4 Design

4.1 Design Context

4.1.1 Broader Context

Our project hits a lot of different areas. The table below describes a few such areas.

Area	Description	Examples
Public health, safety, and welfare	**How does your project affect the general well-being of various stakeholder groups?** Our project affects stakeholders by improving the quality of lectures for students and improving student interaction for professors. Students are able to make their voices heard and professors are able to get more feedback from students easier.	Our application allows students to ask questions during lecture and additionally the student can ask anonymously which allows all students to make their voices heard during lecture allowing for questions to be answered quickly. Professors are alerted to new questions being asked so they can answer them in a timely manner. Professors can also create polls during class to gauge the class's understanding of material as well as use the feature for attendance or other interactive learning applications. Our chat rooms will be active 24/7 which will help students to have access to help around the clock from their professor, TAs or other students.
Global, cultural, and social	**How well does your project reflect the values, practices, and aims of the cultural groups it affects?** Many members of the engineering profession are often quiet introverted people. These traits can make it hard to ask questions or communicate in large lectures. Our application reflects these traits by helping to create a solution to the lack of communication	Our application will allow for anonymous messaging. Since many students are often nervous to ask questions during large lectures, these students still need to make their voices heard. By allowing anonymous messaging, there is no reason for students to be nervous to speak up or ask a question they may deem as “stupid”.
Environmental	**What environmental impact might your project have?** Our application will require many devices to connect to our application at a time which will inadvertently require a large amount of power especially for lectures of hundreds of people or more. This high power consumption could have a negative environmental effect.	Increasing/decreasing energy usage from nonrenewable sources, increasing/decreasing usage/production of non-recyclable materials Our application could increase energy usage from non-renewable resources but overall the impact of our application will be very little.

Economic	**What economic impact might your project have?** Our application could have a large beneficial financial impact on students	Our application will provide similar functionality to other apps like Top-Hat. Top hat currently requires a paid subscription. Our application could replace top-hat which could save students money from having to buy a subscription. Students are often tight on money so anything helps.

4.1.2 Prior Work/Solutions

Since our application is classified as an interactive learning application, there are two primary comparable applications that ISU Students are very familiar with, Piazza and Tophat. Both of these applications promote classroom interaction and communication. Our application will utilize certain positive components that both tophat and piazza have and also build upon that to create a free and easy to use application for both students and staff.

Pros:

- Get live direct feedback from students/TA's/Professors
- Free to use, no third party service
- Ability to ask/respond to questions during/outside class
- 24/7 access to past lecture discussions
- Can gather statistics for professor/TA's to use for grading purposes

Cons:

- No built in lab/exam feature (tophat.com)
- No interactive textbook (tophat.com)
- No automated attendance/grading (tophat.com)
- No scheduling feature of networking events/interviews (piazza.com)
- No email feature for contacting large companies (piazza.com)
- No connection to canvas for grading (tophat.com)

4.1.3 Technical Complexity

Our project is sufficiently complex because it has both multiple components that must communicate with each other, and challenging requirements that aren't currently met by another solution. Our three main components are the frontend web application, the backend APIs and websockets, and the MySQL database that stores information for the APIs. Additional layers of complexity are added by our choice to use a Docker to deploy our backend application, as well as the fact we must implement both websockets and REST APIs that cooperate and are in sync with each other when providing information to the frontend web application. As mentioned previously, Tophat and Piazza are both somewhat similar to what we hope to build, however neither provides the full functionality that we are aiming for. Our project is unique in that it will combine the real-time polling aspect of Top Hat with the forum-like structure of Piazza to create a solution that we believe will be very valuable.

4.2 Design Exploration

4.2.1 Design Decisions

- *Design Decision:* **Frontend and Backend Frameworks**
  The underlying frameworks that the frontend and backend are built on directly influence many details of the project. The frameworks should be versatile enough to allow for any current and future functionality that may be required of the application, but also efficient enough that development will not get blocked or throttled by the frameworks or by maneuvering them to fit our needs. To accomplish this, the frameworks should be relatively simple to use, but easily extensible to add any specific functionality.

- *Design Decision:* **Page Layouts**
  User experience and look-and-feel are both vitally important aspects of a successful software product. The user experience should be smooth and intuitive; no feature should feel obtuse or require training to use. The look-and-feel also should be comfortable and visually appealing. Incorporating both of these aspects into the project begins with careful design of the page layouts. The page layouts define what the user sees, how the user moves through the application, and how the application's features are presented.

- *Design Decision:* **API Definitions**
  In order for the frontend and backend to work together, there needs to be an agreed-upon set of interfaces the backend provides and the frontend uses. These interfaces will serve as the only form of communication between the two sides of the application, so it is important that these interfaces are established early and before major implementation work begins.

4.2.2 Ideation

For **Page Layouts**, we considered a number of different options to set this up.

1. We first discussed having the exact same layout for all of our users, just some users would have options disabled. In this case we are considering the professor having the same page layout as the student, but the student would not have the option to reveal the anonymous user's name or see the list of participants in a discussion.
2. We discussed having a very basic page that only contains basic core functions. In this case we think the users would be able to navigate through the website with ease, but we would also lose some functionality if we were to keep only the core functions.
3. We then considered having separate pages for all users. In this case, the student's view would be a completely separate page, filled with separate components, from the professor and same goes for the TA. This design option, in theory, would be much more straightforward when only considering one user per page and only considering the functionalities they would need. The issue with this option is that we are copying over a lot of the code per page.
4. We considered having a more intricate page layout, which would include the options to change it to dark mode, rearrange the classes and other functionalities that will help the user to customize the pages. The issue with this, is that we would have to add all alternative page layouts for each view and each user. This is a farther reach for us, and in turn we chose to not consider this for the first iteration of the MVP.
5. We considered implementing each page layout as a whole instead of breaking it up into components. This way it would be easy to reuse the page layout and keep it consistent throughout, but this would also mean a bunch of repeated code which will lead to bugs later on.

This would also make it more difficult to make simple changes to things that could be considered components (like the top nav bar and such). So we chose against this option.

4.2.3 Decision-Making and Trade-Off

We ended up choosing option 1. We took into account many different factors, some being design based, and others over a series of constructive feedback from our project manager based on our initial evaluations. We went with this option because it ended up being the most practical choice to meet our specific project needs. This project has a lot of complexities to it, so we ended up going with a very streamlined design that will get a well rounded project out in the timeline we have by setting attainable goals to meet the requirements.