HandRaise

Design Document

Team Name: sdmay22-40

Client: Dr. Md Maruf Ahamed

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

Team Email: sdmay22-40@iastate.edu

Team Website: sdmay22-40.sd.ece.iastate.edu

Revised: 11/30/21

Executive Summary

Development Standards & Practices Used

- IEEE 1008-1987 (Standard for Software Unit Testing)
- IEEE 23026-2006 (Standard for Software Engineering – website)
- HTTPS standards
- Java coding standards
- Flutter coding standards
- GIT standards

Summary of Requirements

Design a web-based interactive learning platform (LMS) for large lectures

Allow professors to open polls, open and participate in discussions, and view student engagement statistics

Allow TAs to participate in polls and discussions and view restricted engagement statistics

Allow students to ask questions anonymously or by name and participate in polls and discussions

Applicable Courses from Iowa State University Curriculum

Com S 227/228 – Basics of object-oriented programming, data structures

Com S 309 – Software development standards

S E 417 – Software testing

Engl 150/250/314 – Lightning talks, weekly assignments, team communication

New Skills/Knowledge acquired that was not taught in courses

Flutter – the frontend framework for this project

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Team

1.1 Team Members

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

1.2 Required Skill Sets for Your Project

Web development, app development, frontend development, backend development, programming, creating concrete system design patterns, basic security, software testing, documentation and presentation skills, and more.

1.3 Skill Sets covered by the Team

Due to the nature of the project, all team members will contribute towards web and app development, programming, testing, documentation, presentations, and more. Therefore, our team has divided primarily into two project groups, frontend and backend, but our project contributions are not necessarily restricted to these groups.

Frontend team: Robert Walling, Nick Oswald, Jeremy Tracz, Daniel King

Backend team: Brian Sayre, Vance Kaw, Michael Kies

1.4 Project Management Style Adopted by the team

Waterfall + Agile for frontload planning, incremental development, and allowing for quick changes based on the current iteration of the application.

GitLab for version control and managing the project development.

1.5 Initial Project Management Roles

Nick Oswald: Facilitator, Initiator
Michael Kies: Initiator, Explorer
Brian Sayre: Facilitator, Arbitrator, Explorer
Vance Kaw: Arbitrator, Explorer
Daniel King: Explorer, Information provider
Robert Walling: Gatekeeper, Facilitator
Jeremy Tracz: Arbitrator, Explorer

2 Introduction

2.1 Problem Statement

Our project looks to increase participation in large class settings. In large classrooms it can be hard for students to ask questions if they are sitting far away or stay engaged with so many other students in a lecture hall. Along with this, being called out in a large lecture hall can be intimidating, so anonymity can help increase overall participation and useful discussions. Also, right now it is difficult for the professors to gauge how involved all of the students are in lectures as well as get valuable feedback on what topics people are struggling with. This app will allow professors to see statistics for class participation, gauge how well TAs and students are performing, and ask students questions to increase class participation. Our app will allow students to feel like they are having one-to-one communication with the professor and TA whether or not they want their name to be displayed to the class.

2.2 Requirements & Constraints

Ability to run on different browsers and OS. (functional requirement)

Handle up to 2000 concurrent users. (Estimated using the top five largest lecture halls at Iowa State) (functional requirement)

Maintain security with multiple levels of access. (realistic constraint)

Code should be well documented and documented on a (minimum) weekly basis. (qualitative)

Project must be completed within 1,500 person hours. (resource constraint)

Project should be testable by anyone working on it. (realistic constraint)

Web pages should be consistent across the site. Buttons, navigation aids, and other data should have the same feel and location as previous pages. (UI requirement)

All code shall be either archived, deleted, or pushed to the dev branch each week. Once working, code will be pushed to master. (resource constraint)

Front end shall be intuitive and easy to navigate for every user. (qualitative aesthetic requirement)

2.3 Engineering Standards

IEEE 1008-1987 (Standard for Software Unit Testing): This standard will help us in our testing phase of the project. This ensures that we are testing our app efficiently by covering as many cases as we can in order to prevent faults in our code. Reference: https://ieeexplore.ieee.org/document/27763

IEEE 23026-2006 (Standard for Software Engineering -- Website Engineering, Website Management, and Website Life Cycle): This falls perfectly into our project plan, as it has good and detailed constraints for the development, use, maintenance, and life cycle of our website.

HTTPS standards: The app will be originally built using html.

Java coding standards will be used for the backend through SpringBoot.

Flutter coding standards: This is an option if Facebook's React Native fits the project well.

GIT standards: Git will be our main hub for connecting code and managing tasks that need completed.

2.4 Intended Users and Uses

Students:

- Sign up
- Ask questions in lecture with text, image, audio, and/or video to be answered by TAs and Professors either anonymously or by name.
- Reply to conversations/discussion
- Participate in polls
- View archived discussions and polls

TAs:

- Sign up
- Reply to conversations/discussion
- View/Grade Responses to poll questions
- View archived discussions and polls
- See student contribution data

Professors:

- Sign up
- Open discussion
- Open polls
- Reply to conversations
- See student and TA contribution data
- View archived discussions and polls

3 Project Plan

3.1 Project Management/Tracking Procedures

Our group is using the project management style of Waterfall+agile. This way, we can use the traits of both styles that will fit our group for the short amount of time we have available for our project. Along with that, a combination of waterfall and agile will allow us to change the design of the project in a faster manner if needed.

Our group will be using GitLab for version control and to track progress. We can create and assign tickets for the week using the software while we manage a master branch of the project.

3.2 Task Decomposition

Frontend task decomposition:

- Create Frontend (ReactJS)
    - Create UI mockup
    - Decide on necessary components
    - Fetch data from back end
    - Display data correctly
    - Send data to back end
    - Add features
        - Users can signup and login with different roles
        - Students can ask questions that are answered by TAs and/or Professors
        - TAs and/or Professors can create polls for students to answer
        - TAs and/or Professors can see archived questions and polls
        - Professors can look up student and TA participation data
    - Write tests and third party review process
        - These will be tests for the code to check its functionality
        - The review process will be through third party people who will make comments on the UI, so we can get unbiased feedback.

Backend task decomposition:

- Create Backend (Spring boot w/ MySQL)
    - Spring Boot
        - Setup database architecture
            - Make decisions on how we need to set up the backend to meet our requirements
        - Setup server architecture
            - Connect with the MySQL tables on the server
        - Encrypted Passwords
            - Passwords of every user will be encrypted when stored in the server table.
        - Write tests (JUnit and Mockito)
            - Decide on what we will be mocking to test our tables.
    - MySQL
        - Design Tables
            - Decide what data we will need to store in MySQL and the relations between the tables
        - Implements tables on given server
            - Create the tables in using MySQL on the school server

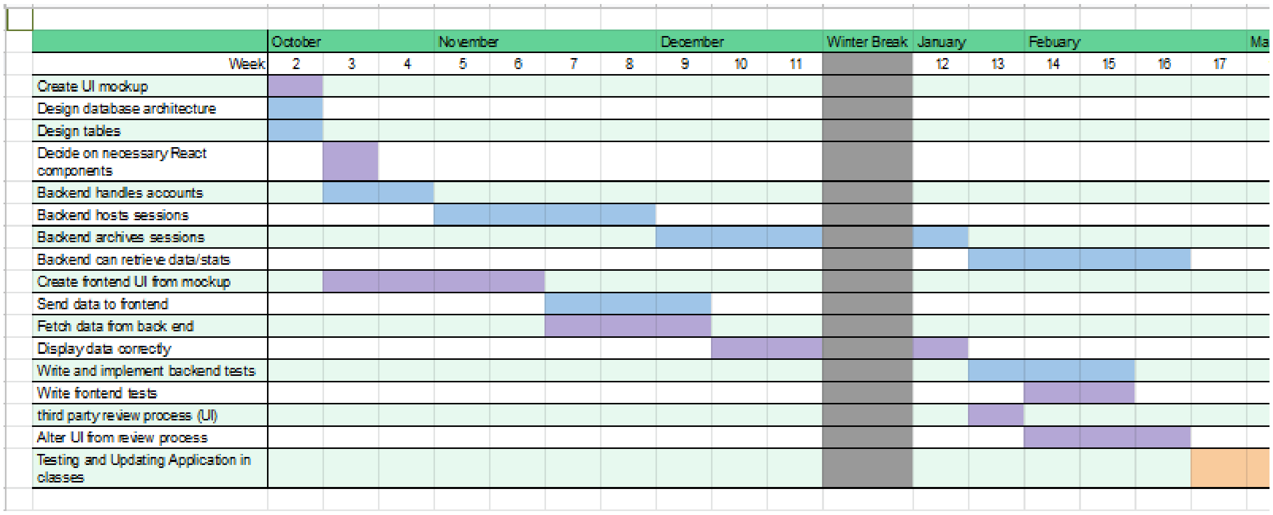
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Our team is also using and updating a Gantt chart to track our project progress. Key milestones include:

- Design phase completed
- First implementation of UI Mockups
- Peer reviewed implementation of UI
- Round trip (front end can communicate with back end)
- Web based
- Android based
- SQL Database tables have been set up and will be in normal form with appropriate granularity.
- Connect Backend to MySQL and get the correct response on a local server.
- 2000 concurrent users connected with no error messages
- Base Features
- Successful question and reply posted with text, image, audio, and video for both
- Successful archive and opening of conversations
- Successful poll and response of users
- Correct response for class statistics for professor, TA, and user data

3.4 Project Timeline/Schedule

Original Gantt chart:



	October			November				December			Winter Break	January		Febuary				Ma
Week	2	3	4	5	6	7	8	9	10	11		12	13	14	15	16	17	
Create UI mockup	🟪										▒							
Design database architecture	🟦										▒							
Design tables	🟦										▒							
Decide on necessary React components		🟪									▒							
Backend handles accounts		🟦	🟦								▒							
Backend hosts sessions				🟦	🟦	🟦	🟦				▒							
Backend archives sessions								🟦	🟦	🟦	▒	🟦						
Backend can retrieve data/stats											▒		🟦	🟦	🟦	🟦		
Create frontend UI from mockup		🟪	🟪	🟪	🟪						▒							
Send data to frontend						🟦	🟦	🟦			▒							
Fetch data from back end						🟪	🟪	🟪			▒							
Display data correctly									🟪	🟪	▒	🟪						
Write and implement backend tests											▒		🟦	🟦	🟦			
Write frontend tests											▒			🟪	🟪			
third party review process (UI)											▒		🟪					
Alter UI from review process											▒			🟪	🟪	🟪		
Testing and Updating Application in classes											▒						🟧	🟧

Our project progress is mostly on track - there were slight delays due to needing to learn the project frameworks. However, we are looking forward to soon having a functional edition of the application and starting testing soon in the second semester of senior design.

3.5 Risks And Risk Management/Mitigation

During project development, we have identified the following primary risks and risk mitigations:

Risks	Risk Mitigation
ISU servers may not be able to handle 2000 concurrent user requests	Optimize our application. Petition for more ISU servers.
ISU servers may not be able to archive all the question and participation data	Add features to limit the amount of data stored and/or the amount of time the data is stored. Petition for more storage on our server(s).
We might not have enough person hours to complete all of the features (either desired or required) of the application	Plan out which features we want to create first based on importance. Once the round trip is done, keep creating (and testing) new features on a weekly basis.
Students and/or professors might not want to use our application	Advertise our application to professors. Advertise to students that our application is free for students.
Sensitive data leak	Encrypt data traffic. Hash emails and passwords. Limit data access between roles.

Users may not know how to use our application	Spend time designing and implementing an intuitive UI. Change based on user feedback.

3.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in total number of person-hours required to perform the task.

Task	Category	Projected hours (Person Hours)
Create UI mockup	Frontend	8 hours
Decide on necessary components	Frontend	8 hours
Fetch data from back end	Frontend	40 hours
Display data correctly	Frontend	40 hours
Send data to back end	Frontend	40 hours
Write tests and third party review process	Frontend	60 hours
Setup database architecture	Backend (Spring Boot)	15 hours
Setup Server architecture	Backend (Spring Boot)	20 hours
Encrypted passwords	Backend (Spring Boot)	20 hours
Backend handles accounts	Backend (Spring Boot)	20 hours
Backend hosts sessions	Backend (Spring Boot)	80 hours
Backend archives sessions	Backend (Spring Boot)	60 hours
Backend can retrieve data/stats	Backend (Spring Boot)	50 hours
Write tests	Backend (Spring Boot)	60 hours
Send data to frontend	Backend (Spring Boot)	10 hours
Design tables	MySQL	8 hours

Implement tables on given server	MySQL	15 hours
Implement Server logic to handle requests	Backend (SpringBoot)	15 hours

3.7 Other Resource Requirements

One resource we will need is ISU servers. We are provided with one server to work with, but if the project expands into something more permanent or is used campus wide, additional servers will need to be provided.

During the product testing phase, we are planning to require additional resources (devices/mock users/etc) in order to ensure the app can meet client demand.

4 Design

4.1 Design Context

4.1.1 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.

4.1.2 User Needs

Professors:

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

Teaching Assistants:

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

Students:

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

4.1.3 Prior Work/Solutions

Due to the fact that the HandRaise application falls in the same category as a learning management app (LMS), there are a number of previous solutions in the space. The primary two comparable applications include TopHat, PackBack, and Piazza, which all attempt to promote class communication. However, the HandRaise application differs from the elements of all these applications in several ways, and builds upon the positive features of each of these applications to create a free, easy-to-use application for Iowa State University students.

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)

4.1.4 Technical Complexity

Our project requirements include several features which will increase the technical complexity of the project. The primary cause for this complexity is the requirement that all messages, polls, and responses in the app need to be nearly instantaneous, and instructors need to be able to view student engagement statistics while still keeping the data secure. Some notable factors which increase technical complexity are listed below:

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

4.2 Design Exploration

4.2.1 Design Decisions

Three design decisions include which resources, languages, and frameworks we are using to build our application. These design decisions include:

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.

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.

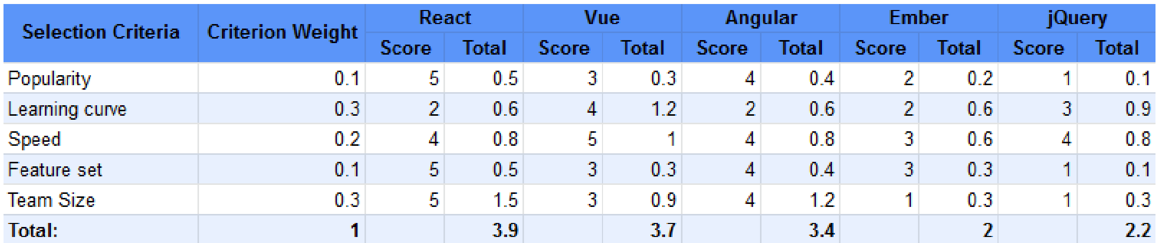
4.2.2 Ideation

One design decision was our frontend framework, for which we initially chose React, but later transitioned to Flutter. The updated lotus blossom for our decision is displayed below.

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 learning curve	Lightweight	**Vue**	Smaller community
Dependable	JavaScript	Good for small teams						Component-based
			React	**Angular**	**Vue**			
				Frontend Framework				
			Ember	**Flutter**	**JQuery**			
Fastest development framework	Two-way data binding	Good for extensive projects	Fast development framework	Easier to learn than other options, but still includes powerful features	Flexible and fast for web development, including the ability to adapt to different screen sizes	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	Practical for small development teams	**Flutter**	Android studio compatibility	JQuery javascript file required	**JQuery**	Large library
Difficult learning curve	Easy add-ons	Very large community	Faster to develop and maintain a web application	Little ios support	Can be difficult to design the UI without experience	Functionality maybe limited		

4.2.3 Decision-Making and Trade-Off

Initially, 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. Pictured below is the weighted decision matrix to help quantify those criteria.



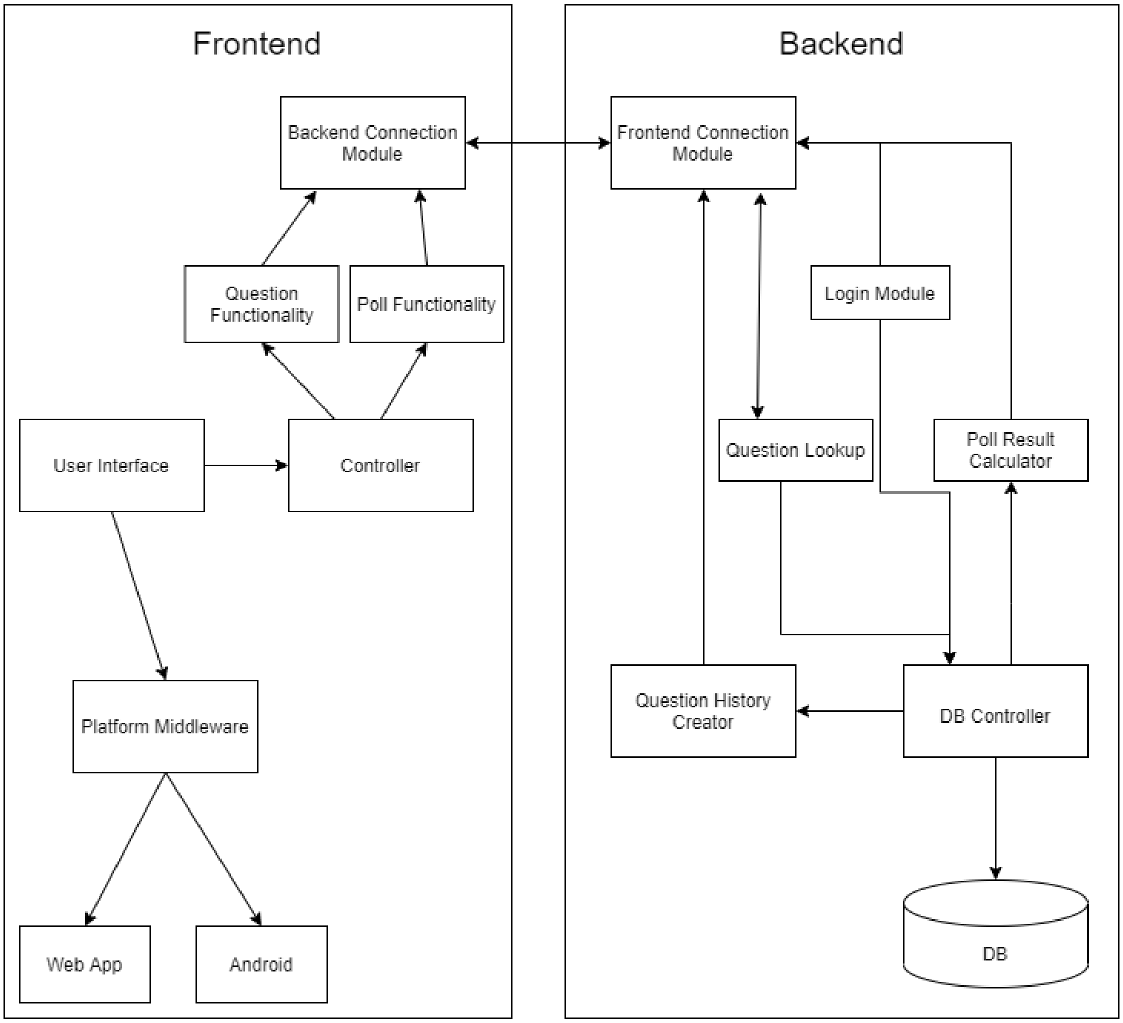
Selection Criteria	Criterion Weight	React		Vue		Angular		Ember		jQuery	
		Score	Total	Score	Total	Score	Total	Score	Total	Score	Total
Popularity	0.1	5	0.5	3	0.3	4	0.4	2	0.2	1	0.1
Learning curve	0.3	2	0.6	4	1.2	2	0.6	2	0.6	3	0.9
Speed	0.2	4	0.8	5	1	4	0.8	3	0.6	4	0.8
Feature set	0.1	5	0.5	3	0.3	4	0.4	3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

However, after several weeks and additional research, we determined that Flutter would be the more effective option for frontend development based on the fact that it was most compatible with web and android development, and would hopefully be easier to learn. This was an important criteria due to the fact that none of the frontend team members had previous experience with Flutter or many of the other options in the decision matrix used originally. Flutter returned the following scores for the decision matrix: Popularity, 4, Learning Curve, 4, Speed, 4, Feature set, 5, team size, 5. This made it competitive with our top options from the original decision matrix.

4.3 Proposed Design

So far, our team has worked on implementing the critical features of the application, including login and signup features for three levels of users: professors, TAs, and students, including three different user interfaces for each of these users. In addition, our group has worked to implement the chat functionality and is working towards the polls feature which will allow professors to track classroom participation. These are established by adding controller functionality for HTTP requests. On the backend, the team has implemented tables in Spring Boot and is working on websockets in order to facilitate effective roundtrip communication between the frontend and the backend of the application.

4.3.1 Design Visual and Description



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.

4.3.2 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.

4.3.3 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.

4.4 Technology Considerations

Based on our previous knowledge and course experience, these choices made the most sense for our project. For frontend, we decided to go with Flutter because it allowed for web and mobile development and worked well for groups. However, this decision did require us to learn the language, which was one setback. For backend, using MySql and Springboot allowed us to use the skills and technology learned in Com S 309 and other previous classes.

4.5 Design Analysis

We will complete this section once we have finished implementing our design.

4.6 Design Plan

5 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The testing plan should connect the requirements and the design to the adopting test strategy and instruments. In this overarching introduction, given an overview of the testing strategy. Emphasize any unique challenges to testing for your system/design.

5.1 Unit Testing

Frontend:

- We will unit test our Flutter frontend using the standard Flutter testing, which includes unit testing, widget testing, and integration testing.
- Every feature that we add will be unit tested for each of its functionalities. For example:
    - Page switching
    - User submissions
    - User login
    - Error outputs

Backend:

- We will be using Mockito to mock objects as part of automated unit tests.
    - Mocking objects created through mockito such as getAccountID to test the functionality of the controller and repository methods.

- We will also test the service part of our unit testing.

5.2 Interface Testing

We will be using Postman for simulating the connection between backend and frontend. This is critical to make sure that speedy development continues while both front and backend are being built initially. Along with that, JMeter and SoapUI are two other tools that have the ability to test API and other interfaces. If we need more tools for testing the interfaces, these two may help us out in the future.

When testing the combination of multiple interfaces, error handling and mockito will be very important. These will allow us to see many of the problems that arise while connecting the interfaces together. Debugging becomes much easier if well made mockito tests are used.

5.3 Integration Testing

Frontend to backend. Round-trip testing.

- Round-trip testing is easy to tell if it is working, as we will have error handling that will catch anything that goes wrong.
    - For example, if there is a wrong email address, there will be an error returned. Or, if the frontend cannot get information from the database, there will be an error as well.
    - Another example would be any CRUD features that we implement.
- The IDE debugger will be used a lot during the testing phase so that we can see exactly where there are issues from the connection.
- This step is critical as if there are problems getting information from the backend, the frontend cannot continue making features for the project.

Frontend:

- We will use interface testing for the frontend so that we can mock fake users to sign into that do not directly come from the database. This way, if there are any problems with the round-trip testing, the frontend can still develop features without losing too much time.
- This step is critical since each page needs to be working correctly and displaying the given information, so having a mock user will speed up the process significantly.

Backend:

For the integration testing, we will be testing our Spring Boot Application.

For instance, we will need to create an account database for the login information of the user. In the account database, we will be testing the user type which includes the id, email, phone, name, and role.

Part of the testing we will need to do is making sure the entities and tables that we intended to have relationships actually do and they work as intended.

The tools we will be using are Spring Boot Test and JUnit testing

5.4 System Testing

Initially, we will rely on widget and unit testing for frontend, and mockups for backend testing. We will be utilizing manual testing with volunteers in order to get responses that a computer will not be able to give. Such as how the flow of the program feels or whether the program looks good or not. These are very important criticisms that will help give us a more polished program.

5.5 Regression Testing

Part of our requirements for merging branches with new features is that they will need to pass all prior unit tests developed. This can be achieved using CI/CD on GitLab. Along with that, as stated in a previous document, we will be merging all changes into a develop branch before merging the final changes into master. This way, the master branch will always be working and we won't need to back track our progress if an unexpected error occurs. In particular, we need to test the security of our system and the accuracy of data being stored.

5.6 Acceptance Testing

We plan to work closely with our client to ensure that both functional and non-functional requirements are being met. We will verify our testing with our client on a regular basis to ensure that our testing meets our client's standards. We will also regularly perform regression testing after making significant changes to HandRaise to ensure it does not lose any desired functionality.

We will also check with our client to see if he has any additional requirements or recommendations, and include these in our testing as well.

5.7 Security Testing (if applicable)

Security testing will be a large part of our project, since students will be logging into the program using their netid and password. We will write algorithms to break our encryption algorithm and ensure that no essential data is viewable to users.

We will also plan to use fuzzing with our testing to ensure that unexpected inputs, whether to the login page or otherwise, do not compromise the security of the system.

We will also implement JUnit tests for our Spring Boot application that make sure users can only hit the API endpoints they are authenticated for and they don't have access to data they shouldn't.

5.8 Results

So far, as we transition out of the design phase, our testing has been relatively basic in order to verify performance of the application. However, as we continue implementing the design, our tests will increase in complexity, and as we complete real-life testing of the application, we will have more data and more information to report.

6 Implementation

We are planning to continue implementation moving towards the presentation on December 9th, and then additionally into the second semester. We will continue modifying our application as we move into the testing phase, where we will gain insight into any areas for improvement.

7 Professionalism

This discussion is with respect to the paper titled " Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Our group analyzed our areas of responsibility using the IEEE code of ethics.

Area of responsibility	Definition	NSPE Canon	IEEE differences
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	Perform services only in areas of their competence; Avoid deceptive acts.	The IEEE code of ethics includes a focus on work competence, namely code 6, "to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations." In addition, the IEEE code includes sections about disclosing information promptly and completing professional work.
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	Act for each employer or client as faithful agents or trustees.	IEEE code 5 states, "to be honest, and realistic in stating claims and estimates based on available data." In addition, IEEE code requires the rejection of all bribery, which falls under the umbrella of financial responsibility.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	Issue public statements only in an objective and truthful manner; Avoid deceptive acts.	IEEE code 5 states, "to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest, and realistic" and also to "credit properly the contributions of others." In addition, the IEEE code of ethics also includes a section on avoiding false or malicious actions, rumors, or any other statements contradictory to general engineering ethical standards that could be detrimental to yourself and others.

Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	Hold paramount the safety, health, and welfare of the public.	The IEEE code holds “paramount the safety, health, and welfare of the public” and strives to comply with ethical design. The IEEE includes this as its first code of ethics and places an emphasis on health, safety, and well-being.
Property Ownership	Respect property, ideas, and information of clients and others.	Act for each employer or client as faithful agents or trustees.	The IEEE code seeks to “avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist.” In addition, the code seeks “to avoid unlawful conduct in professional activities.” These IEEE codes, 3 and 4, apply to the concepts of respecting property and ideas of clients and others.
Sustainability	Protect environment and natural resources locally and globally.	Conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession.	The IEEE code 1 “strives to comply with ethical design and sustainable development practices...and to disclose promptly factors that might endanger the environment.”
Social Responsibility	Produce products and services that benefit society and communities.		The IEEE code 2: “improve the understanding by individuals of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems.” IEEE code also places an emphasis on avoiding harm to others or engaging in harassment or other illegal and unethical behaviors of any kind.

The IEEE code of ethics differs from the NSPE version only slightly in each area, and most of the concepts are the same. The IEEE code of ethics also includes an important clause, reading, “to support colleagues and co-workers in following this code of ethics, to strive to ensure the code is upheld, and to not retaliate against individuals reporting a violation”, which relates to every aspect of the code of ethics.

7.2 Project Specific Professional Responsibility Areas

Area of responsibility	Definition	Does it Apply? How are we doing in the context of our project?
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	This certainly applies to our project and we are seeking to meet this responsibility in the context of our project by meeting and checking-in with our client and completing all class assignments on time, as well as moving ahead with frontend and backend app development.

Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	This applies because we must understand the need of educators and students to have access to this type of classroom management software at low costs while not sacrificing value. For this reason and others, we seek to offer this application free of cost to Iowa State students, faculty, teaching assistants, and more.
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	We adhere to this area of responsibility by being understandable and upfront with our client and also being completely honest when completing each of the class assignments.
Health, Safety, Well-Being	Minimize risks to safety, health, and well-being of stakeholders.	This does apply to our app development and we will always minimize risks to safety, health, and well-being of our users and stakeholders as our top priority.
Property Ownership	Respect property, ideas, and information of clients and others.	This strongly applies to our project because we must be careful not to rip off any of the ideas implemented by prior solutions in this space, including TopHat. We are planning to carefully implement new features which do not fully overlap with solutions already implemented. In addition, our app processes student questions and class information, which the app needs to keep confidential and only viewable to students, TAs, and professors.
Sustainability	Protect environment and natural resources locally and globally.	These concepts do apply to our project and we always seek to leave the smallest environmental impact possible during app development. In addition, we seek to produce a low/no-cost product that will benefit students, TAs, and professors with a high level of accessibility and a positive impact on education.
Social Responsibility	Produce products and services that benefit society and communities.	

7.3 Most Applicable Professional Responsibility Area

One area of professional responsibility important to both our project and which our team has demonstrated a moderate or high level of proficiency in the context of our project is Communication Honesty.

Communication and honesty is very important to our project so that we can develop a good relationship with the client and with each other. It is also important so we can adhere to each of the IEEE code of ethics and work on the project responsibly and ethically. In addition, honesty is important to evaluate all of our work and make sure it is not too similar to previously existing solutions in the space, including TopHat.

We have demonstrated this area of responsibility by being upfront and honest with our client, and also being completely honest when completing each of the assignments. We have also had good team communication, and all team members have been involved in completing the assignments and having a role in the team, which has helped.

Establishing good communication has improved our teamwork and also helped us avoid conflict. Hopefully, moving forward we can keep our communication and honesty when communicating with each other and with the client in order to create the best possible application.

8 Closing Material

8.1 Discussion

As we complete our design phase, we can verify that our plans for the application will meet all requirements. We will have more information to complete this section later on.

8.2 Conclusion

So far, our group has worked diligently to complete all lightning talks, course assignments, and application development to a high quality standard. As we transition from a successful design phase to implementation, and then testing the app, we are hopeful that Iowa State students, TAs, and professors will find the app useful, allowing them to enhance the high quality education offered at the university free of charge.

8.3 References

“The Active Learning Platform for Online, in-Person and Blended Courses.” Top Hat, 14 Sept. 2021, https://tophat.com/.

“Angular vs React 2021: Which JS Framework Your Project Requires?” Insights on Latest Technologies - Simform Blog, 18 Nov. 2021, https://www.simform.com/blog/angular-vs-react/.

Author Bio Jeel Patel Designation: Founder - Monocubed Jeel Patel is the Founder of Monocubed and is the main curator & writer of the content found on this site. With ideals , et al. “10 Best Front End Frameworks for Web Development in 2021.” Monocubed, 30 Nov. 2021, https://www.monocubed.com/best-front-end-frameworks/.

Hibbard, James. “The 5 Most Popular Front-End Frameworks Compared.” SitePoint, SitePoint, 13 Apr. 2021, https://www.sitepoint.com/most-popular-frontend-frameworks-compared/.

Hibbard, James. “The 5 Most Popular Front-End Frameworks Compared.” SitePoint, SitePoint, 13 Apr. 2021, https://www.sitepoint.com/most-popular-frontend-frameworks-compared/.

How Can We Help You? - Packback. https://help.packback.co/hc/en-us.

“JScripters.com.” Jquery Disadvantages And Advantages | JScripters.com: Developing a Site with Javascript, https://jscripters.com/jquery-disadvantages-and-advantages/.

Thomas, Gavin. LORECENTRAL, 14 Nov. 2017, https://www.lorecentral.org/2017/11/advantages-disadvantages-jquery.html.

“Vue.js.” Wikipedia, Wikimedia Foundation, 24 Nov. 2021, https://en.wikipedia.org/wiki/Vue.js.

8.4 Appendices

8.4.1 Team Contract

Team Members:

1) Nick Oswald 2) Michael Kies

3) Brian Sayre 4) Vance Kaw

5) Daniel King 6) Robert Walling

7) Jeremy Tracz

Team Procedures:

1. Day, time, and location (face-to-face or virtual) for regular team meetings:

Tuesday, Thursday 5:30 - 6:30 (Ended whenever needed). Virtual.

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

Discord (desktop and mobile app).

3. Decision-making policy (e.g., consensus, majority vote):

Majority vote.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

Meeting minutes will be stored in a document in Google Drive shared with the team. Brian will keep meeting notes and times.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

We expect each member of the team to be a part of team meetings, and to arrive within 20 minutes after the meeting time. This can be waived if the member has an emergency or conflict that was expressed before the meeting time.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

A bare minimum of one push to git per week. Deadlines should be met or worked on at a reasonable rate if problems are occurring. Team assignments should be completed on time.

3. Expected level of communication with other team members:

As stated above, discord will be the main method of communication. We expect everyone to engage in conversation in the team meetings and provide at least some feedback/updates on what they have done

4. Expected level of commitment to team decisions and tasks:

Each team member should be committed to their task and work on what was assigned to them. Roles can be moved around if they communicate a want to change tasks.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

Nick Oswald: Client interaction, Individual component design, Scrum master.

Michael Kies: Team organization, Testing

Brian Sayre: Team organization, development, meeting notes

Vance Kaw: Team organization, Individual component design

Daniel King: Individual component design, assist other members as needed

Robert Walling: Individual component design, team organization, testing

Jeremy Tracz: Client interaction, individual component design

2. Strategies for supporting and guiding the work of all team members:

Communicate any problems occurring so that the team knows where work is needed. Pair programming will be used. Teams can be flexible if problems are occurring.

3. Strategies for recognizing the contributions of all team members:

Maintain a board on GitLab. Weekly standup to discuss what each team member accomplished since the last standup.

Strategies for encouraging and support contributions and ideas from all team members:

Allow time for everyone in meetings to bring up ideas and talking points with the group. If, for some reason, someone is not comfortable bringing it up with the whole group, they may suggest it to the team communicator or scrum master, who may bring it up for them.

Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their opportunity or ability to contribute?)

Communication is key. Every weekly meeting, each member will explain what is going on during their development in order to make sure that there are no conflicts within the code and overall architecture.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

Effective communication and planning for the project. Work contributed to the project each week by every team member.

2. Strategies for planning and assigning individual and team work:

Maintaining a board on GitLab with all issues/tasks. Assign tasks at sprint planning meetings as necessary.

3. Strategies for keeping on task:

Maintain consistent communication with the group and stakeholders.

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

Personal warning then verbal warning at the meeting if it happens again.

2. What will your team do if the infractions continue?

If the team member is still infringing the contract, we will message TA to find a harsher punishment.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1) Daniel King____ DATE 9/16/2021__________

2) Vance Kaw____ DATE 9/16/2021__________

3) Michael Kies_______________________________________ DATE 9/16/2021__________

4) Brian Sayre_______________________________________ DATE 9/16/2021__________

5) Robert Walling____________________________________ DATE 9/16/2021__________

6) Nicholas Oswald__________________________________ DATE 9/16/2021__________

7) Jeremy Tracz______________________________________ DATE 9/16/2021__________