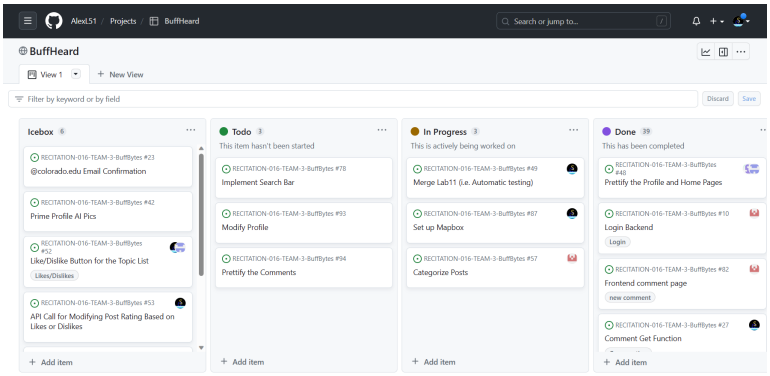
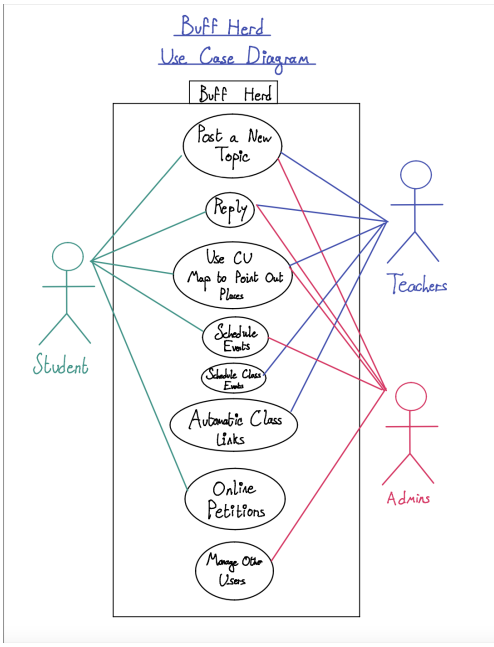
Group 16-03 (Buff Bytes) Final Project Report

- **Title:** BuffHeard
- **Who:** Sean Carter, Max Fogler, Jesus Carnero, Ben Lipman, Alex Ludwigson
- **Project Description:** An accessible web application for CU students. This app would replace the disarrayed constellation of club Discords, Piazza posts, and Canvas forum posts with a united space for CU students to socialize, coordinate meetings, and discuss academics. Verified professors and staff members can answer questions and provide clarifications.
- **Project Tracker** - GitHub project board:
    - [BuffHeard (github.com)](https://github.com/users/AlexL51/projects/1)
    - 

- **Video: (Sean)** 5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.
- **VCS:** [AlexL51/RECITATION-016-TEAM-3-BuffBytes (github.com)](https://github.com/AlexL51/RECITATION-016-TEAM-3-BuffBytes)
    - **Test Cases (Jesse) (Posts and Comments)**
    - **Video demo (Sean)**
    - **README.md in GitHub (Jesse)**
    - **Project documentation (Jesse)**
- **Contributions:**
    - **Sean:** I began by working with Max to set up the database structure files (create.sql and insert.sql) and designed a way for us to store nested comments inside our database. I designed many APIs, including the login and register API and the API to get information from the database to display the user's profile info. Towards the end of the project timeline, I implemented the Mapbox API integration that allows users to add a map to their posts. This is the feature I worked on that I'm the most proud of.

- **Max:** My first work was on designing database structure and the backend APIs which GET and POST comments and their replies. This included creating relational structures so that comments were tied to specific posts and replies (especially among the nested comments) were tied to their parent comment. Then I worked on front end design for the profile and home pages. This included fitting background images, adjusting fonts, and establishing good Bootstrap positioning. Lastly, I designed and completed our group's project presentation. This included coordinating with my team members and implementing their feedback and comments into a short, yet pretty-looking, presentation.
- **Jesus:** My role in this project was working on the front end of the application. I began my work by making the wireframes for all the expected pages of the application. Later I implemented the register and login front end, along with the header and made the general theme of the front end design. Then I worked on lab 11 and designed some test cases for when our application would be finished. Finally I've been working on making general edits to the front end of the pages we have so they look nicer and more complete along with user testing.
- **Alex:** Initially I worked on some of the first APIs and then spent most of my time on the page where you see a post with all of its comments. I had to write several APIs for this feature and spent a lot of time getting the logic in the ejs files to appropriately nest the comments and indicate who is responding to who as well as appropriately inserting a into the database while maintaining this chain. After this I mainly did minor frontend changes and testing.
- **Ben:** The first thing I worked on was setting up the frontend layout for the home and profile page. I then wrote the API to GET all the information for the home page. I didn't write the original API for the profile page, but I rewrote it to get all of the topics a user has posted, and the comments on those topics, so that they could be displayed on the profile page. I also modified the API for the comment page. From there, I worked on making the profile page look nicer, as well as the home and comment pages.

Use Case Diagram:



- **Test results:** We ran our testing with 4 different CU students who we asked to register, login, make posts on the web page, add some comments and check their profile information.

For the first test, the users registered an account with whatever username they wanted and used any password they wanted as well. Once that was done users then went on to login. After registering we had the users attempt to login with a similar password to the one they had (Capitalizing a letter or changing a character) or with an empty password to make sure there were no issues.

Once logged in, we prompted the tester to make posts and comments. On this part of the test we encountered our only errors. The users were able to make normal posts with no issues whatsoever. However, one of the users attempted and was successful in adding a completely empty post. The fault in our code was a simple fix, all we did was check for empty strings rather than null to correct the issue. As for our

second error, it was the exact same situation but it happened when the users tried making empty comments instead of posts.

The users were then allowed to use the app freely by posting and commenting on whatever they wanted. Once they made a few posts we asked the users to open their profile info to make sure all their posts were there. We then encountered 0 errors. None of the users seemed to have any trouble navigating through the application which was expected as the layout was very simple.

- **Deployment**: **(Max)** Link to deployment environment or a written description of how the app was deployed and how one might access/run the app. The app must be live, working, and accessible to your TA.