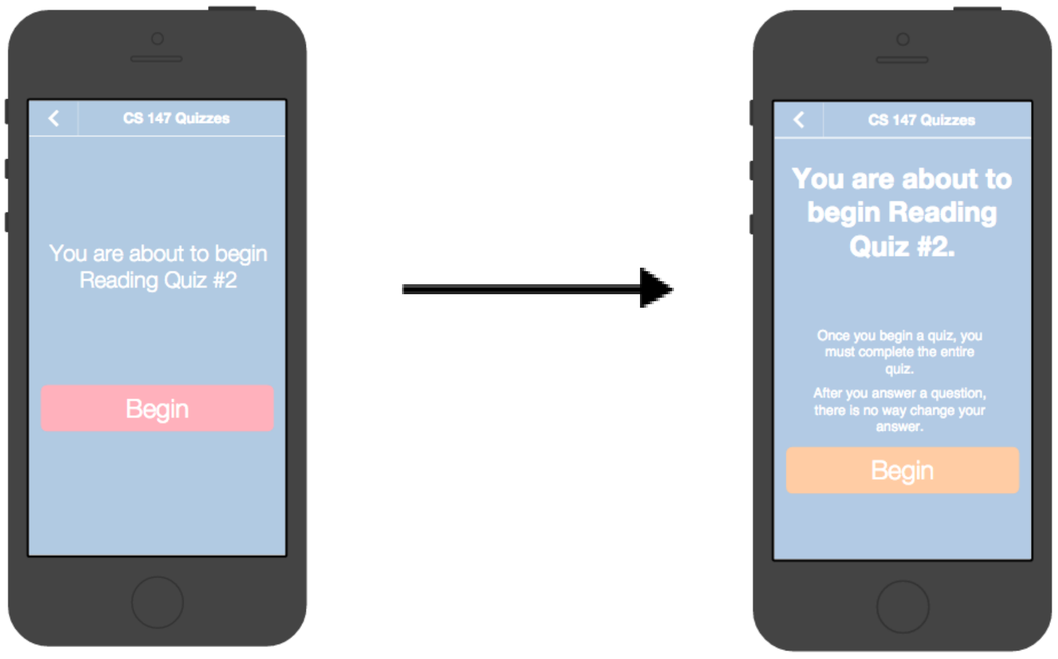


Instructions for deleting a class

3. **[H2-5: Error Prevention][Severity 3]** There was no confirmation to ensure that only people who were actual instructors could sign themselves up as instructors for the course. To fix this problem, we included a required instructor code field for signing yourself up as an instructor for a class.



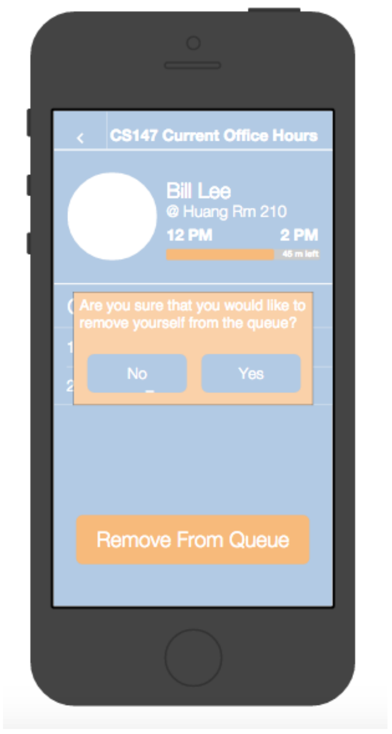
4. **[H2-3: User Control & Freedom][Severity 4]** Users were not able to go back to change their answers on quizzes after they choose an answer the first time. In addition, there was no warning to users that this was the case until they wanted to go back to change an answer and couldn't. We wanted to keep it so that users could not go back to change answers because we wanted the quizzes to be very quick and fluid. Instead, we added warnings before users start a quiz to notify them that they cannot back and change their answers.


CS 147 Quizzes
You are about to begin Reading Quiz #2
Begin
CS 147 Quizzes
You are about to begin Reading Quiz #2.
Begin

Once you begin a quiz, you must complete the entire quiz.

After you answer a question, there is no way change your answer.

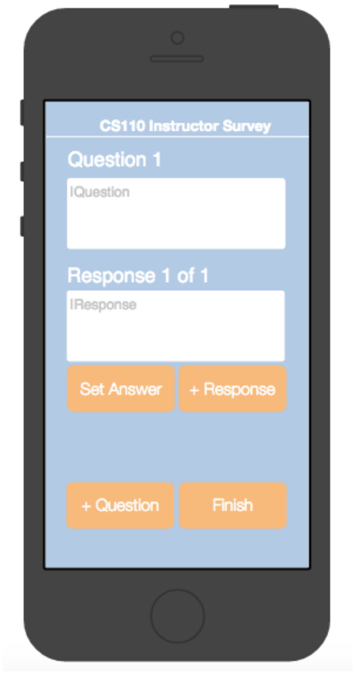
5. **[H2-9: Help Users Recognize, Diagnose, & Recover from Errors][Severity 4]** Previously, if a student removed themselves from the queue by accident, there was no way to recover. To fix this, we now ask the user for confirmation before they are removed from the queue.



CS147 Current Office Hours
Bill Lee
@ Huang Rm 210
12 PM
2 PM
45 m left
Are you sure that you would like to remove yourself from the queue?
No
Yes
Remove From Queue

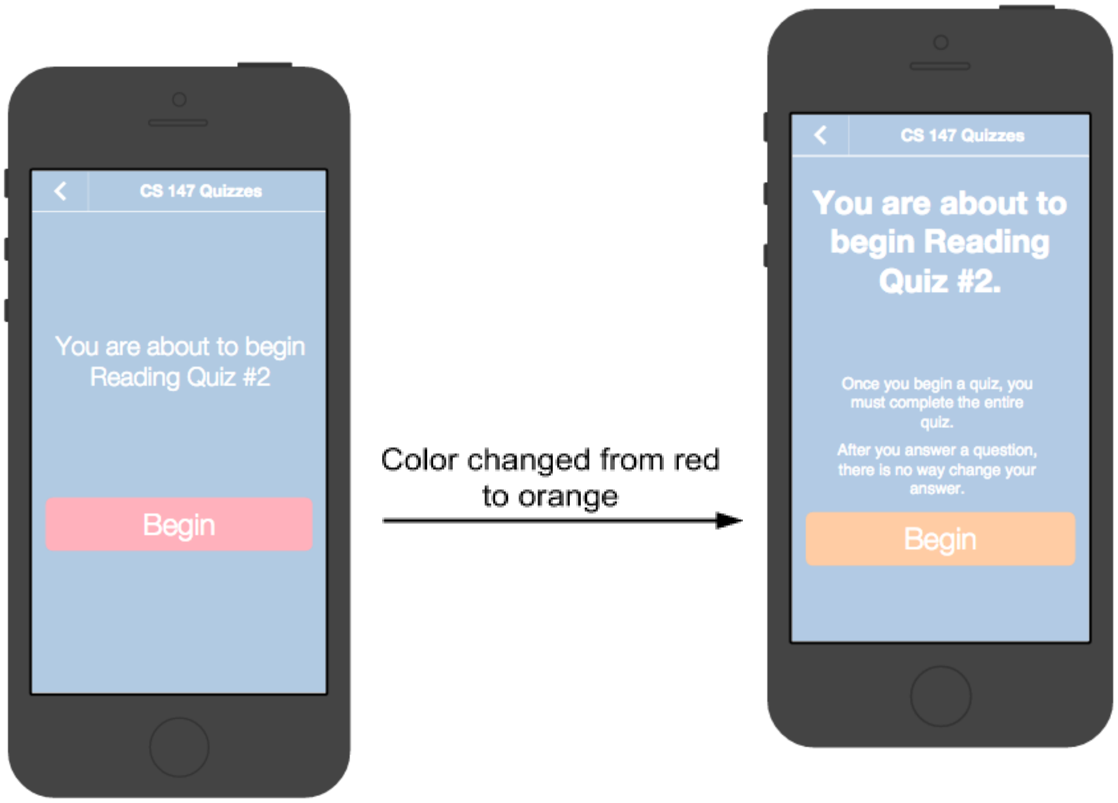
6. **[H2-2: Match Between System & Real World][Severity 3]** Searching for a class is difficult for users because they have to just hope that they enter the correct thing. If once you start typing, Kindergarten offered suggestions for classes based on what you were entering, this problem would improve. We did not fix this in our prototype because, ultimately, we wanted to focus on implementing tasks that would be most valuable for future user testing. If we were to continue redesigning and implementing our application further, we would likely change this, however.

7. **[H2-10 Help and Documentation] [Severity 4]** The process of creating a poll as an instructor was not very clear to users. Testers were confused about the specifics of words we were using on that screen such as what exactly a "response" was and what clicking the Done button actually did. In addition, users complained about how including the functionality for Adding a Poll on the same screen as the list of open and closed polls was very crowded. To address, these problems, we created an Add a Survey button that takes instructors to a separate screen to create a survey.

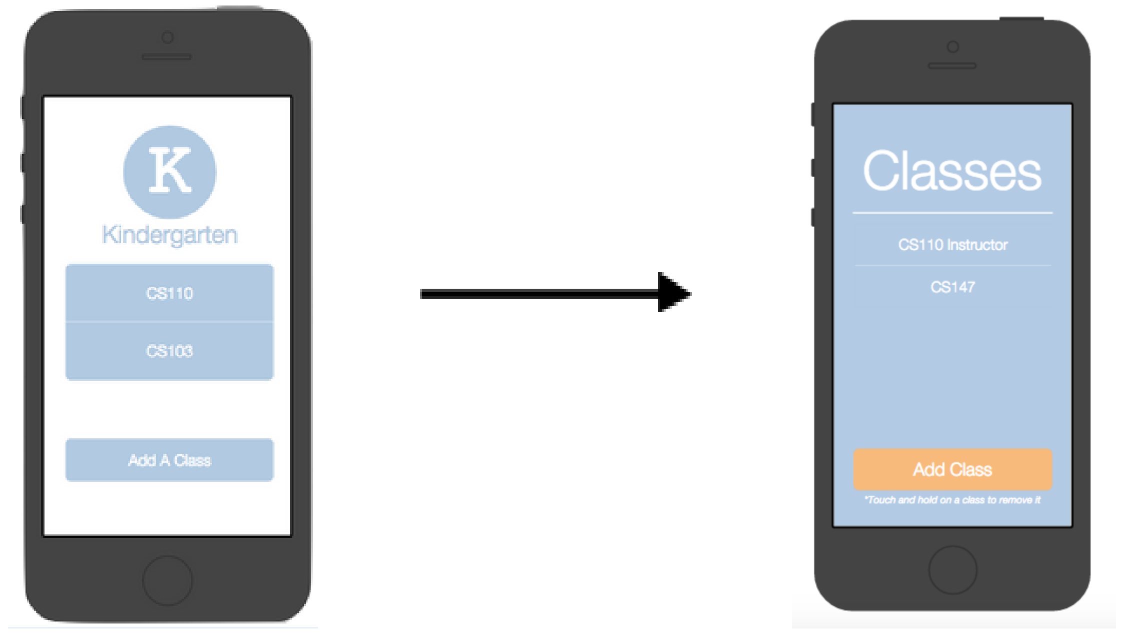


Other changes:

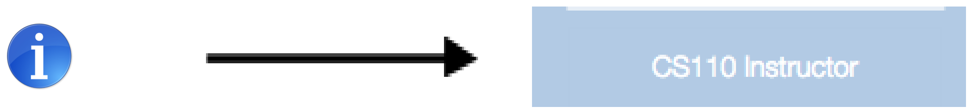
1. It was pointed out in our heuristic evaluation that the red that we were using for many of our buttons is the same red that is used in many iOS apps to represent deleting something. This caused some testers to be hesitant before performing actions because of the color of the button. To address this, we changed our colors from red to orange.


CS 147 Quizzes
You are about to begin Reading Quiz #2
Begin
Color changed from red to orange
CS 147 Quizzes
You are about to begin Reading Quiz #2.
Once you begin a quiz, you must complete the entire quiz.
After you answer a question, there is no way change your answer.
Begin

2. We changed our home screen that lists a user's classes from having a white background with blue buttons and text to having a blue background with white text. We did this so that the screen would fit better with the rest of the system thematically.



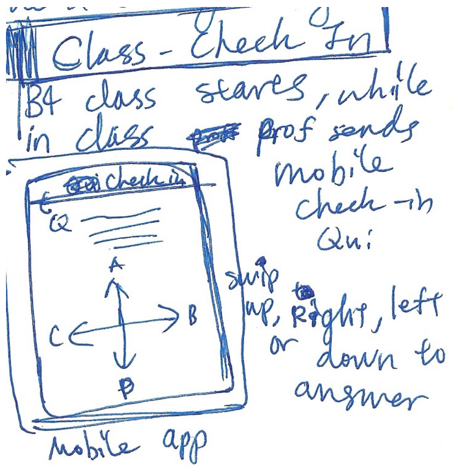
3. Previously, we used a small icon to indicate that a user was an instructor for the class that they were currently looking at. Users found this confusing, however, because that button so often indicates a button that one can click to get more information. We found that testers in our heuristic evaluation thought that clicking on it would give more information. To fix this, we stopped using this icon to indicate that a user was an instructor for the given class and now just type out the word "Instructor".

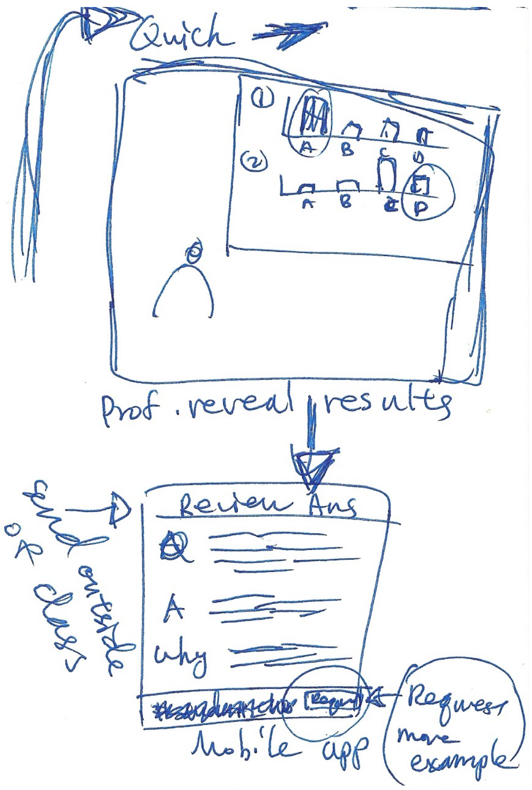


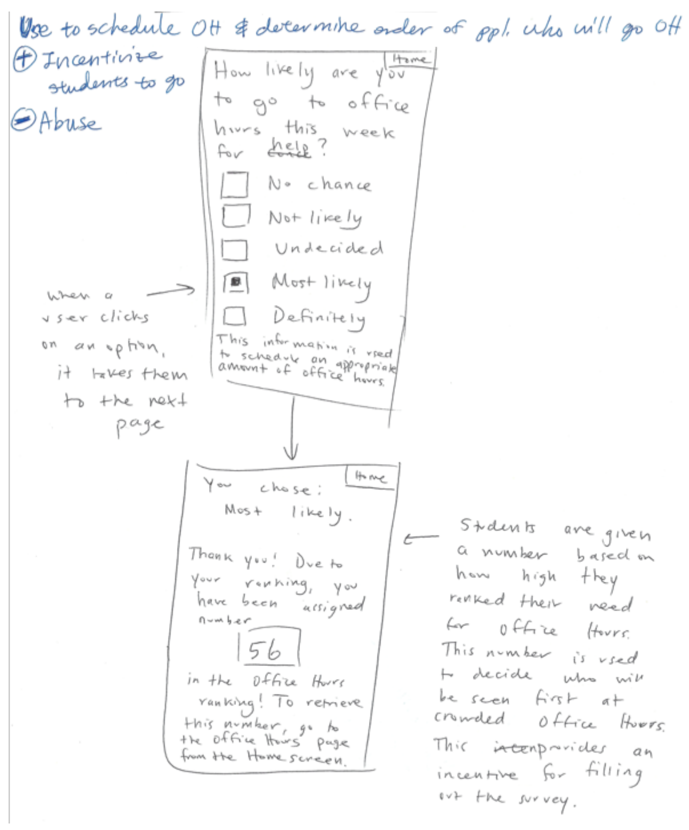
Design Evolution

We had a remarkable stability in the essential problem addressed by the app, and therefore, many (but not all) of the initial sketches remain relevant to the current design, and we took many of those ideas from the initial sketch all the way through, sometimes changed, sometimes changed.

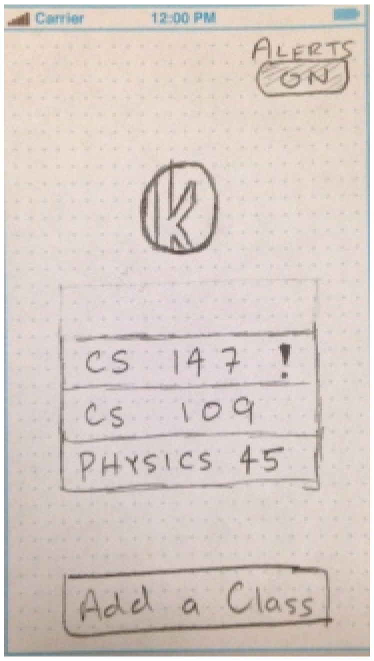
Below are two initial sketches: the “Class check-in” idea, which was to be a sort of Tinder for class, and the “Office Hours scheduler” idea. Besides the user interface conceit of swiping to answer the quizzes, which was found to be unintuitive by actual users in the lo-fi design, the essential structure of the “Class check-in” idea remains in the quiz system in the hi-fi prototype. And although the fundamental idea of a *poll* for office hours from the “Office Hours scheduler” idea was not adopted, we took the idea of the *queue* which is there in the second screen of that idea and ran with it.

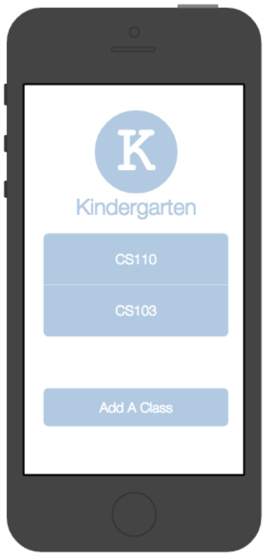

Class - Check In
B4 class starts, while in class prof sends mobile check-in Qui
check in
Q
A
B
C
D
swip up, Right, left or down to answer
Mobile app

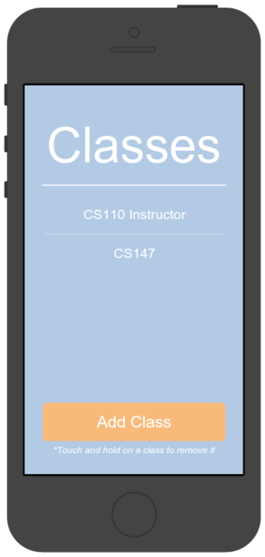



Use to schedule OH & determine order of ppl. who will go OH
+ Incentivize students to go
− Abuse
Home
How likely are you to go to office hours this week for ~~[illegible]~~ help?
No chance
Not likely
Undecided
Most likely
Definitely
This information is used to schedule an appropriate amount of office hours.
when a user clicks on an option, it takes them to the next page
Home
You chose: Most likely.
Thank you! Due to your ranking, you have been assigned number
56
in the Office Hours ranking! To retrieve this number, go to the Office Hours' page from the Home screen.
Students are given a number based on how high they ranked their need for Office Hours. This number is used to decide who will be seen first at crowded Office Hours. This ~~[illegible]~~ provides an incentive for filling out the survey.

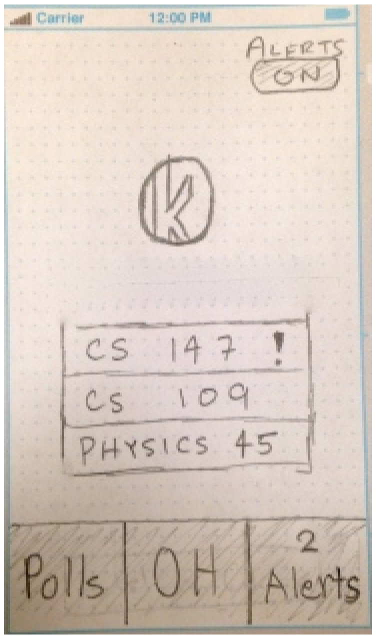
Here is the design progression of what the home screen for the app would look like (in the case of the hi-fi prototype, after the login screen). The “Alerts” picker was eliminated after the lo-fi testing, when we realized that it cluttered the design. The “Add Class” button was changed after the medium-fi testing when we were told by a heuristic evaluation team that it did not stand out enough.

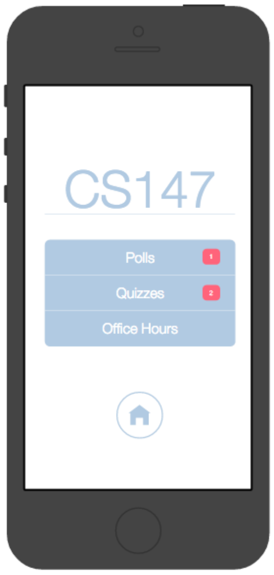


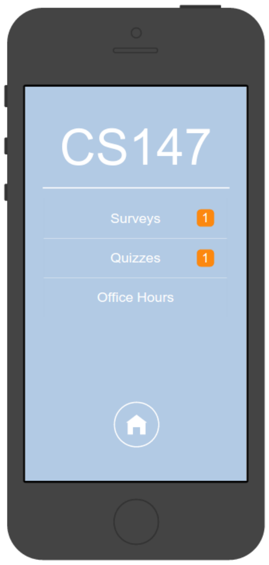




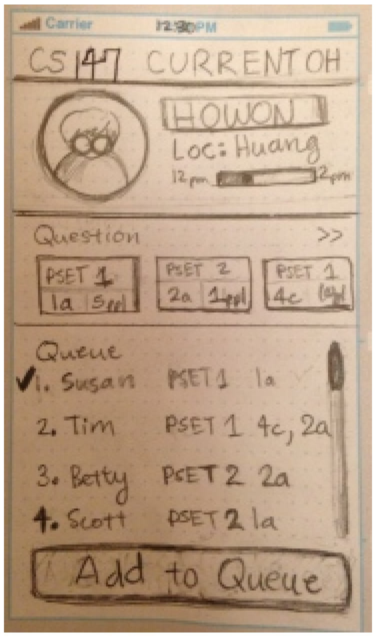
This is the menu screen *for a specific class*. We envisioned it as a slide-up menu screen in the lo-fi design, but realized that this functionality would be better done in a separate screen. We also folded in the separate "alerts" screen, since viewing the alerts would happen in the normal functioning of the app.

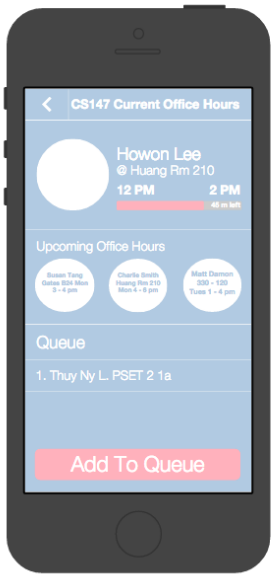


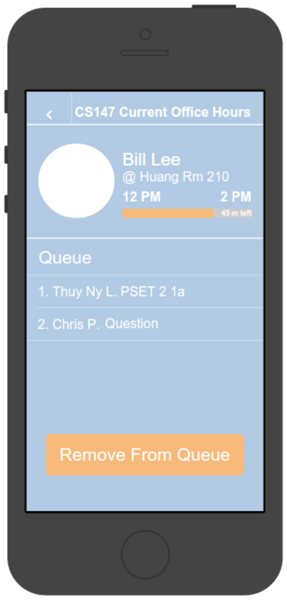




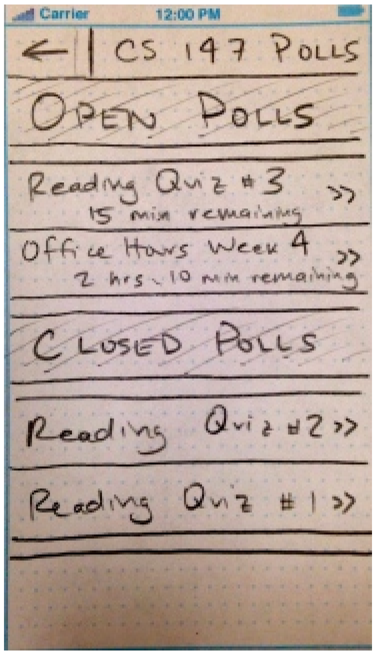
The *Office Hours* screen allows the student to inspect the office hours for the class and how many people are in any current office hours which are happening. This screen, except for the overall color change, did not change significantly throughout the design.

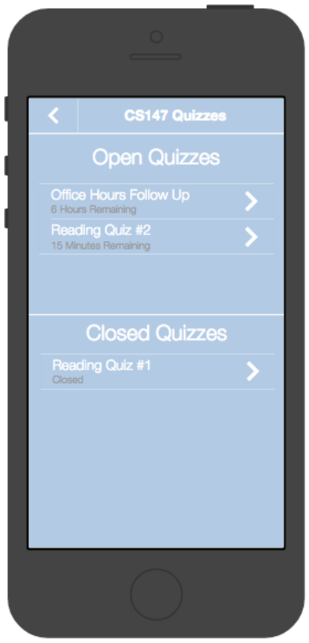


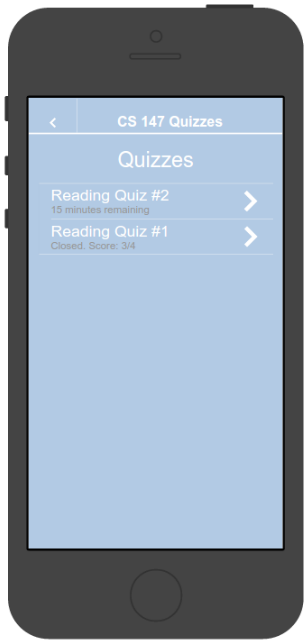




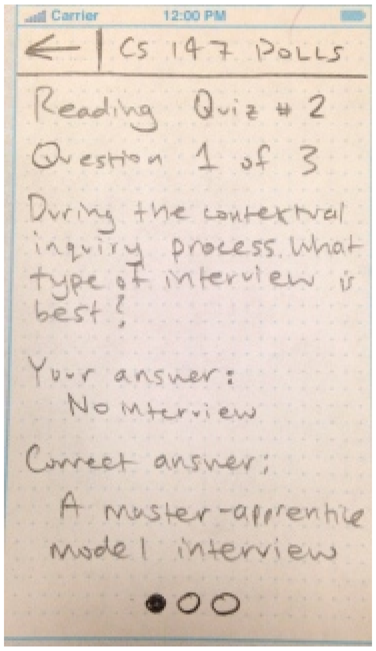
This is the screen from which one enters the open quizzes, to take them, or the closed quizzes, to review them. The separate sections for open versus closed quizzes were folded into one on recommendations made after the medium-fi prototypes, because the semantics were obvious enough simply putting the open/closed information on the small quiz blurb.







This is viewing the reading quiz itself. With the exception of the elimination of the swiping conceit, it remained unchanged between the medium-fi and hi-fi prototypes.


Carrier 12:00 PM

CS 147 Polls

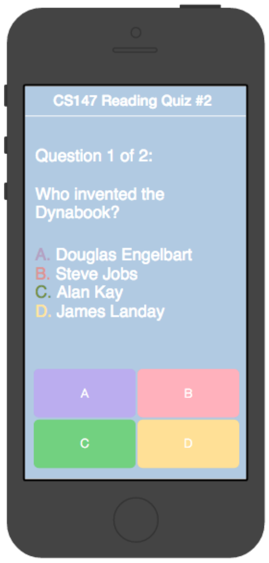
Reading Quiz # 2

Question 1 of 3

During the contextual inquiry process, what type of interview is best?

Your answer:
No interview

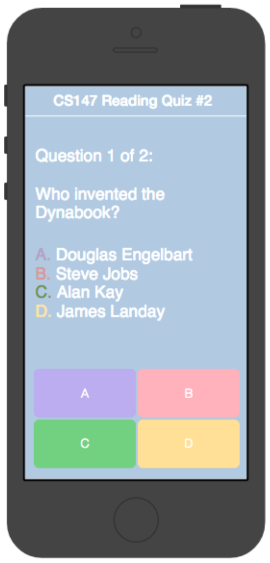
Correct answer:
A master-apprentice model interview


CS147 Reading Quiz #2

Question 1 of 2:

Who invented the Dynabook?

A. Douglas Engelbart
B. Steve Jobs
C. Alan Kay
D. James Landay

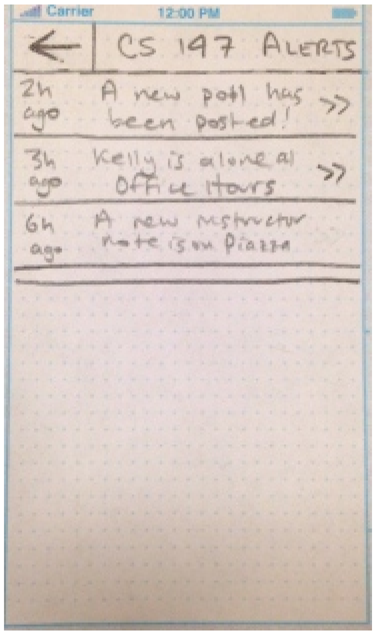

CS147 Reading Quiz #2

Question 1 of 2:

Who invented the Dynabook?

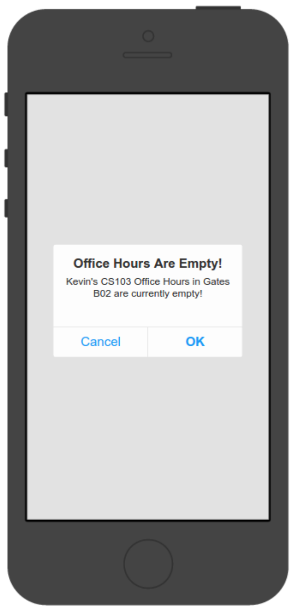
A. Douglas Engelbart
B. Steve Jobs
C. Alan Kay
D. James Landay

Alerts is a listing of push notifications for that class. There are push notifications for in-class quizzes and for office hour alerts. This functionality was folded into another screen for the medium-fi and hi-fi prototypes, but there is also a toast for this functionality here.

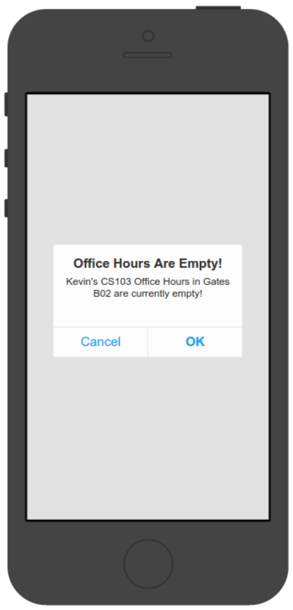

Carrier 12:00 PM

CS 147 Alerts

2h ago	A new poll has been posted!	>>
3h ago	Kelly is alone at Office Hours	>>
6h ago	A new instructor note is on Piazza	

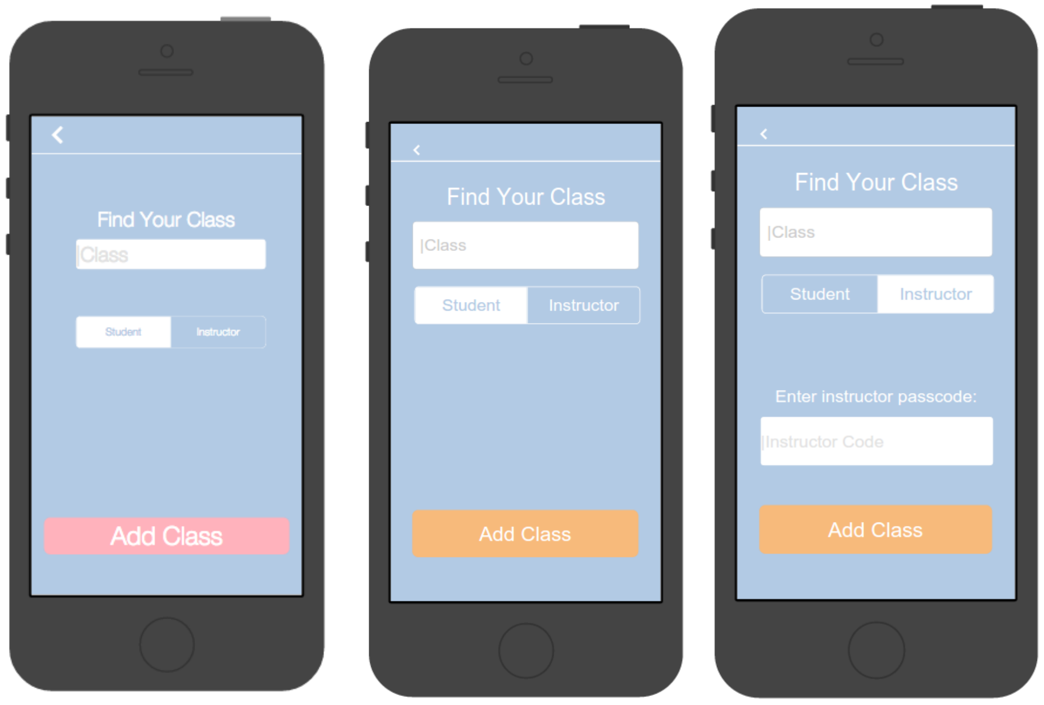

Office Hours Are Empty!

Kevin's CS103 Office Hours in Gates B02 are currently empty!


Office Hours Are Empty!

Kevin's CS103 Office Hours in Gates B02 are currently empty!

This is the "Adding a Class" screen, which was not present in the lo-fi version. The only comment the heuristic evaluation team had for us is that there was no authentication for the instructors. Therefore, the latter two screens show the hi-fi prototype, where we added the authentication.



Prototype Implementation

We use Cordova with Ionic as a runner in order to build the prototype, with collaboration online using Git and Github. This allowed us to write HTML and CSS and some Javascript in order to compliment generated code using the Proto.io medium-fi prototype, allowing us to present a prototype in a native app. This is technically a wrapper around a webView, but with native features in a JS API. This allowed us to avoid much of the travails of extremely rapid iOS or Android development and focus on the UI work, and write much, much less code in general, working much more inside the prototyping tool for the whole duration. We even did many of the interactions themselves in Proto.io. Ionic is a runner, which means that it served mainly as a convenient way to not have to deal with the Cordova command line interface, which is not very convenient for developing the HTML on the browser before on the Cordova build. Although *a priori* we decided that the application's usage fit a native app and therefore we developed these as targets, it was also absolutely trivial to target a mobile web app basically instantaneously, which is another advantage of this approach.

They would not really help us to be a solid base for a real, fast application, without a bevy of extremely arcane tricks to get it to be much faster. For example, you will note Cordova's bad treatment of hardware acceleration, which will mean that button clicks will occasionally respond sluggishly. In an actual system, this would require a fair amount of thought to get working properly, and more complexity than you would think - although probably less than creating the whole native app in the native language - but for a prototype it's not bad.

There are not any Wizard of Oz techniques anymore in the hi-fi prototype. However, a solid majority of the data remains hardcoded, including the two default classes shown, and the two quizzes shown. The gathering of such data is orthogonal enough to the main task of designing the UI that we decided that the data should remain hardcoded, except for the parts which are entered and interacted with.

A solid part of the instructor's functionality is somewhat missing: although the screen exists for the creation of a quiz or a poll, and therefore one can envision it, it is not yet functional, meaning that a quiz or poll cannot actually be created on the instructor's side of the app as of yet. This is mainly due to the focus we implicitly had from the beginning, of creating the app from mainly the student's point of view. This would be the next step to add (besides a full backend) before beginning implementation on a fully functional prototype, and to deployment and production.