Design Document: Requirements

1.1 Requirements

Functional Requirements:

- Have the ability to run on different browsers and OS
- Code should be testable
- Code should be well documented
- Students are able to perform a hand-raise after entering in a brief question/comment
- Professors are notified whenever a hand-raise occurs during lecture
- Professors are able to clear a hand-raise after it is dealt with
- Students are able to post questions with file attachments under a course
- Professors are able to post polls under a course
- Professors are able to post announcements with attachments under a course
- All users are able to reply to post discussions, if open
- Students are able to respond to polls
- The system displays poll results after a poll closes
- Professors and TAs are able to view detailed poll participation after a poll closes
- Students are able to perform all actions anonymously to other students
- Professors are able to reveal the identity of anonymous students
- All users are able to view every existing post under a course
- Professors and TAs are able to delete posts and replies
- All users are able to view course participation statistics, including the most frequent posters
- All users are able to change their password

Resource Requirements:

- An internet-accessible server with access to data storage
- Enough server capacity to able to handle up to 500 concurrent users (Iowa State's largest lecture hall having a capacity of 431) (Constraint)

Aesthetic Requirements:

- Should look like a simple tool to use (complexity is hidden to users unless enabled)
    - Adequate spacing between posts
    - Only a small number of options to interact with a post

- Options and information should only be visible if contextually applicable and significant (e.g. posts' discussions should not be visible when looking at a list of all posts)
- Minimal, simplistic, clean interfaces

User Experiential Requirements:

- Design should be intuitive and easy for users to use
- Should give the impression of having all functionality in one central location, but without an overwhelming number of functions
- No downtime/long loading time between posting and seeing the post
    - Meaning quick updates to avoid users wondering if their post posted correctly

Economic/Market Requirements:

- Our market is comprised of students and educators
- Needs to be a low to zero cost application; free to use and cheap to host
- Should provide a better educational experience in the classroom for our market

Environmental Requirements:

- The application should make for a safer learning environment
- The application will add a layer of privacy to classroom interaction promoting people to reach out with questions.

UI Requirements:

- UI should have a very simple flow, easy to navigate and understand for all users.
- UI should have a modernized design to improve the overall user experience

1.2 Engineering Standards

- IEEE 1016: Software design description
    - We will use this standard when planning out our project. We will make data driven decisions to help create the best product. We will create diagrams for our architecture to help give visuals of our projects setup
- IEEE 1028: Software Review
    - We will perform regular code reviews to ensure the quality of our application. This standard talks about having personnel, users, customers, and other interested parties review the code as well as the product to ensure quality which we will do
- ISO/IEC/IEEE 26515:2018: Developing information for users in an agile environment
    - We will use agile during the development of our application. We chose agile so that we can work on many tasks of the project in parallel and have constant communication with our stakeholders allowing us to make changes as needed
- IEEE 9274.1.1- JavaScript Object Notation (JSON) Data Model Format and Representational State Transfer (RESTful) Web Service for Learner Experience Data Tracking and Access
    - We will use JSON notation for our communications between frontend and backend. We chose this because both of our frameworks are JS based which makes interacting with JSON very easy
- IEEE 7002- Standard for Data Privacy Process
    - We will follow this standard in order to safeguard the answers that students provide. We chose this because, even though their information isn't overly sensitive, we still need to be cautious and safeguard their anonymity as much as possible