3 Design

3.1 Design Context

3.1.1 Broader Context

Area	Description	Examples
Public health, safety, and welfare	The project (HandRaise) will increase welfare by increasing the opportunity for students to ask questions in class, hopefully increasing their knowledge in the subject and helping their GPA.	A student is too nervous to ask questions in class with over one hundred people, so they ask anonymously through HandRaise.
Global, cultural, and social	Our project reflects the values of students by giving them an extra opportunity to participate in larger classes where they would otherwise be more hesitant to reach out.	For example, students with disabilities who are unable to participate in class traditionally will be able to ask/answer questions through HandRaise.
Environmental	The cost of electricity for the servers that run HandRaise and the devices that access it will affect the environment based on the method the electricity was obtained.	If electricity was obtained by burning fossil fuels, there is an impact on the environment because of that.
Economic	Iowa State University can provide this application as a resource for students and faculty (i.e. this will be included with the cost of tuition). This will hopefully lessen (at least in some small part) the financial burden of higher education.	Instead of requiring students to purchase TopHat, classes can use our free application.

3.1.2 User Needs

Professors

- Professors need a way to create different types of polls and quizzes that can/will be answered by students
- Professors need to be able to view student questions and either resolve them in class or start a discussion thread
- Professors need a way to gauge student participation and attention level in class
- Professors need a way to see student participation metrics and grades on past quizzes.

Teachers Assistants

- TA's need a way to respond to questions that students ask in class without interrupting the lecture
- TA's need a way to view and grade polls and quizzes that the professor publishes and creates on the fly

3.1.2 User Needs (Cont.)

Students

- Students need a way to ask anonymous questions in a large lecture hall.
- Students need a way to engage in a class even when a professor can't interact with them personally.
- Students need a platform where they can interact with other students in a class in a controlled environment.

3.1.3 Prior Work/Solutions

Pros

- Extend the functionality of the current products
- Get direct feedback from students about what we could do to improve our product and add those desired features
- Students don't have to buy the third-party service
- The ability to respond to another student's question is similar to Piazza but can be done more informally (like during lecture).

Cons

- There will be no Interactive Textbook like Top Hat (tophat.com)
- TopHat has a built-in Exam feature that will not be available to HandRaise (tophat.com)
- Labs will not be available in HandRaise (tophat.com)
- We will not have 24/7 support for HandRaise (tophat.com)
- There will be no automated participation scoring (packback.co)

3.1.4 Technical Complexity

- The project requires fast responses from the server after a users input with hundreds of concurrent users
- To compete with currently used software the product must look professional and be intuitive to the students
- Storing user data means we have an extra responsibility of maintaining security standards for our product
- There is a challenge in building a product that is easily maintainable and extendable in the future

3.2 Design Exploration

3.2.1 Design Decisions

List key design decisions (at least three) that you have made or will need to make in relation to your proposed solution. These can include but are not limited to, materials, subsystems, physical components, sensors/chips/devices, physical layout, features, etc.

Dedicated ISU Servers - these servers will be dedicated to our service. The client-server architecture will be used so that many clients (user groups) can be connected to the server in-class session.

React.js - this frontend language will be used as it is particularly good at dynamically displaying webpages that receive new information (such as a response to a question). It can also be easily ported and code can be reused in mobile applications.

SpringBoot / Java - This will be the language/framework that will be used for the backend side of HandRaise. This allows us an easy connection to SQL and the server. Along with that, our group is very familiar with Java, so having it be the base language makes the project easier to implement.

MySQL - HandRaise will be using MySQL for the database of the project. This will allow us to easily catalog each user, their account, and many other variables that we need to keep track of throughout the course of the project.

3.2.2 Ideation

Lotus blossom for frontend decision

Very popular (i.e. lots of online resources)	Open-source	Most Robust	Used before by team members	Made by Google	TypeScript	Open-source	TypeScript / JavaScript	Support libraries
Great for high traffic	**React**	Made by Facebook	Good for large applications	**Angular**	Steep learing curve	Light weight	**Vue**	Smaller community
Dependable	JavaScript	Good for small teams						Component-based
			React	**Angular**	**Vue**			
				Frontend Framework				
			Ember		**JQuery**			
Fastest development framework	Two-way data binding	Good for extensive projects				It has Plugins	It comes with an MIT license and is Open Source	jQuery is flexible and fast for web development
Not fit for small development teams	**Ember**	Preliminary cost is high				JQuery javascript file required	**JQuery**	Large library
Difficult learning curve	Easy add-ons	Very large community				Functionality maybe limited		

3.2.3 Decision-Making and Trade-Off

Selection Criteria	Criterion Weight	React		Vue		Angular		Ember		jQuery	
		Score	Total	Score	Total	Score	Total	Score	Total	Score	Total
Popularity	0.1	5	0.5	3	0.3	4	0.4	2	0.2	1	0.1
Learning curve	0.3	2	0.6	4	1.2	2	0.6	2	0.6	3	0.9
Speed	0.2	4	0.8	5	1	4	0.8	3	0.6	4	0.8
Feature set	0.1	5	0.5	3	0.3	4	0.4	3	0.3	1	0.1
Team Size	0.3	5	1.5	3	0.9	4	1.2	1	0.3	1	0.3
Total:	**1**		**3.9**		**3.7**		**3.4**		**2**		**2.2**

We chose to go with React as our frontend framework. To make this decision we took into account popularity, the learning curve associated with the framework, the speed, the feature set, and the recommended team size. As you can see above, we used a weighted decision matrix to help quantify those criteria.

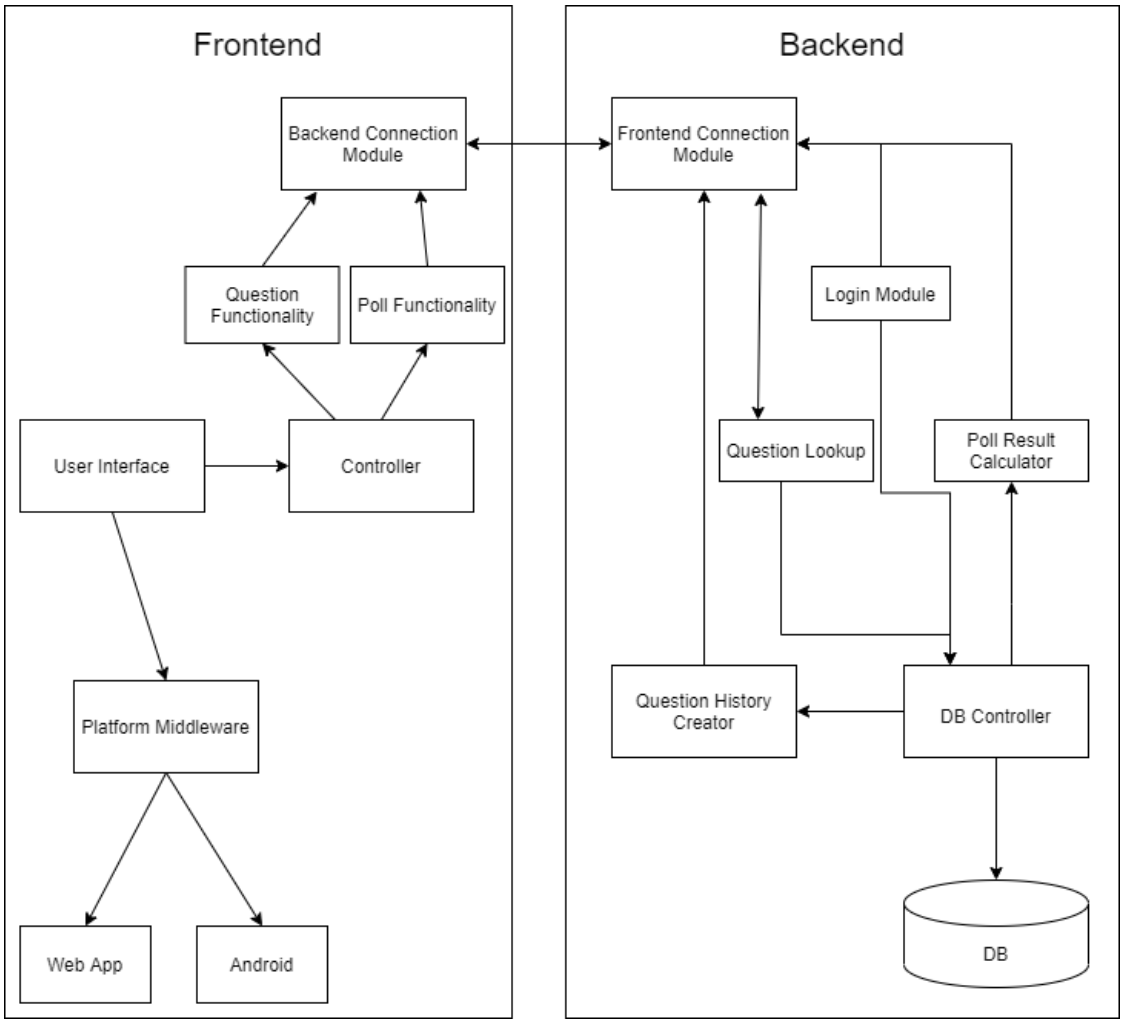
3.3 Proposed Design

We have used Springboot and SQL in the past in classes such as ComS 309. It has a lot of resources available online for all of our members to learn, and experiment with different methods of backend development.

Our team has one member who has used React in the past. React is really good for building components that can be connected to create a whole application. It is great for modularity and creating reusable components, which is useful for frontend development and interface creation.

Other than technical designs, our team has begun work on the UI and database table mockups to be expanded upon throughout the course of the project.

3.3.1 Design Visual and Description



3.3.1 Design Visual and Description (Cont.)

Our project is broken into two main ideas: the frontend and the backend. The front end is responsible for allowing all users to be able to ask questions, respond to polls, and see past questions and polls. Separate modules are needed to send and fetch data to the backend, create the view the user will see, and allow the user to interface with the system. The backend is responsible for storing data in the database and finding any information in the database. Lists of previous questions & answers along with poll results need to be sent back to the frontend via the frontend connection module.

3.3.2 Functionality

Our program should be as user-friendly as possible. Design decisions and implementations should be hidden from the user, so all they need to worry about are the questions and answers being passed through our system.

Our program will start with a login page. This will allow professors, TA's, and students to log in to their respective accounts. From there students and TA's will be able to join classes set by the professors and view discussions, questions, and anything else that is made in development.

Students will be able to join a class using a class code. From there, they will be able to ask questions, like questions that they want to know as well, and answer polls the teacher releases. The questions students ask can be asked anonymously. This way, any nervous student can still ask the teacher questions without sharing their name with the class.

Professors, on the other hand, will be able to make a new class, get a class code for students to join, and monitor almost all of the information being sent into the class. Professors will also be able to see the most liked questions so that they can answer the questions that most students are struggling with. Along with this, they will be able to poll the class for attendance or participation purposes. Professors will be able to see all the statistics that occur during the class.

There is no visual made at the moment. However, our team is working on a UI mockup this week.

How well does the current design satisfy functional and non-functional requirements?

Strong separation of concerns allows for the project to be modular and add more functionality in the future. The use of a database controller will allow for concurrency for thousands of users. The connection between the front and back ends will include encryption protocols to allow for security with login information.

3.3.3 Areas of Concern and Development

Our primary concerns are the following:

- Iowa State server's capacity to store all necessary files.
- Privacy of users' data and information.
- To work efficiently with minimal lag times.

Immediate Plans and Questions are as follows:

- Developing a backend that can handle a load of a couple of hundred users with sufficient response times.
    - Creating a sound design based on previous experience and researching the tools we will be using.
    - Load testing on our servers frequently during development to ensure we are meeting the requirements
- Making sure that the Iowa State server can handle the large number of archived discussions that will need to be saved for the project.
    - Research ways to archive the data that will have a minimal data cost to save in the database.
    - Find how much data the servers are able to handle so we have a guideline.
- Having a secure backend and frontend with minimal risk of data breaches and having an architecture that ensures privacy of our users' data (i.e a secure application).
    - Researching encryption and good security practices for online applications
    - Having a continuous development environment so that any security concerns or bugs may be found and dealt with quickly and efficiently.
    - Setting up a good testing environment so bugs and issues may be routinely looked for and resolved.

3.6 References Used

https://www.monocubed.com/best-front-end-frameworks/

Advantages and Disadvantages jQuery – LORECENTRAL

Jquery Disadvantages And Advantages | JScripters.com: developing a site with Javascript

https://tophat.com/

https://help.packback.co/hc/en-us

https://en.wikipedia.org/wiki/Vue.js

https://www.simform.com/blog/angular-vs-react/

https://www.monocubed.com/angular-vs-react/

https://www.sitepoint.com/most-popular-frontend-frameworks-compared/

https://www.ideamotive.co/blog/best-frontend-frameworks