CECS 443 Team Assignment #3

CECS 443 - Software Project Management and Testing


California State University, Long Beach
College of Engineering
Group 4
4/30/2023

## 1. Project Title and Authors

- Team Name:
    - Beach Raters
- Team members:
    - Justin Hoang
    - Kay Kayale
    - Anthony Tran
    - Ashton Mulkey
    - Carine Gordillo
    - Joanna Hsu
    - Jolene Ngu

## 2. Preface

Our **vision** for this app aims to be a game-changer for students, providing a user-friendly and accessible platform for sharing experiences and opinions about different aspects of campus life. With this app, CSULB students can make more informed decisions about their campus experience and provide a platform to experience different perspectives. Our polling/voting app for CSULB will have a diverse **readership**, including students, teachers, parents, and administrators. Students will be able to use the app to vote for student council representatives, provide feedback on school policies, and participate in quizzes. Professors will use the app to gather feedback from students on their teaching methods and to conduct quizzes. The **rationale** for developing a rating/voting website for CSULB is based on the need for a more efficient and inclusive way to gather feedback from students, parents, and teachers. Currently, there is no centralized platform for collecting feedback, which makes it difficult to gather and analyze data. By implementing a rating/voting website, we can improve the quality and quantity of feedback, which will help the school to better understand the needs and expectations of the school community.

## 3. Introduction

In our polling website for California State University, Long Beach students, we have created a way for students to post polling and rating questions about any food recommendations, class questions or more. Think of it like a Yelp but exclusively for CSULB. Other users are able to see the post and answer accordingly if it is a polling or rating question. If it is a polling question, the students are able to vote on the choices. If it is a rating question, then the students can answer it with a 5 star system. Anyone with a CSULB student can use this service and it will keep all users anonymous. This project is a **need** for students at CSULB because a polling/voting project website can be used to gather feedback, make decisions, conduct

market research, support education and research, and engage with a community. The **need** for a polling/voting project website will depend on the specific goals and objectives of the project.

## 4. Architectural Change

**Front-end:** We used HTML and CSS for styling and React JS for functionality. The **rationale** of using HTML and CSS is to style the pages by changing the margins, colors, and other minute details. React JS creates the intractable pages and all of the button functionality.

**Back-end:** We will be using express JS and MongoDb to do all the backend connectivity. The **rationale** is Express JS serves as the middle man between the front end website and backend server. MongoDb will serve as our back end server and this is where all the imputed information will be stored for future references.

**Database:** We will be using MongoDB. The **rationale** for using MongoDb was because it collects user sign in information, posts information, and it is overall a popular backend database. For the collection collecting the sign in information, each user has its own unique ID identifier to differentiate itself from different users and it collects the user's sign in data. For the collection collecting the posts, it collects both Polling and Rating posts. Each post has its own unique ID identifier to differentiate itself from the different post. Each post also has a "postType" that identifies if the post is a rating or polling, it also has a "User" to collect the user who posted the post, and all of the post's information associated with that post.

**APIs:** We used Express Js to create a RESTful API architecture. The **rationale** behind using Express JS is because it is a way to update the database using all of the front end functions. It can Get, Post, Put, and Delete data from the backend. For example, when creating an account, a user types in all of their information and it is "POST" into our database. This sends the sign in information into our database for future references.

**Component-Based Design:** This approach involves breaking down the user interface into reusable components, such as buttons, forms, and navigation bars. The **rationale** for using component-based design in a high-level architectural design is to improve the maintainability, reusability, and consistency of the system. By breaking down the user interface into reusable components, the development team can create a library of building blocks that can be used and reused throughout the system.

## 5. Detailed Design Change

We had to do a lot of design changes. We were unable to implement the Admin View into the system. The admin was able to see the number of reports, delete flagged posts, and manage users. The rationale behind this change was because we were unable to implement the changes in a reasonable amount of change and it was not a functional requirement for our polling and rating app.

Another design change we had to do was remove the ability to like or report a post. This removes features that are related to sorting the post by likes, and the report function. The rationale behind the change was because we were unable to fit these changes into our timeline. Even though we were unable to implement these features, the main portion of our polling and rating app works.

Another design change we did not implement was “Post” was the parent class of “Rating” and “Polling”. Instead of making a parent class and having 2 child classes associated with it, we made 2 unique classes, “Rating” and “Polling” with their own unique data and no parent class. The rationale behind the change was because it was easier and less time consuming to implement this change. At the end, the data is able to display successfully.

## 6. Requirement Change

**User**

**1.5 The user MAY report another user’s Rating / Poll post if it is inappropriate!**

**1.5.1. If a user is logged in, the user MAY report the post if they find it offensive or harmful for the app’s environment**

Because of the circumstances of our situation, we have collectively agreed to take out the reporting feature in posts. The overall design of the website will not be changed since it was previously a feature where the users have the **option** to do so or not. Nothing else will be changed due to the removal of the report feature since it does not affect much throughout the website. This feature will be implemented in the future during revision where we seek to improve the website to help create a safer environment for users.

**System**

**2.3. The system SHALL allow users to view Ratings / Polls while not logged in**

**2.3.1. The system SHALL allow all users to view the results of a Rating / Poll**

We have decided to remove the feature where the system will allow all users to view the results of a Rating / Poll due to the fact that we believe that a user needs to be logged in to view posts. The overall design of the website is not changed because we are still able to implement the main requirements. This requirement will not be implemented in the

future because we believe that there should only be actual users viewing and using this website.

**2.5. The system SHALL allow for the sorting of the list of Ratings / Polls**

**2.5.1. The system SHALL allow for the option to view a sorted list**

Due to time constraints and project priorities, we have decided to remove the feature where users will for the sort of the list of Rating./ Poll. The complexity of this feature will require the development of an algorithm that sorts the posts of which we do not have the resources to allocate time to do so. The overall design of the website is not changed because we are still able to implement the main requirements and will not change the site very much as the posts will just be not in order. This requirement will be implemented in the future during our iterative phase where the project team will revisit and refine the website's visuals.

**2.6. The system SHALL notify an admin if a post exceeds a certain amount of reports!**

**2.6.1. If a post exceeds (n) amount of reports, an admin SHALL be notified about these posts and all of the report flags on it.**

Due to time constraints and project priorities, we have decided to remove the feature where user will set a start and end date while creating a poll. The overall design of the website is not changed because we are still able to implement the main requirements. Assuming that there will not be a report button, it shouldn't change the project at all since this relies on a report button.This requirement will be implemented in the future during our iterative phase where the project team will revisit this feature to improve on the safety and community guidelines of our product.

**2.7. The system SHALL prioritize posts with the most "Likes"**

**2.7.1. The more "Likes" a post receives, it SHALL be displayed first in our homepage**

Due to the priorities of our project, we have decided to remove the feature where the system prioritizes the posts with the most likes. This is due to the fact that we would have to create a very complex algorithm in order to sort and prioritize the most popular posts. The overall design of the website has not changed since we still have everything. It will just look a little bit different since the posts are not organized based on popularity. This requirement will be up for consideration in the future since it will give a way for users to look for the most liked food places on/near campus.

**2.8. The system SHALL create a unique link to share the post"**

**2.8.1. When a post is created, there will be a unique link associated to it so users can share the post with each other**

Due to the time constraint of our project, we have decided to remove the feature where the system creates a unique link to share the post. Because our program only works locally, we did not fully deploy it for online use so the links will not work when the users share them.