



# BUFF TRAFFIC

## Project Report

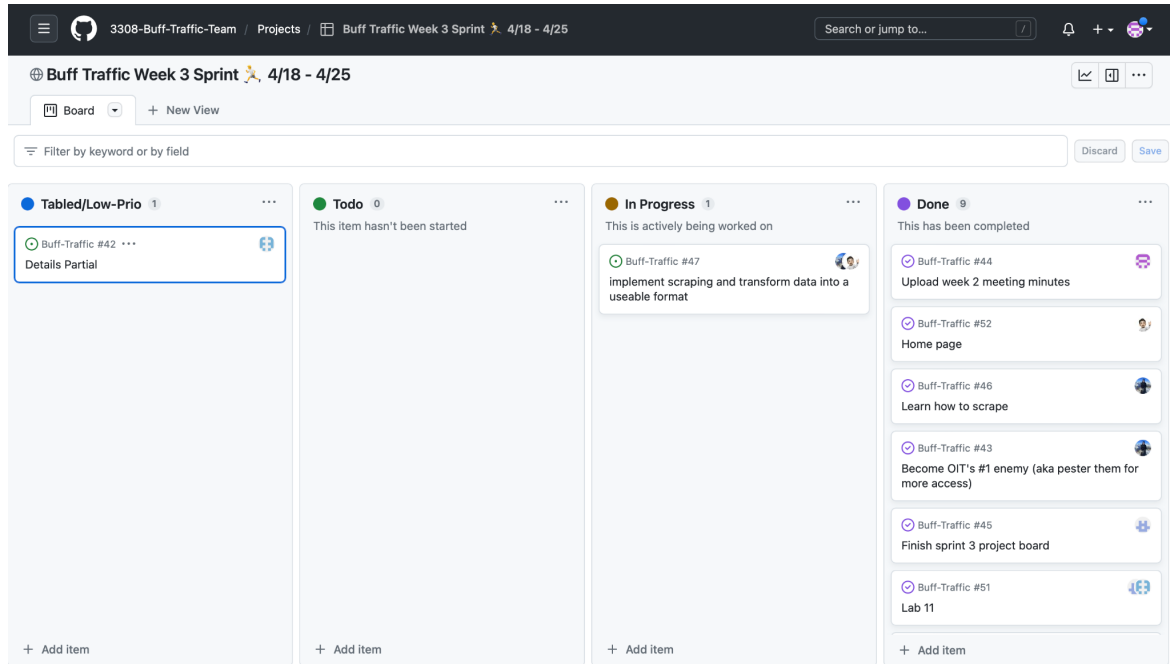
By Yohanes Limasalle , Lilly Nguyen , Noam Tuchman , Che Meng Her , and  
Ben Garduno

### Project Description

Have you ever walked into The Rec and realized it's more packed than a can of sardines? If only there was a way to predict and plan ahead to prevent the dreaded Rec Center rush hours. Introducing **Buff Traffic**, an app to track current and historic traffic trends at The Rec so you can spend more time working out and less time awkwardly waiting on the side for your turn. For CU Boulder students, who regularly go to the Rec Center, Buff Traffic is a website that provides real-time data on how busy the Rec is. Unlike the current Rec Center app and simply googling the location, our product provides live and predicted facility counts in a more visually appealing way. This application will pull live data from CU's Rec Center database so that users can gauge how busy or not busy facilities are. Users can create profiles. With these profiles, users can favorite rooms that are available within the Rec Center. When logged in, the user is presented with a live feed of how busy those favorite rooms are at a quick glance.

# Project Tracker

<https://github.com/3308-Buff-Traffic-Team/Buf-Traffic/projects?query=is%3Aopen>



## Video Demo

<https://drive.google.com/file/d/1yxYQLUyG9cOFs4s8xwMhglBhECOrMbfl/view?usp=sharing>

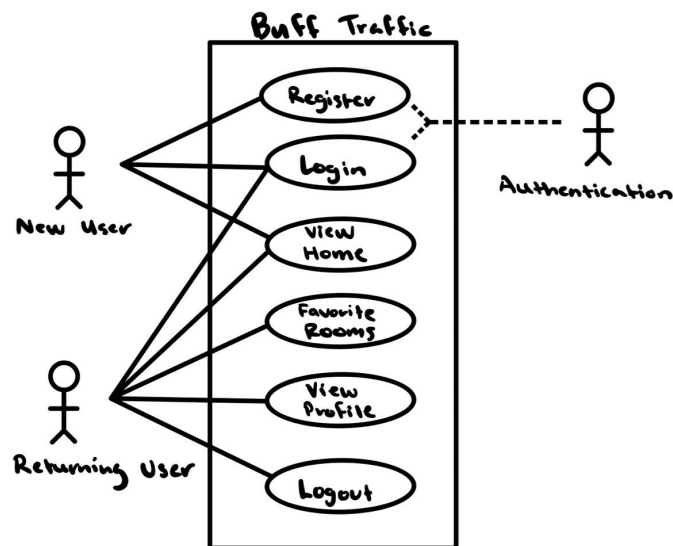
## VCS

<https://github.com/3308-Buff-Traffic-Team/Buf-Traffic>

## Contributions

- **Yohanes** - Contributed via backend structure and project management/organization. Backend design can be seen [here](#), helped link the resulting data into home.ejs, created the readme, maintained project board and release notes.
- **Ben** - My role was to work on both the frontend and backend of the website aiding others where I could. This included using javascript, HTML, css, sql, and ejs. I made improvements to existing code to add more features such as error messages. I also did initial testing for the website, created the navbar and partials, made/helped code api's, and worked on the favorites section on the homepage.
- **Noam**- My role was almost fully on the backend. I worked on pursuing the rec's IT department to get their facility count data. Then I processed that data and put it into the database. I helped organize the data being displayed to the user in SQL queries. I would always help debug in the back end whenever anyone needed help. I largely contributed to the login system. I used javascript, python, and lots of SQL, and made the dockerfile. Honestly, my contributions wouldn't have mattered if I didn't have the rest of the team working to make the site a reality.
- **Lilly**- My contributions consisted of working on the front-end and some back-end. I mostly created the design and look of our website through the initial design of wireframes and logo. I worked on the register, login, and profile pages to make them look nice and uniform. This included using HTML/CSS and EJS. I edited the navbar to display appropriate options for the current user's session and helped largely with the presentation and project report.
- **Che**- My contributions and efforts were spread across the front end and the back end. Initially, my efforts only saw the creation of some pages with some HTML and CSS. However, I'm glad that I was also able to branch out and work on the database system as well as making sure that the data being pulled from our database onto our home page was up and running and reflecting the data that we wanted to be there. And of course, helped others whenever I was able to in areas such as git actions and development as a whole.

## Use Case Diagram



## Test Results

### ~Test 1~

- Title: Login
- Description: User should be able to, if they have an account, log into our system
- Data: Users table, Users-to-favorites join table
- Environment: Azure csci3308teamblync.eastus.cloudapp.azure.com:3000
- Testers: friends on campus (similar age, tech-savvy, aware of the rec center)
- Steps:
  - Go to the login page
    - Expected result: User identifies the logo on the top of the home page, indicating a login feature
    - Actual Result: Easily spots the login button in the corner of the screen and clicks on it
    - Notes: Login feature successfully identified and interacted with
  - Enter credentials

- Expected Result: User identifies the login fields and enters the correct information
- Actual Result: User is confused by the “username” field and is unsure if this is supposed to be their email address; password is successfully inputted
- Notes: We changed the placeholder of the field to say “email” to avoid confusion when logging in
- Unsuccessful login
  - Expected Result: If the user misenters their login credentials, will be prompted that the information provided does not match with our database
  - Actual Result: User types in an invalid username and password and the red message pops up saying “Incorrect username or password” as expected
  - Notes: Successful error checking
- Successfully login
  - Expected Result: If the user credentials were correct, will take them to the home page, where their preferences are displayed
  - Actual Result: User inputs the same email address and password as when they registered and is successful redirected to the home page
  - Notes: Successfully logged in with correct credentials

## ~Test 2~

- Title: Registration
- Description: User should be able to create an account in our system
- Data: Users table
- Environment: Azure csci3308teamblync.eastus.cloudapp.azure.com:3000
- Testers: friends (see above)
- Steps:
  - Go to login page

- Expected Result: From the home page, user navigates to the login page
- Actual Result: User easily identifies and clicks on the login button on the navbar
- Notes: User notes that the interactive button via javascript is a nice touch
- Go to registration page
  - Expected Result: From the login page, user navigates to the registration page
  - Actual Result: After glancing at the login page, user recognizes the link to navigate to the registration page where it says “Don’t have an account Sign Up”
  - Notes: Changing the link color to blue makes it more apparent that “Sign Up” is a clickable link
- Register for an account
  - Expected Result: The user enters information according to the fields/prompts and attempts to create an account
  - Actual Result: User types in the correct data for each field labeled “First Name, Last Name, Email, and Password” and clicks Register button where they are redirected to the login page
  - Notes: Successful attempt and the placeholders were brief yet descriptive enough for the user to understand
- Unsuccessful registration (password is too short)
  - Expected Result: If the user enters a password that is less than 6 characters, it will throw an error message explaining the situation
  - Actual Result: User types in a password that is 3 characters long and a red alert message pops up on the page indicating that the “Password must be 6 characters or longer”
  - Notes: Successful error checking
- Successful registration

- Expected Result: If the user creates an account successfully, will direct user to the login page
- Actual Result: User types in the correct data for each field labeled “First Name, Last Name, Email, and Password” and clicks Register button where they are redirected to the login page
- Notes: Successful attempt

### ~Test 3~

- Title: Home Display
- Description: The main dish, user should be able to see current and projected headcount at the rec, assorted via rooms
- Data: Traffic Table, Users-to-favorites table
- Environment: Azure csci3308teamblyncc.eastus.cloudapp.azure.com:3000
- Testers: friends (see above)
- Steps:
  - Page navigation
    - Expected Result: User can navigate to our app via URL
    - Actual Result: Using ‘/home’, user was able to navigate to our home page
    - Notes: Add a default ‘/’ route to redirect to home page
  - Information displayed
    - Expected Result: User can view current headcounts of the rec, assorted by room
    - Actual Result: Cards of various rooms are displayed but in no particular order and some rooms have no images displayed; there is a blue percentage bar at the bottom of each card
    - Notes: Percentage bar is good at displaying the data visually. Need to add more images to rooms for uniformity
  - Intuitive UI
    - Expected Result: User is able to interact with our UI (scroll for more rooms, login, click on specific rooms for more details, etc) as they would any familiar web app

- Actual Result: User successfully scrolls through the rooms but can only star rooms at the moment
- Notes: Future implementations should include a details partial to show more information about the room such as a description and possible sparkline of how busy the room has been

#### ~Test 4~

- Title: Favoriting Rooms
- Description: User should be able to favorite rooms on the home page and have it displayed on the top
- Data: Users-to-favorites table
- Environment: Azure csci3308teamblync.eastus.cloudapp.azure.com:3000
- Testers: friends (see above)
- Steps:
  - Be logged in to view favorites section
    - Expected Result: User is logged in and can then view the favorites feature through the gold stars on the stars
    - Actual Result: After successfully logged in for the first time, user sees a “Favorites” section at the top of the home page and also cards with gray stars in the corner of them.
    - Notes: Slightly UI changes to be made: even if user is not logged in, the favorites sections still displays and the stars are still able to be clicked although the feature won’t actually work properly
  - Favorite a room by clicking on the star in the corner of the card
    - Expected Result: User can select multiple rooms by clicking on the stars on each of the cards. The stars should turn yellow after selection.
    - Actual Result: User scrolls through the page of rooms and stars the “Ice Rink” and “Buff Pool”. The stars turn yellow when clicked and remain that way.
    - Notes: Successful selection of rooms
  - Refresh the page



- Expected Result: View favorited rooms on top of the home page
- Actual Result: User refreshes the page and the selected rooms from the previous step are displayed in the “Favorites” section of the home page
- Notes: Successful favorites feature; minor note to add a note like “No rooms favorited yet” or “Click there to favorite a room” to indicate the presence of this feature to a first-time user

## Deployment

<http://csci3308teamblync.eastus.cloudapp.azure.com:3000>