Buff Games

Team Members

- Member 1: Evan Feng
- Member 2: Logan Oram
- Member 3: Caden Davis
- Member 4: WIlson Narog
- Member 5: Dylan Sandusky
- Member 6: Ryan Natvig

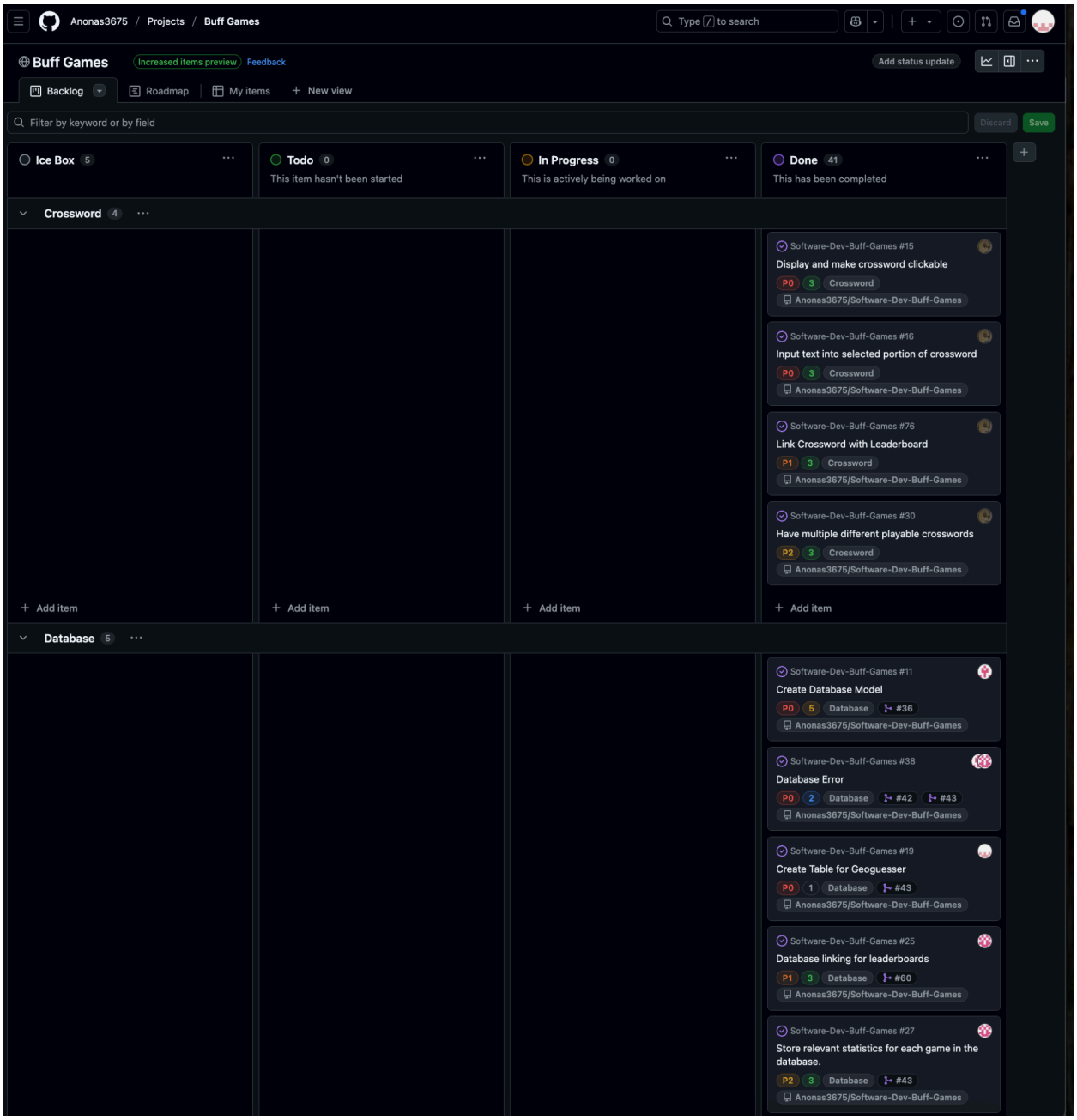
Project Description

Our project is a CU-themed mini game hub inspired by popular games like Wordle, Connections, and GeoGuessr. It provides a fun, local experience for the CU community by offering mini-games that users can play and share their results from. Our games include a GeoGuessr-style challenge featuring CU-specific HDRI images, a Wordle with CU-centric words, a crossword puzzle filled with CU-related terms, and a trivia game focused on CU knowledge.

We used a range of technologies to bring this to life: PostgreSQL for database management, Handlebars for building a simple and dynamic UI, and external APIs like World and Panorama APIs for location data. We developed our backend with Node.js, tested it with Mocha and Chai, used GitHub for version control, and hosted the site through Render. Our goal is to create an engaging, community-driven platform that feels both familiar and uniquely CU.

Project Tracker – GitHub Project Board

- https://github.com/users/Anonas3675/projects/1
- Screenshot of Project Tracker:



Demo Video

- Link to 5-minute Video Demo: https://o365coloradoedu-my.sharepoint.com/:v:/g/personal/dysa4879_colorado_edu/EQCTsmN0MMpOnrYyuuG_t8YBpI1IIMmWw56JJFS8VDQIOA
  Audience: Potential user or customer
- Github Video link: https://github.com/Anonas3675/Software-Dev-Buff-Games/blob/main/MilestoneSubmissions/Buff%20Games%20Project%20Demo.mp4

Version Control System (VCS) – GitHub Repository

- https://github.com/Anonas3675/Software-Dev-Buff-Games

Contributions (Max 100 words per member)

Member 1 – Evan Feng
For our Geoguessr-style project, I implemented the core gameplay functionality that allows users to select a location on the map and receive a score based on their accuracy. I also captured and created the interactive panorama scenes around campus to provide an immersive experience. On the design side, I developed the user interface for the homepage, login, and registration pages, enhancing the overall aesthetic with custom fonts and subtle animations to fit the gaming theme. Small details like a pixel-style font and a buffalo GIF contributed to the site's charm and user engagement, making the experience more visually appealing and fun.

Member 2 – Caden Davis
I created and implemented the scoreboard page alongside creating the back end connections with the database. I worked alongside Wilson to implement and modify the database to meet the changing needs of our group. Beyond these two things I fixed issues with our login and registration page, made it so that the username was displayed on every page for our account on the navbar, solved trivia. I spent the rest of my time either doing labs or reviewing pull requests and solving merge conflicts.

Member 3 – Logan Oram
I developed and implemented the crossword page, utilizing Handlebars and CSS to design its structure and layout, and JavaScript to add interactivity and responsiveness. I also handled the API integration to connect the page with the server. Additionally, I redesigned the database schema to improve crossword storage. Beyond the crossword page, I contributed to the design of the navigation bar and logout page, and made various fixes to the design of Wordle, Trivia, the scoreboard, and the home page.

Member 4 – Dylan Sandusky
I designed and implemented the wordle page. I created the handlebars and CSS to render the page, the javascript to make it interactive, and the API calls that allowed me to communicate to the server. Functionalities include entering guesses, saving guesses, loading guesses, checking if a word is valid, checking if the word is correct, and giving visual feedback. I also fixed a bug with register when the same username was registered multiple times. Finally, I fixed the render Database when it broke.

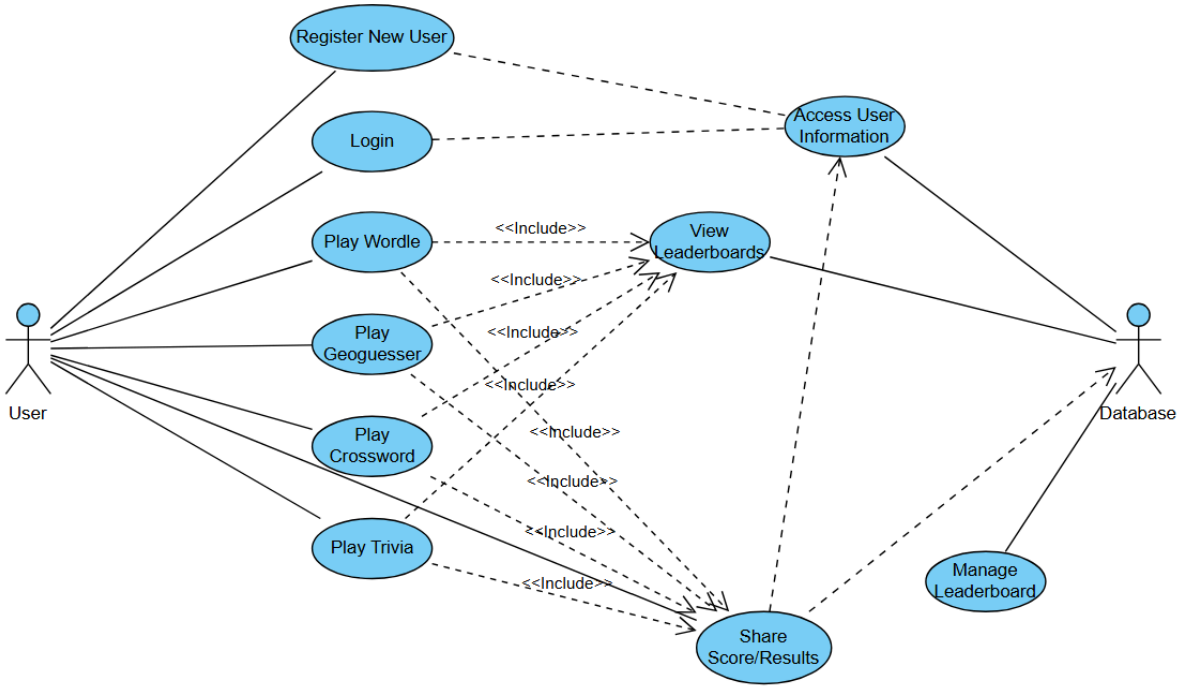
Member 5 – Wilson Narog
I created the initial implementation for the database designed around a general idea for what we would need for the project. I worked with Caden Davis to fix an issue with the database connection where we weren't using the correct credentials, which caused docker to try and run an incorrect database. I also worked on the implementation of the trivia page, designing APIs to select a question and display the question. I then created the UI for the trivia page, and worked with Caden on creating the scoreboard page UI.

Member 6 – Ryan Natvig
I worked to create the base login and home pages to allow the rest of the team to implement their games, and have basic website functionality. Furthermore I found and implemented the PhotoSphere library to allow the geoguessr game to have an immersive experience accurate to CU locations. I also worked on connecting the geoguessr game to the leaderboard. I worked closely with Evan to fix all of the bugs in geoguessr and make the game whole. In addition I worked on the styling for the trivia game, and with all around bug fixes.

- Screenshot of GitHub Contributions:

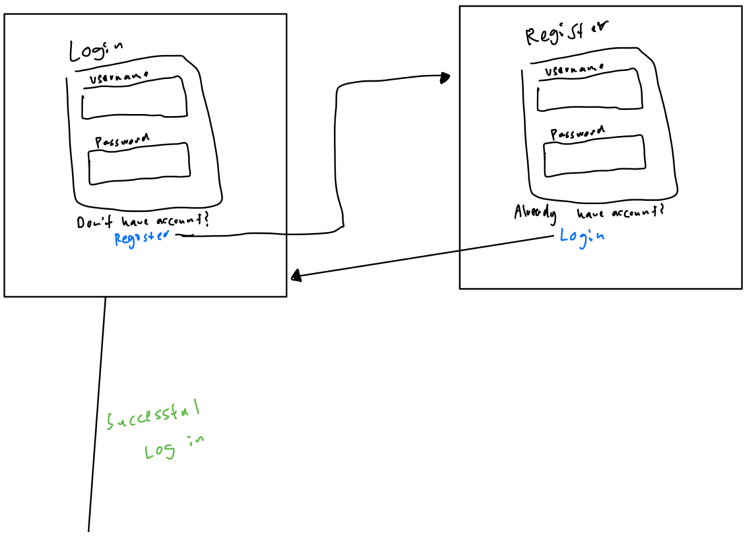
Use Case Diagram

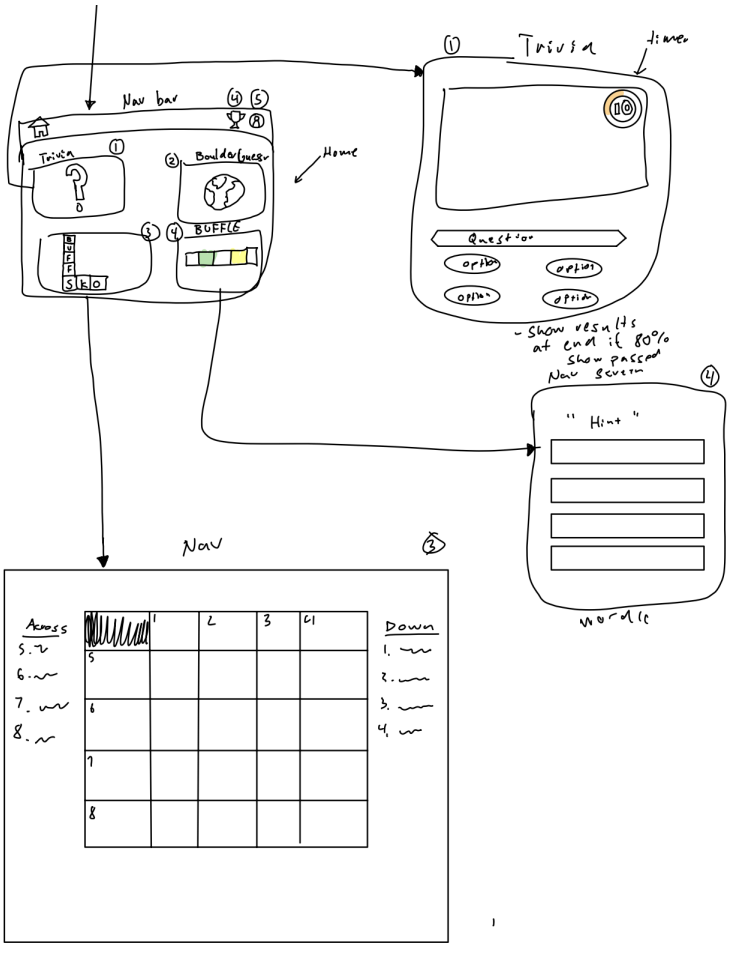
Include your finalized use case diagram here. If the one from your proposal is complete, you may reuse it.

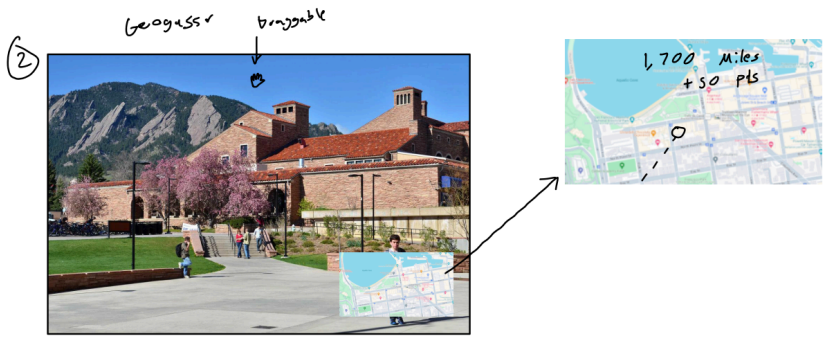


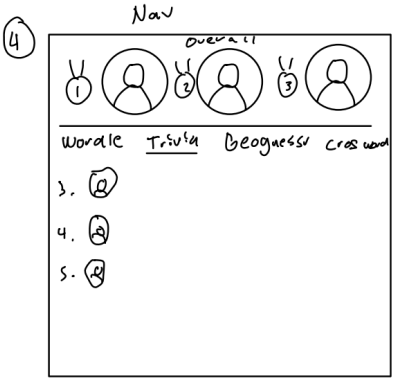
Wireframes

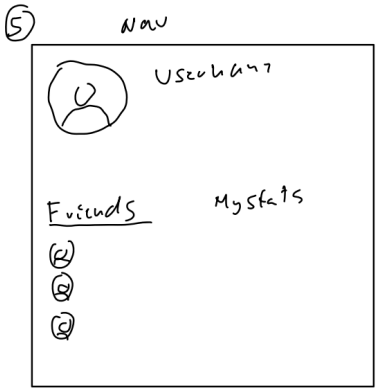
Include wireframes for each page in your application. Photos of hand-drawn wireframes are acceptable.




Nav bar
Trivia
Boulderguess
Home
BUFFLG
Trivia
timer
Question
option
option
option
option
- Show results at end if 80% show passed Nav screen
" Hint "
wordle
Nav
Across
Down


Geoguessr
Boulderable
1,700 Miles
+50 pts


Nav
Overall
Wordle
Trivia
Geoguessr
Crossword


Nav
Username
Friends
Mystats

Test Results

Tested 3 users:

- User 1: A middle aged parent of one of our group members
- User 2: A CU undergraduate economics major
- User 3: A CU undergraduate CS major

Results:

- Test Case 1: Wordle page
    - The 1st user was initially confused on how to enter guesses, since no onscreen keyboard was given. After about 30 seconds of confusion they figured it out.
    - The 2nd user commented on changing how feedback was given, since letters would be highlighted if they were a part of the full word with no changes based on frequency. After being discovered during user testing, this issue was later fixed
    - The 3rd user initially tried to open the wordle page on their tablet, at which point I (the observer) realized the wordle game was currently impossible to play without an attached keyboard, at which point I let them play on my computer instead. No issues after that. Creating a onscreen keyboard added to project ice box
- Test Case 2: Crossword page
    - The 1st user had no issues solving the crossword
    - The 2nd user commented on the difficulty of the second crossword, but didn't have any confusion on how to play the crossword itself.
    - The 3rd user had no issues
- Test case 3: Trivia
    - Due to the limited number of questions, and how the trivia page randomly selects a question, 1st user had some confusion when the same trivia question appeared multiple times in a row.
    - The 2nd user had no issues playing the game, but commented on the ambiguity and some of the trivia solutions. (CU biggest rival being CSU, largest collection of dorms actually being main campus, etc)
    - The 3rd user experienced some confusion switching question difficult because they expected a new question to automatically load after selecting a new difficulty. Creating a solution to address this added to project ice box.
- Test case 4: Login
    - The 1st user experienced slight confusion of how to log into the site, until the saw the link to the register page, at which point they had no issues registering an account and logging in
    - The 2nd user had no issues registering an account and logging in
    - The 3rd user attempted to create multiple accounts under the same username, which caused a critical error that made those accounts inaccessible (you couldn't log in to them). Issue added to project board and fixed later

---

Deployment

- https://software-dev-buff-games.onrender.com/

The Render is going to go down May 7, 2025, so to run the software locally, run docker compose up from the ProjectSourceCode folder on localhost:3000

Here is an example of what you're .env file is supposed to sound like:

```
POSTGRES_USER="postgres"
POSTGRES_PASSWORD="pwd"
POSTGRES_DB="db"
SESSION_SECRET="Sjgieryu54eyoieiovges"
POSTGRES_HOST="db"
POSTGRES_PORT="5432"
```

Here is an example of what you're docker-compose.yaml is supposed to look like:

```
services:
 db:
   image: 'postgres:latest'
   env_file: .env
   expose:
     - '5432'
   volumes:
     - ProjectSourceCode:/var/lib/postgresql/data
     - ./src/init_data:/docker-entrypoint-initdb.d
 web:
   image: node:lts
   working_dir: /ProjectSourceCode
   env_file: .env
   environment:
     - NODE_ENV=development
   depends_on:
     - db
   ports:
     - '3000:3000'
   volumes:
     - ./:/ProjectSourceCode
   command: 'npm run testandrun'
volumes:
 ProjectSourceCode:
```