HandRaise - Group 40

Faculty Panel Design Review Presentation

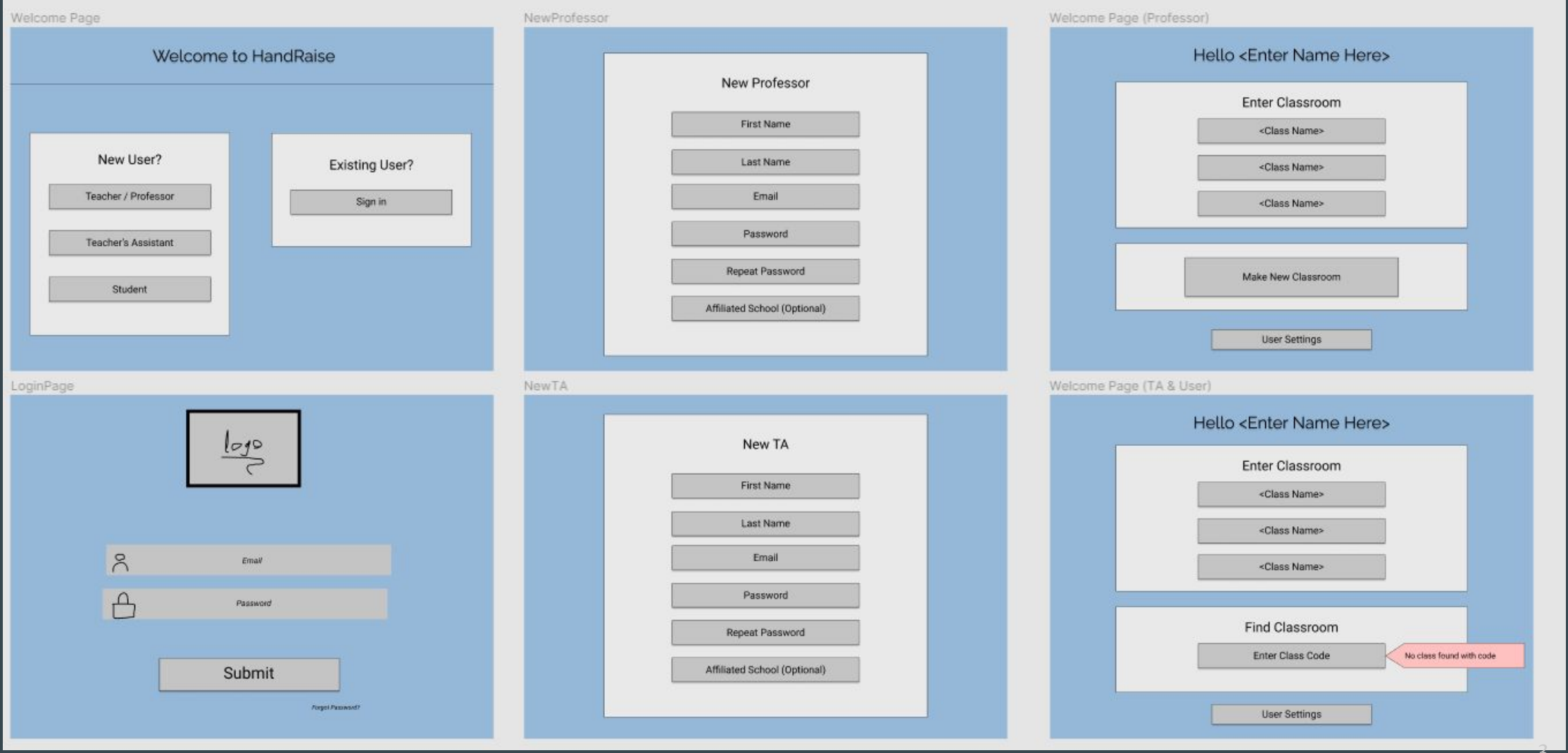
Team: Nick Oswald, Michael Kies, Brian Sayre, Vance Kaw, Daniel King, Robert Walling, Jeremy Tracz

Client: Dr. Md Maruf Ahamed

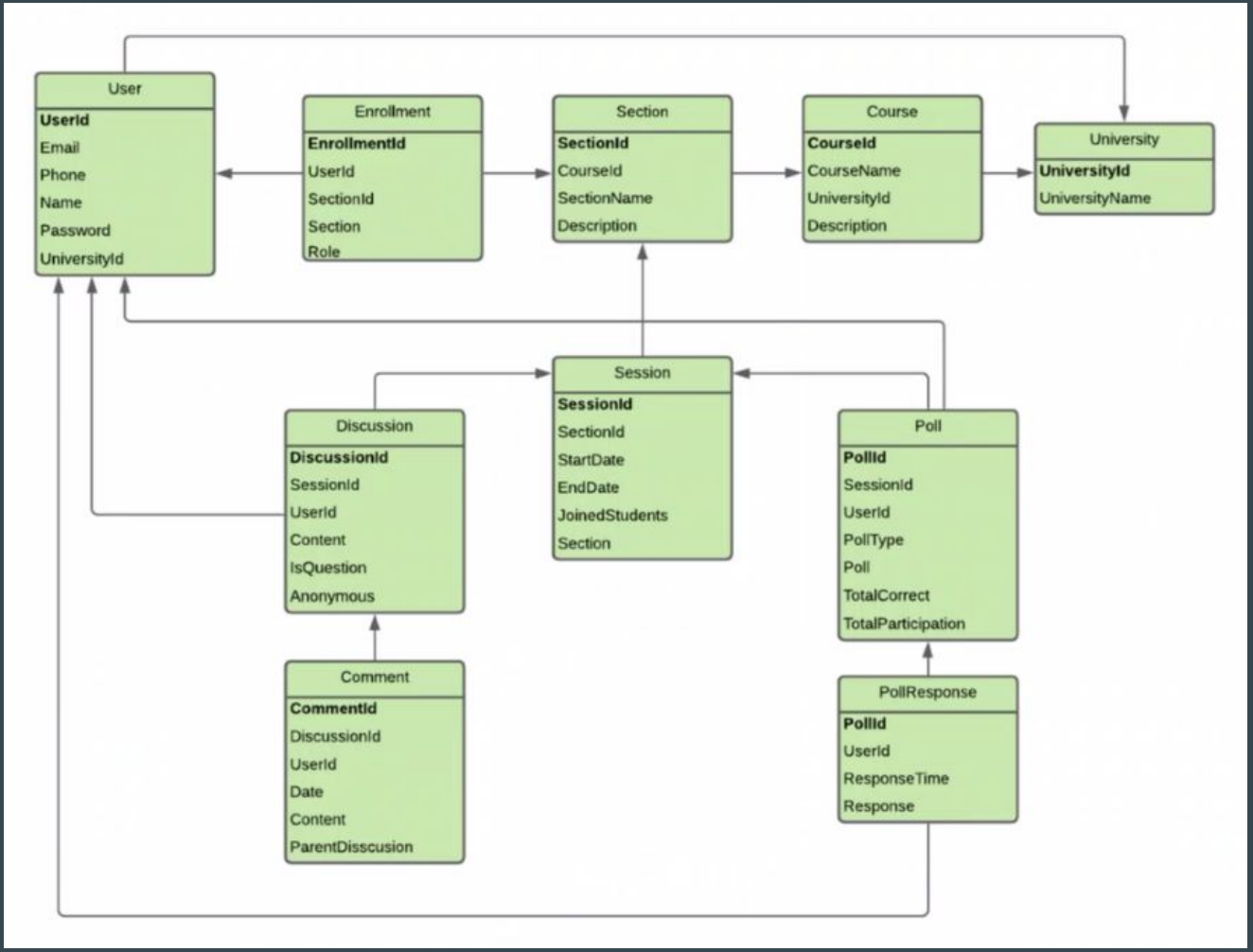
Project Vision

- Our project vision is to design a web-based interactive learning platform for large lectures
- HandRaise allows professors to:
  - Open polls and discussions
  - View engagement statistics
  - Include classroom polls, quizzes, and more and save student responses
- HandRaise allows TAs to:
  - Participate in polls
  - Participate in discussions
  - View restricted engagement statistics
- HandRaise allows students to:
  - Ask questions anonymously/by name
  - Participate in polls
  - Participate in discussions

Conceptual/Visual Sketch



Conceptual Entity Relationship Diagram

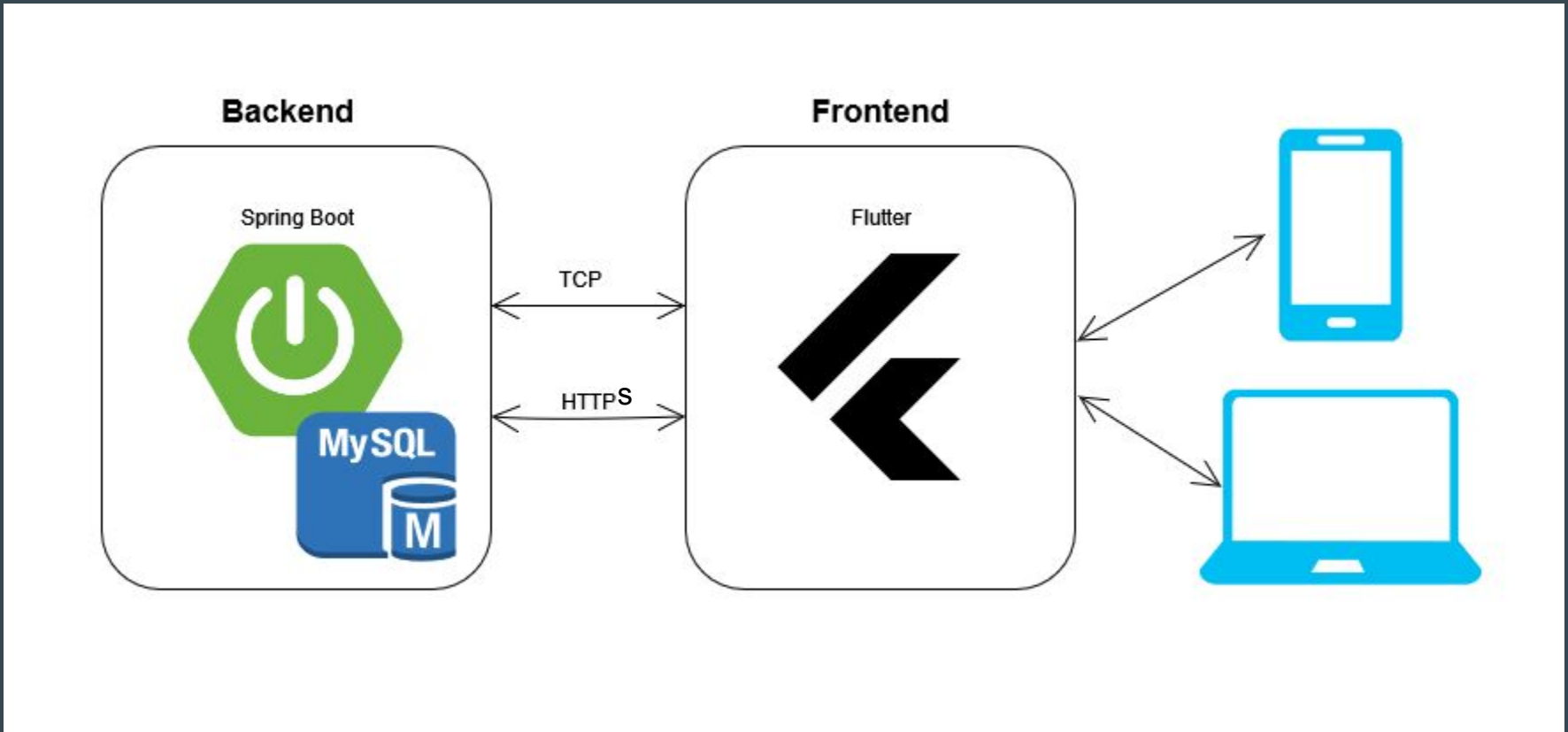


Requirements

- Functional Requirements
  - Ability to run on different browsers and OS.
  - Handle up to 2000 concurrent users. (Estimated using the top five largest lecture halls at Iowa State)
- UI Requirements
  - Front end shall be intuitive and easy to navigate for every user.
  - Web pages should be consistent across the site. Buttons, navigation aids, and other data should have the same feel and location as previous pages
- Realistic Constraints
  - Maintain security with multiple levels of access.
  - Project should be testable by anyone working on it.
- Resource Constraints
  - All code shall be either archived, deleted, or pushed to the dev branch each week. Once working, code will be pushed to master.
  - Project must be completed within 1,500 person hours.

System Design

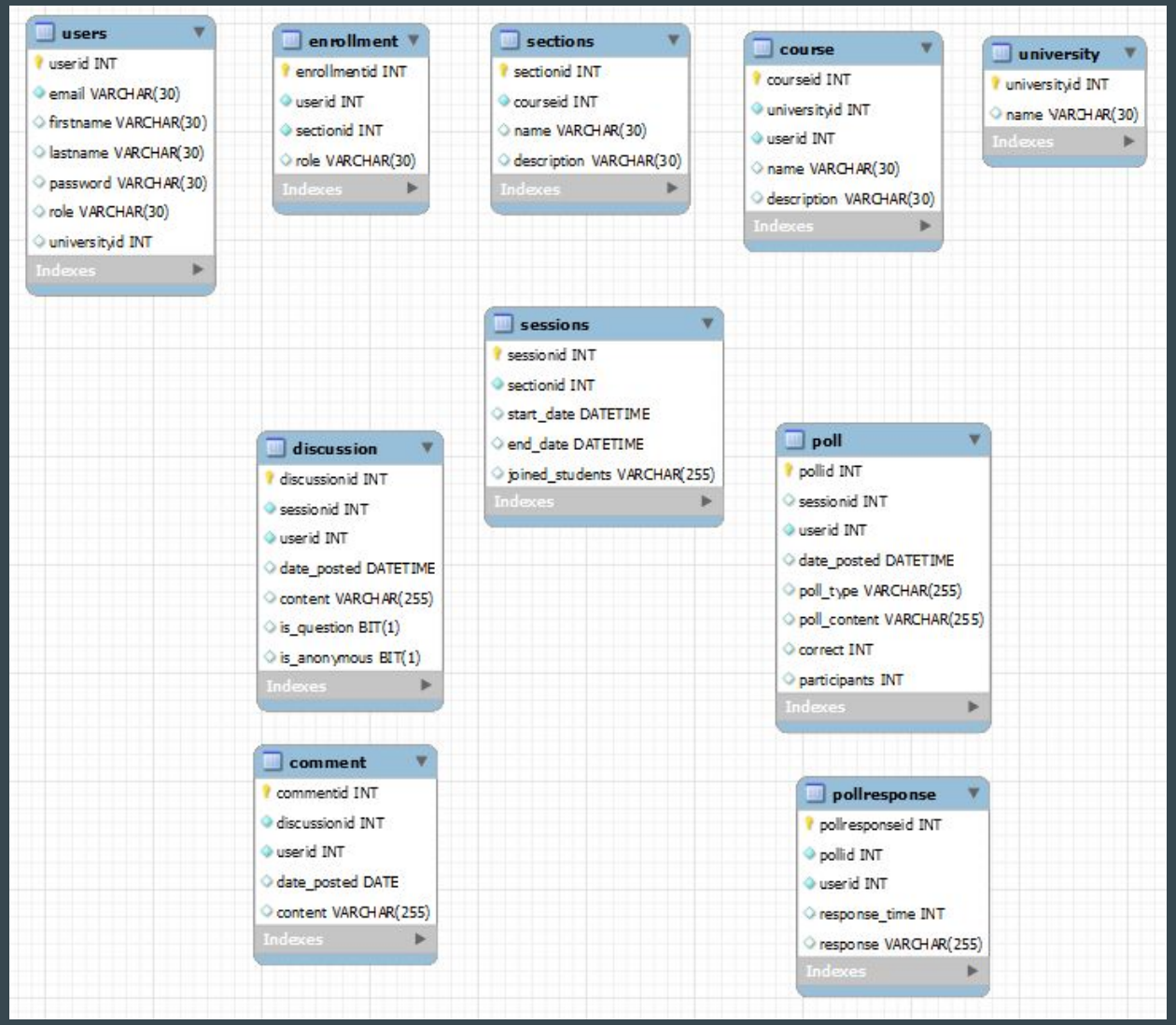
System Architecture



Tech Stack

- Flutter
  - Frontend app development
- Spring Boot
  - REST API
  - Used to access all of the data
  - Websockets to enable live poll features
- MySql Server
  - Stores the data

Backend Entity Relationship



Prototype Implementations (Demo)

Design Complexity

- HandRaise requires instant responses from the server after a user's input
- The product must have a professional appearance and be intuitive to students
- HandRaise must comply to security standards - all user information must remain private and confidential
- HandRaise seeks to go above and beyond accessibility standards in order to be as inclusive as possible

Project Plan

- Frontend
  - Flutter
    - Create UI mockup
    - Decide how to interact with backend
    - Display data
- Backend
  - Spring boot
    - Setup database architecture
    - Setup server
    - Encrypt passwords
  - MySQL
    - Design tables
    - Implement relations

Project Plan - Risks/Mitigations

Risks	Mitigations
ISU servers may not be able to handle 2000 concurrent users	Optimize our application
ISU servers may not be able to archive all data	Add features to limit the amount of data stored or how long data is stored
May not have enough person hours to implement all desired features	Plan which features should be done first to ensure the main use cases are satisfied
Students or professors may not want to use our application	Advertise our application to both groups
Sensitive data leaks	Encrypt data traffic. Hash email and passwords. Limit data access between roles.

Project Milestone, Metrics, and Evaluation

- Design phase is complete
- Frontend to Backend round trip
- SQL database is set up
- Backend connected to database
- 2000 concurrent users connected
- Implement features
  - Question and replies are posted
  - Conversations archived
  - Polls
  - etc...

Testing Plan

- Integration testing - used to mock up fake users and ensure entities and table relationships work as intended
- System testing - Use unit testing, integration testing, widget testing, manual testing, and more
- Regression testing - All branches and new code must pass all prior unit tests
- Acceptance testing - We continue to check in with our client about additional requirements
- Security testing - Use algorithms to encrypt data and use JUnit testing to verify the security of these algorithms

Frontend Testing

- We will test our frontend using Flutter widget testing, unit testing, integration testing, and more as we complete our app development
- All features that we add will be unit tested for functionality
  - Page switching
  - User submissions
  - User logins
  - Error outputs
- We have been doing manual testing at the moment, as the app is still early in development.

Backend Testing

- We will be using Mockito to mock objects as part of our backend unit testing.
  - We can mock objects to test the functionality of several controller and repository methods
- We will use JUnit for unit testing the Spring Boot application
- We will also be writing integration tests for our Spring WebSocket Endpoints for the chat feature of our app.

Conclusions

- We are slightly behind (1 - 2 weeks) of our schedule according to our Gantt chart.
  - Learning Flutter and other frameworks
- We plan to get the chat functionality finalized first, and then we can begin on the many options available for each profile.
  - Creating classrooms
  - Joining specific classes
  - Statistics for users
  - etc.
- However, we will continue refining the app, including testing, during second semester
- If we finish early we will start to refine our mobile application

Thank you!
Questions?