Meeting Notes 11/13

Annie Tang, Casey O'Brien, Katie Siegel, Qui Nguyen

(Rescheduled to Wednesday at 6 because of conflict)
Attendance: Annie, Casey, Katie, Jaya

Team work doc looks good
Talk about Design Doc

Some repetitition? Ex. Key concepts/Under Data Model

Discussing the content diagram

- Arrows from students to EDUvote (for students without phones, not for MVP)

Discussing Key Concepts

- Explaining key concpets
- How to set up a course?
    - Will it make it more difficult for instructors who only have one section?
    - Minimize number of clicks (to create a question, does the instructor have to choose the course, the section, the lecture, and then input the question and responses)?
    - We'll know which lecture the question happened in based on the date of the lecture?
    - What happens when they want to reopen a question at a later date?
    - Or at the same date?
        - Instructor opens question at the beginning and then end of the lecture
        - Create a new question object (Duplicate question)?

What students can access:

- Can students see questions later?
- What if professor don't want the students to have access to questions later on?

Differentiation from clicker:

- How does it keep people from cheating?
- People won't want to hand over their phones
- Make sure this is expressed in design doc (it's under design challenges)

Access Tokens:

- To give the app permission to send text messages on your behalf, you need an access token
- (One access token per dev account)
- Would allow people to send messages on our behalf
- Only one place where we put acces token in actual code, Twilio handles this securely

Users send in a lot of messages and clog the queue:

- Backup is in response sending
- We only send feedback to a student once per question

Some mitigations to standard attacks missing:

- SQL injection
- Cross-side forgery
- Persistent/non-persistent… something?
- Man in the middle attacks?
- (Check out recitation 9)
- Standard rails frame comes with some of these mitigations

How to handle registration?

- Web App: online page for student to registration
- Registration from phone: text in Kerberos
- Should we validate kerberos?
    - Get database of all kerberos (Katie already got this)
- Allow professors to delete students if somethign doesn't look right?

Question display page: Phone number tag not useful?

Design Challenges:

- Types of analysis
- Precompute (aggregate data on timely basis)
- For now, just recompute each time
- Save whenever a computation is done
- If computation power seems sufficient, we probably don't need to worry about this

Models for MVP:

- Courses model
- User model (differentiate between students and instructor? or use boolean?)
- Store response that student gives
    - Table with Question ID, Student ID, Response
- Question model
- Course has many students, instructors (many-many relationship)
- Instructor doesn't need a phone number, how to handle this?

To Use Bootstrap?

- Katie wants bootstrap
- Annie thinks not

Discussion of moving other meetings to Wednesday with 6. Need to check with everyone.