

electurefy

Value Proposition

Make lecture a more effective learning tool for both students and teachers.

Jennifer Farman - Team Manager

Daniel Hok - Development & User Testing

Juan Posadas Castillo - Design & Development

Nikhita Obegadoo - Documentation & User Testing

Demo links

iPhone demo link:

<https://electurefyapp.proto.io/share/?id=27e3a38f-7214-4d5d-b46c-3136ca9ca052&v=2>

iPad demo link:

<https://electurefyapp.proto.io/share/?id=a34570d4-c331-4265-8754-a70cfb1595fc&v=1>

Problem and solution overview

Today's colleges have all it takes for success: dedicated lecturers and motivated students. However, in a lecture hall with a ratio of a hundred students to one instructor, it is hard - if not impossible - for students to effectively communicate their understanding levels.

Electurefy is a mobile app that bridges the communication gap between students and instructors. By allowing instructors to prompt students for feedback at specific moments during lecture, Electurefy provides a simple, user-friendly and non-distracting way for students to indicate their understanding levels to the instructor, thereby making lecture a more effective learning tool and providing instructors with actionable data that allows them to adapt lecture to students needs.

Tasks

Simple task : Monitoring student confusion in real time (Instructor)

This task is identical to the one that we previously had, although its UI is significantly different from our previous version.

This task is performed by the instructor. During lecture, he/she sends out a "bolt", which allows students to indicate either *confusion* (indicated by a question mark) or *understanding* (indicated by a lightbulb). Bolts can be sent to either all of the students in class, or a random subsection of the students (the instructor can switch between these options in "settings", which is an example of added functionality). The instructor can see the student response data as the students reply to the bolts (e.g: 30 confused, 10 understood), and can also swipe between all the bolts that have been sent out during that lecture to see the response breakdowns for previous bolts.

Moderate task : Submitting feedback (ie. responding to “bolts”) (Student)

This task is similar to our previous student task, although again the UI is significantly different from the UI we design for the low-fi prototype.

The student responds to “bolts,” sent out by the professor, to indicate whether they are experiencing confusion or understanding at that particular point in lecture. This data is immediately sent back to the professor, who can then adapt to the student feedback.

Complex task: Reviewing (overall and individual) performances (Instructor)

This task is different insofar as the instructor is not simply traversing lecture data. The professor can review performance from two different perspectives:

Part 1 - Overall & Best/Worst performance - We have added features so that the instructor also has meaningful data to look at (e.g: best/worst lecture, overall/average responses) to look at.

Part 2 - Specific lecture performance - The professor can also look at data for a specific lecture, which includes response data for each bolt sent during that lecture.

Revised interface design

Our interface design went through significant changes as we moved from the low-fi prototype to the medium-fi prototype. We responded and adapted to many of the concerns brought up during previous assignments, and many key elements of our UI underwent significant changes as a result.

UI changes for Task 1: Monitoring student confusion in real time

In response to feedback on our low-fi prototype, we completely changed the UI concept, which originally was dial that shifts from left to right to indicate varying degrees of confusion. We moved to a binary response system (confusion vs understanding), which we felt was more straightforward and which also gave us the ability to more effectively monitor student engagement with the app. This will give us the option to add functionality that ties student engagement with the app to class participation grades. Furthermore, the professor can now choose when and who to ask for feedback, as well as the time delay during which feedback can be sent in. These features reduce the distraction potential of the app, which we realized was a significant issue to users during user-testing and presentation feedback.

Below we illustrate the general flow (storyboard) for the instructor to achieve this task:

Figure 1: Instructor enters his/her credentials at login screen

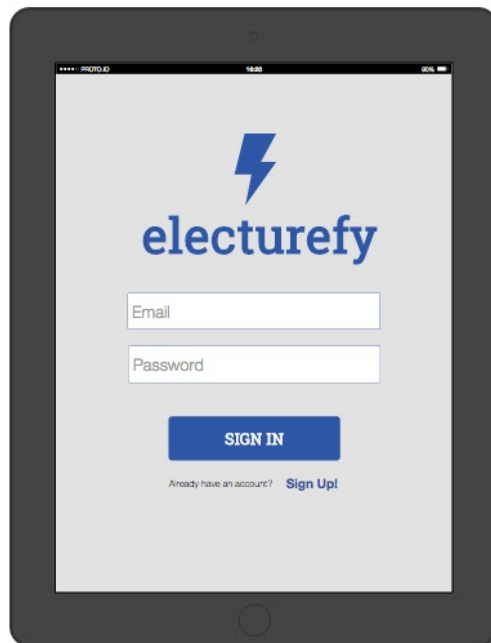


Figure 2: Instructor lands in My Classes section after login. Instructor can see lists of current classes they see and add a class.

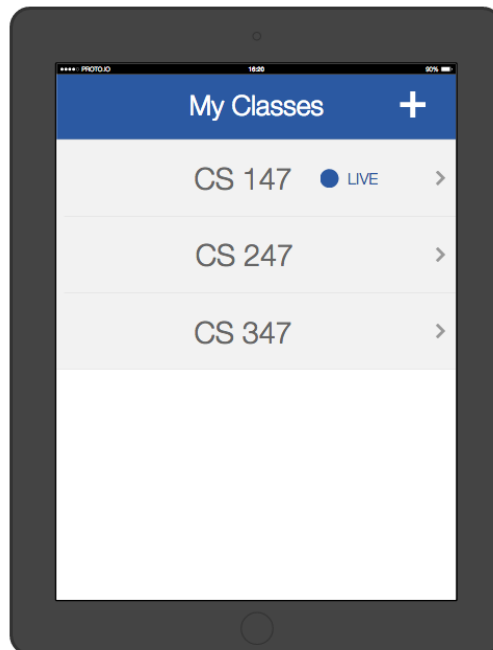


Figure 3: After clicking on a live lecture, instructors land in the 'Current Lecture' page. Here they can monitor their bolts, and slide / click on arrows to navigate through all the bolts sent. Also, they can send a bolt from this screen by clicking the 'Send Bolt' button.

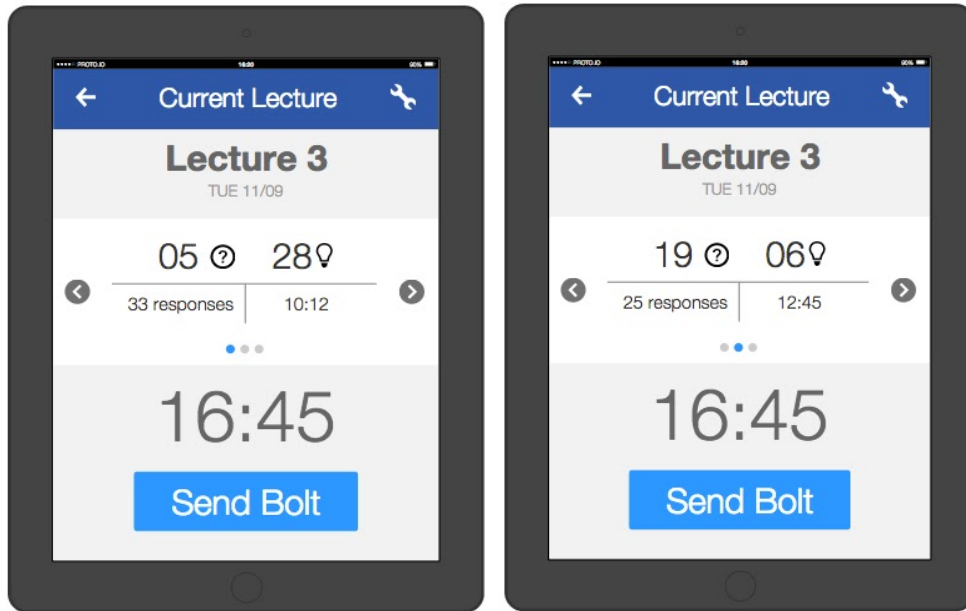
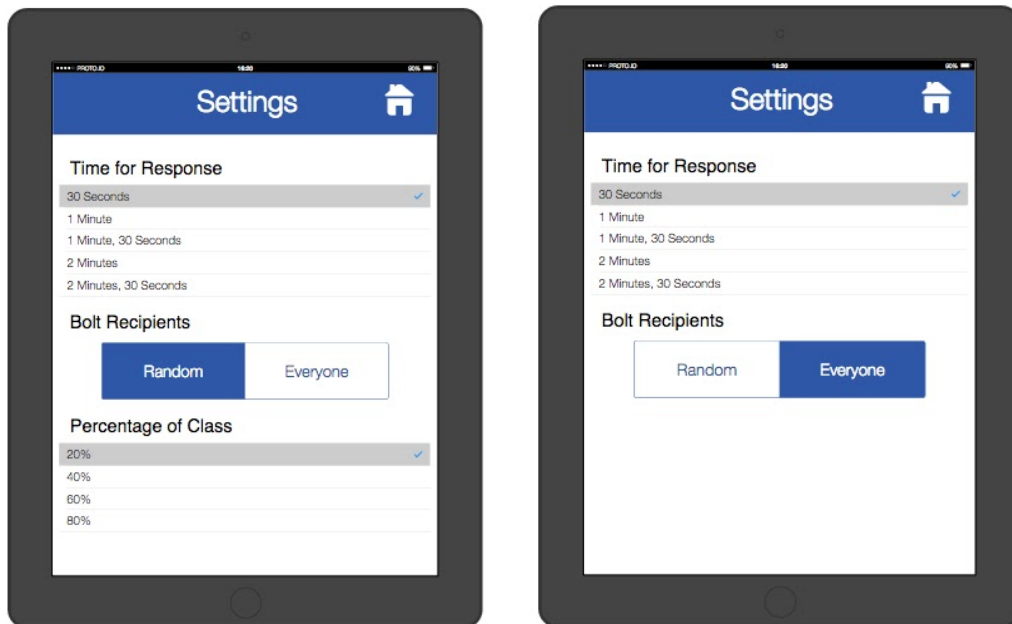


Figure 4: Instructors can click on the Settings (Wrench button) and set max. response time for bolts, and whether it should be sent randomly to a certain percentage, or to everyone.



UI changes for Task 2: Submitting feedback (responding to “bolts”)

The UI changed completely to a binary response choice (confusion vs understanding) instead a dial with a continuous range of choices. This is clearer and less arbitrary, and which benefits student engagement with the app.

When lecture is in session and the student receives a bolt while they are not currently in the app, they will receive a push notification. The student can press “Respond” and be led to a “Bolt Response” screen, where she can choose between a question mark icon (signifying confusion) and a light bulb icon (signifying understanding).

Below we illustrate the general flow (storyboard) for the instructor to achieve this task:

Figure 1: Students get notified of a new **Bolt** sent by the instructor.



Figure 2: After clicking on Response, students land on the current class bolt history, which shows the current bolt sent by the instructor with a gray background.

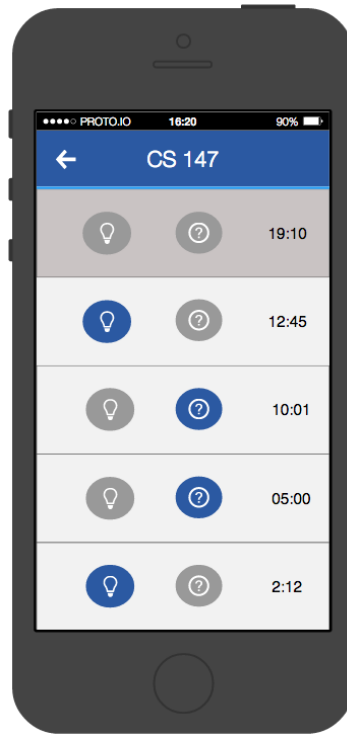


Figure 3 & 4: Students can then select the **bulb** for a positive comprehension, or the **question** otherwise.



UI changes for Task 3: Reviewing (overall and individual) performances

Previously, the UI only supported the traversing of harvested data from individual classes.

Now, in addition to viewing data from individual classes, one can see overall statistics, which give data on the best/worst class, and on the overall/average bolt response rates.

Below we illustrate the general flow (storyboard) for the instructor to achieve Part 2 of the complex task (*review overall & best/worst performance*):

Figure 1: Instructors can access the Statistics section of the app by clicking on a class from their My Classes page.

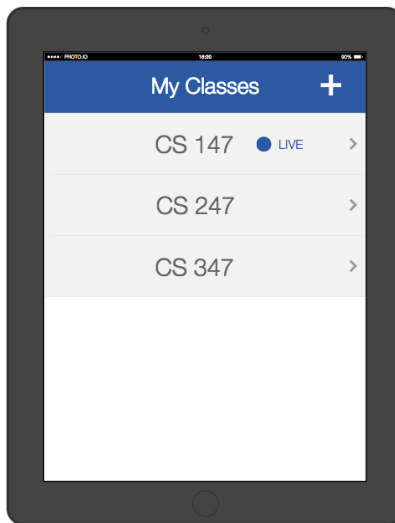


Figure 2: From here, instructors have two options. Click on a specific lecture, or see overall statistics.

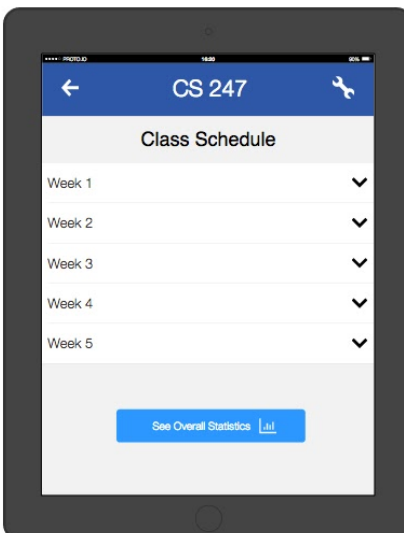


Figure 3: Here, instructors can see their best and worst performance and other statistics. By clicking on the the lectures themselves, they are able to see the specific statistics for that given page.

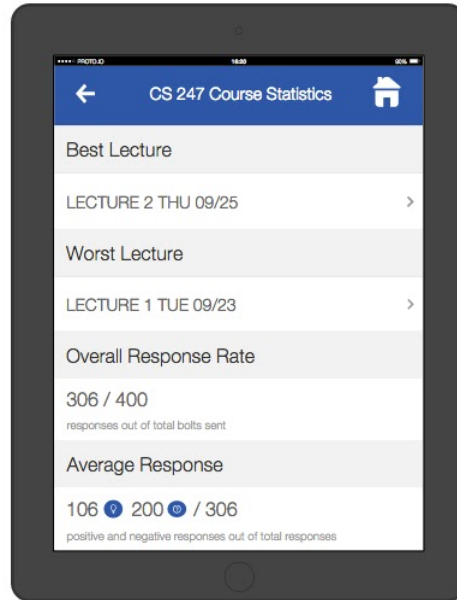
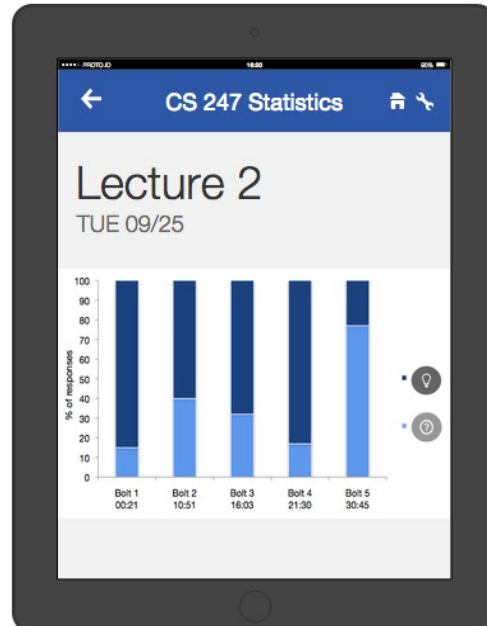
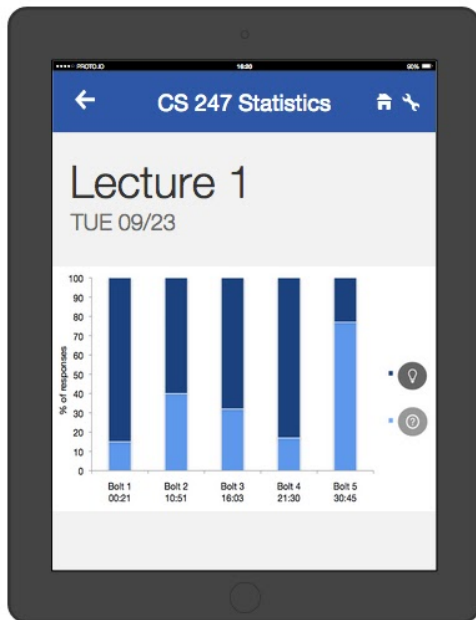


Figure 4 & 5: These are the individual statistics for lectures 1 (worst) and lecture 2 (best)



Below we illustrate the general flow (storyboard) for the instructor to achieve Part 2 of the complex task (*review specific performance*):

Figure 1 & 2: Instructors can choose a specific lecture by selecting the week their lecture was given on.

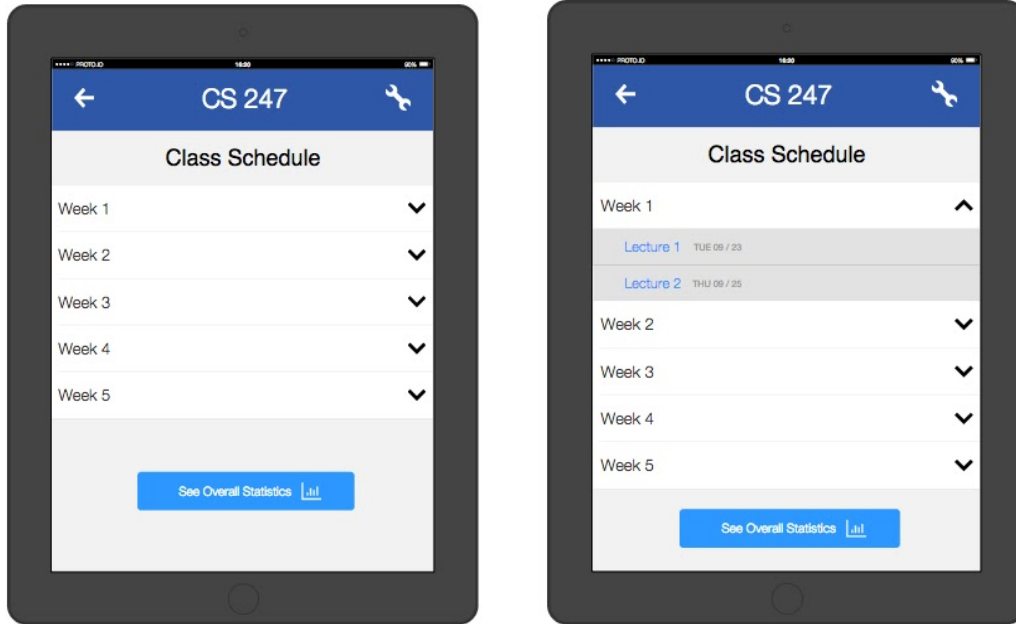
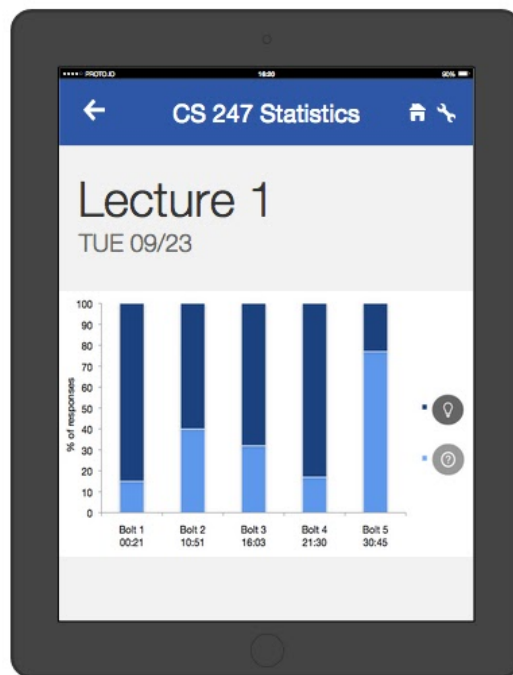


Figure 3: Instructors can access then the specific lecture they selected in the screens above.



Prototype overview

Tool Used

The tool we used was proto.io.

The tool was incredibly helpful in that it had all the elements required to translate our low-fi ideas into a medium-fi prototype. Additionally, the interactions feature helped us seamlessly move from one screen to another, and was convenient for viewing on a projector.

However, proto.io was not a good collaborative tool in the sense that there was no option for version control and consequently only one person could work on the screens at a time.

Limitations/Tradeoffs of the Current Prototype

This current prototype shows how each of our three tasks is to be carried out. Additionally, it has a clean and intuitive UI design. The features left out were signing up for an account, student settings and adding a class, none of which are crucial or relevant to our three tasks or app functionality.

Wizard Of Oz Techniques

- When the student receives a bolt notification on his/her iPhone screen, he/she is in fact receiving a simulated bolt. In real life, this bolt would be sent by the Professor.

Hand-Coded Features

- All our lecture and graphs statistics were hard-coded, since we are not dealing with an actual database.
- Our best and worst lectures were also hard-coded instead of harvested from real data.
- All the bolt information (ie. any sent or received bolts) for both the professor and the student was hand-coded, since the medium-fi prototype does not have the actual implementation of sending and receiving bolts.

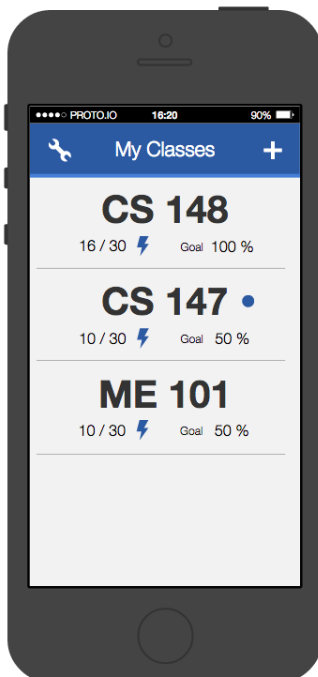
Prototype Pictures

iPhone Photos - Student:

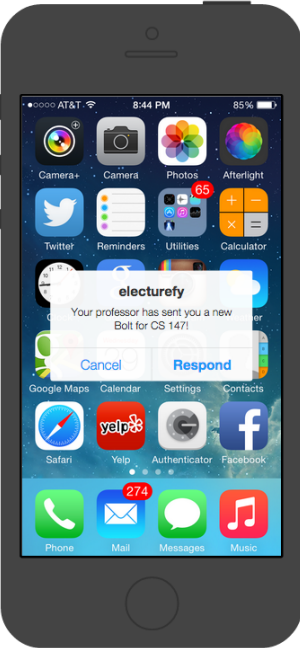
Login



Home Page - iPhone



Push Notification - iPhone

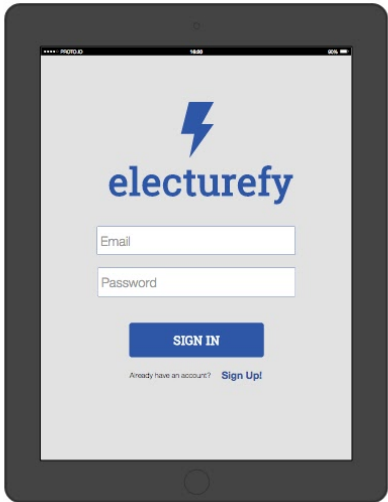


Answer Bolt - Student

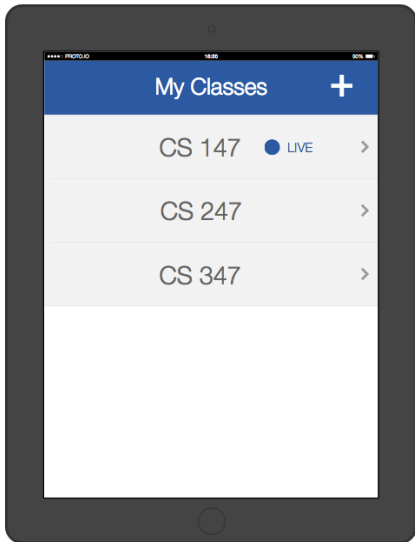


iPad Photos:

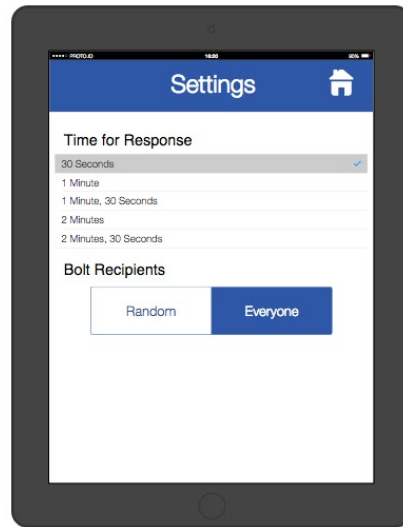
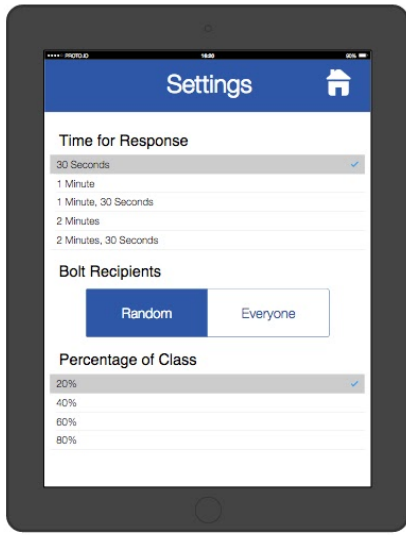
Login



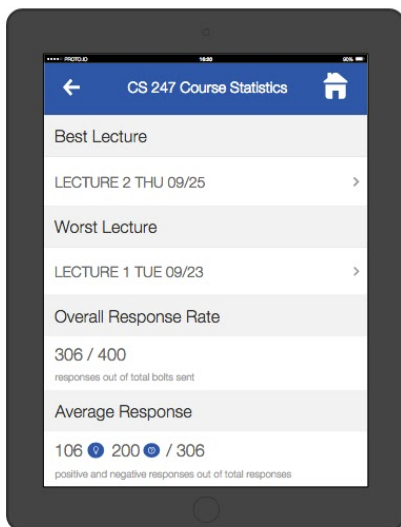
Home page



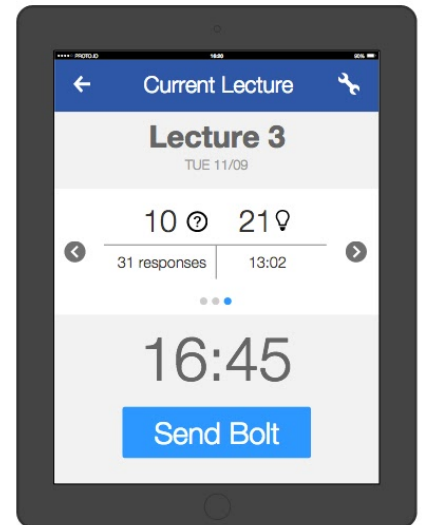
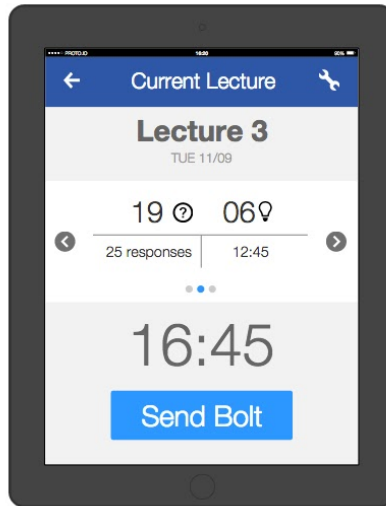
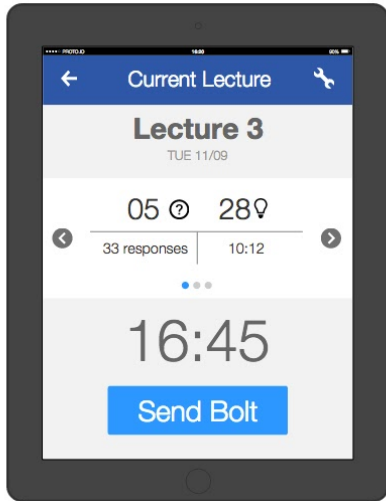
Settings



Course Statistics



Current Lecture



Past Lecture Data

