BUFF GAMES

1. Contributors
2. Summary
3. Tracker
4. Video
5. VCS
6. Contributions
7. Use Case Diagram
8. Wireframes
9. Test Results
10. Deployment

1. Contributors

Member 1: Tayden Brenner - TaydenB

Member 2: Jaden Strommer - JadenStrom

Member 3: Brandon Gonzales - bmgonz27

Member 4: Joe Olapurath - ojoev23

Member 5: Noah Thompson - NoahT0

Member 6: Oliver Schmitz - OliverDSchmitz

2. Summary

For our project we created a new version of wordle. Our wordle is a score based wordle in which you can challenge your friends. The score is based on how long it takes to complete the wordle, as well as how many guesses it took to solve it. Similar to classic wordle you are limited to only using 6 guesses as well as real dictionary words. For casual play of the game, we have a daily game that is played with one word that changes each day. This implementation is exactly like the current wordle except that you can get scores that can place you on a leaderboard. The new addition that we truly added is challenge mode, in which you can add your friend and send them a challenge link. When you send this link you also send them a valid word that they must try to solve playing wordle. They can then send you a challenge wordle back, which keeps the competitive system going back and forth. Your challenge scores can be displayed to you on the

leaderboard tab. Also including an account with a friends list, stats, login, and logout functionality.

3. Tracker

Project Board: https://github.com/users/TaydenB/projects/1

4. Video

Video: https://drive.google.com/file/d/1xfMIGDpK-GHDWh9sfvUQHUBxBCKgrory/view?usp=sharing

5. VCS

Github: https://github.com/TaydenB/group-project-3308

6. Contributions

Tayden Brenner: My first contribution to the project was to create and work on the wireframe through Figma. After that I helped to work on some of the database tables in postgreSQL format, and some handlebar(html) work with the login page. I did the html/css for the game board and keyboard. I created functions that can add, delete, and submit the words/letters into the wordle game board using Javascript. I also added render functionality for the game page using Node. Finally, I hosted and built the cloud for our website using Render.

Jaden Strommer: First, I worked on the profile system, organizing routes and creating pages for account info, stats, and editing user details. I then helped clean up our Handlebars templates, and added some custom CSS to make the UI more consistent. After that, I implemented database updates for daily challenge stats, fixed issues

with leftover guesses, and improved how plays and averages were tracked. I also debugged several server-side errors, helped to reorganize our routes, and containerize and test the app using Docker Compose. Throughout the project, I used Node.js, Express, PostgreSQL, Handlebars, and GitHub to develop, and test features.

Brandon Gonzales: The first thing I contributed to the project was doing the use case diagram. After that I started on the styling for the website. I added a welcome page using HTML and set the route to be our default page using JS. I styled the welcome page and added buttons to navigate to the content of our application. I then continued on the CSS path by making the whole theme for our website into dark mode. I then also styled the header and footer using CSS. I did a lot of front end work on the project.

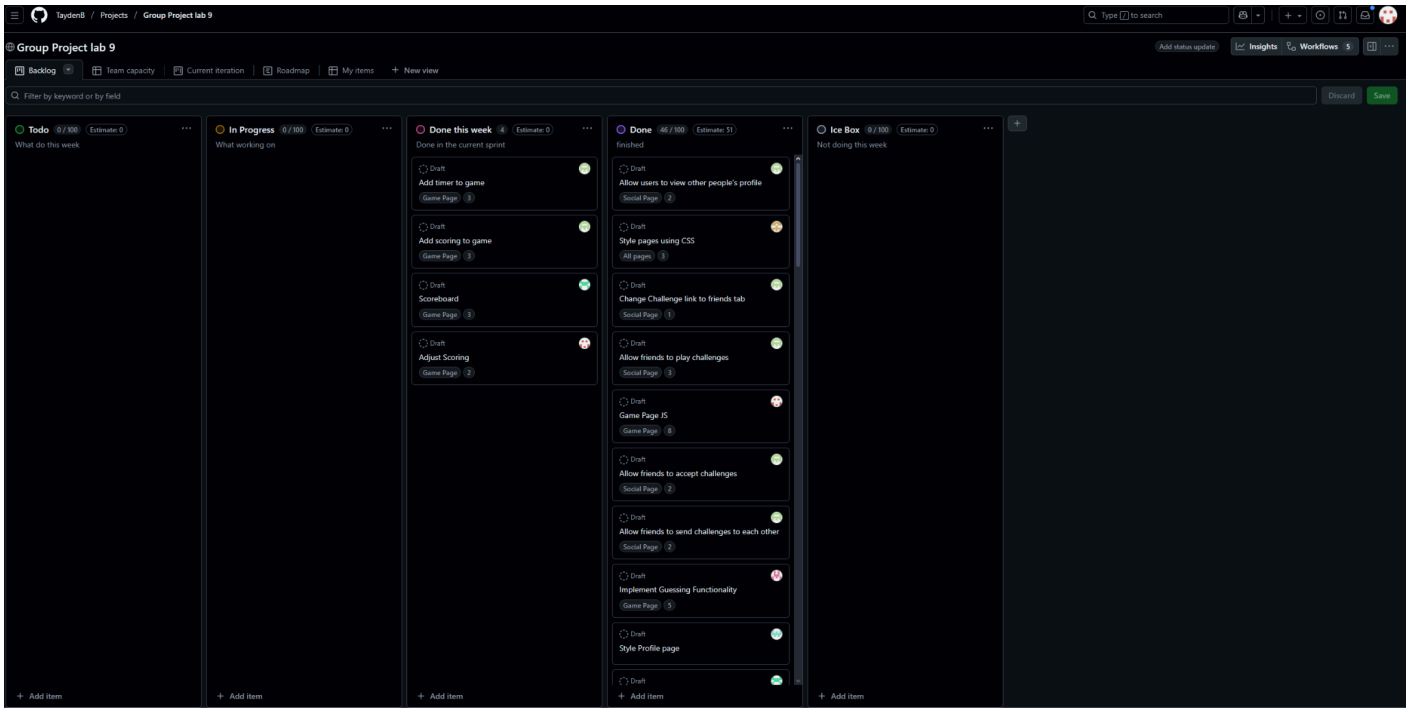
Joe Olapurath: I first started with cleaning up index.js and creating routes. I also created functionality for creating challenges and creating SQL tables for that. I also cleaned up our SQL tables and reduced redundancy. I also added a lot of the core functionality for playing the games with guessing words and seeing what parts of the word are correct or are in the word. I also added an implementation for seeing if it's the right word and if they had too many guesses. I also helped with deploying it on Render.

Noah Thompson: I started with the register and login pages, specifically the backend functionality where I used Node. From there I primarily worked on social features. These include sending/cancelling friend requests, accepting/declining friend requests and sending/accepting challenges. I also wrote test cases to test the friending feature using Chai and Mocha. Next I helped with the game functionality. I abstracted it to work for both the daily and challenge

games, I added typing, a timer and updated the database with scores, times and guesses once they complete the game.

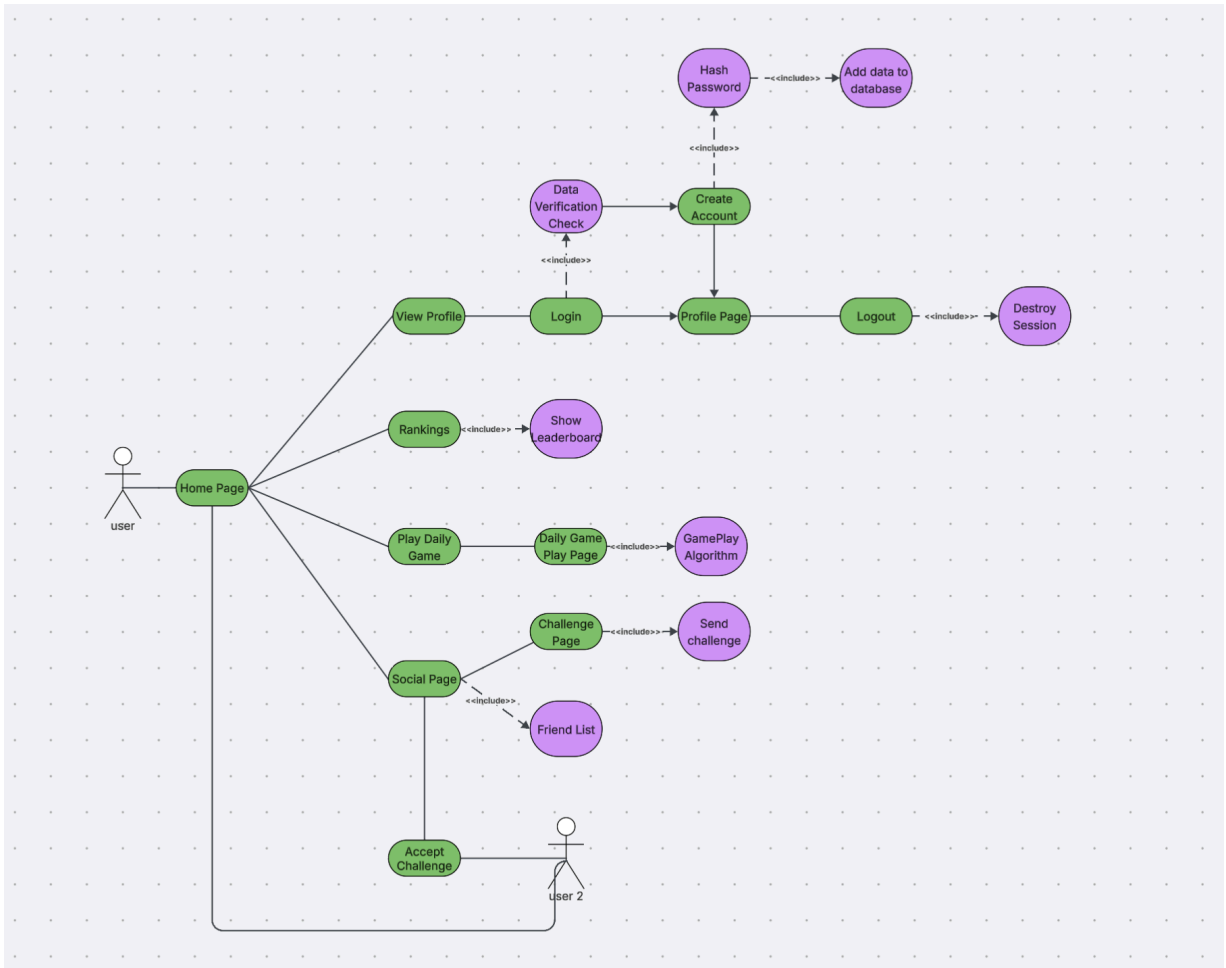
Oliver Schmitz: I first created most of the SQL tables we would use for future data storage. I then worked on the functionality of the daily game by importing a list of 5-letter words and then writing a hash function based on the local date to “pick” a word. I then moved on to working on the scoring functionality for the daily game, based on a clock and the number of guesses, with the user given 5 minutes to finish. Then I worked on the scoreboard functionality for the daily game, linking it to the database and creating HTML for the partial.

Project Board:

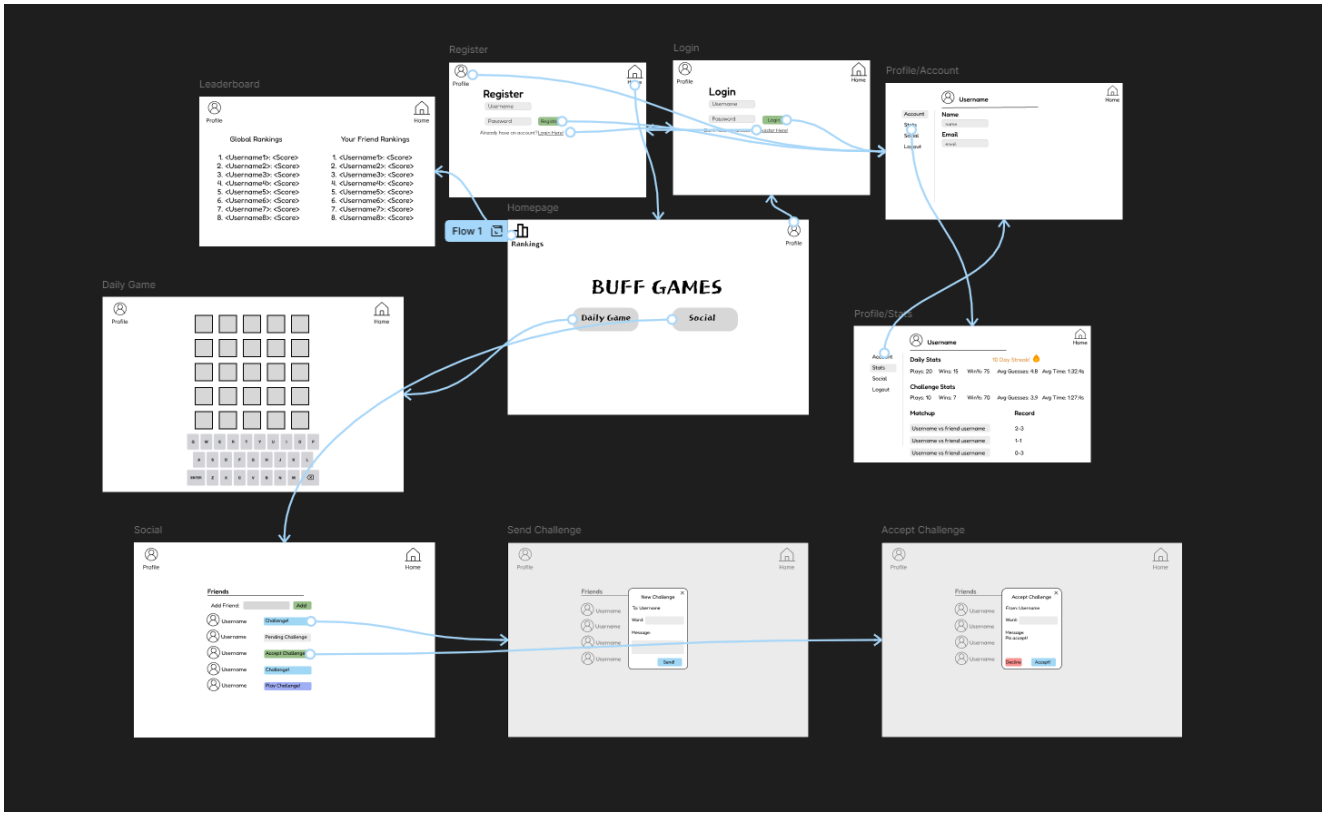


Recent Github Commits:

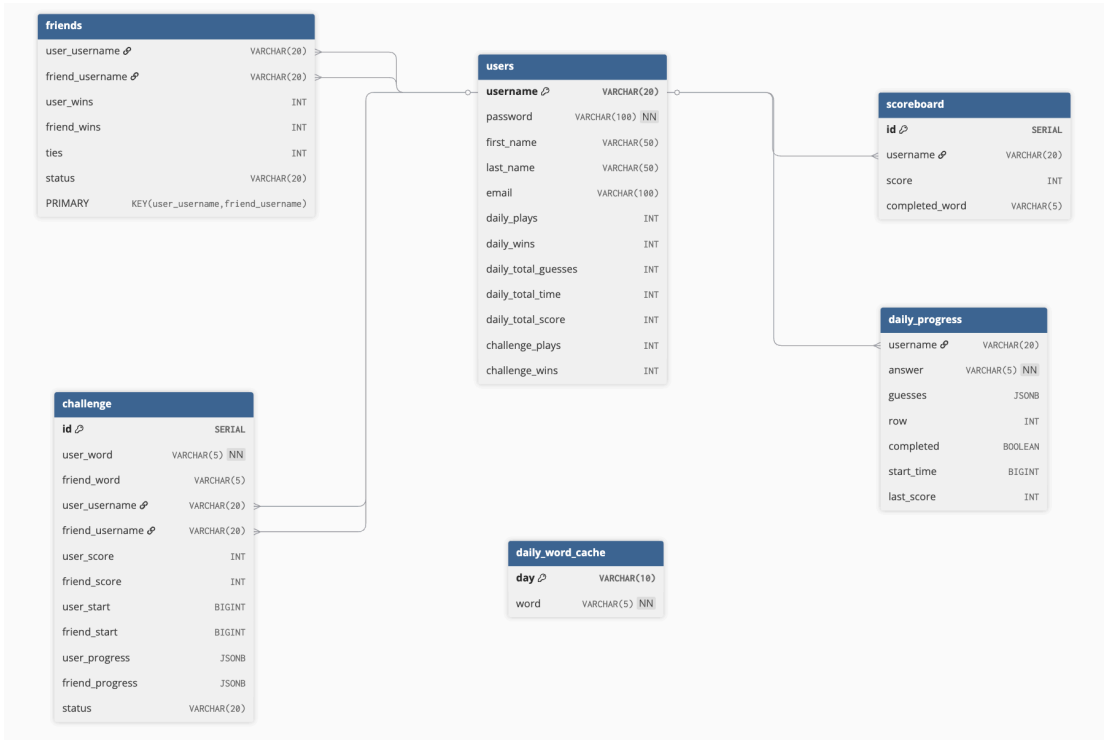
7. Use Case Diagram



8. Wireframe


Register
Login
Profile/Account
Leaderboard
Register
Login
Global Rankings
Your Friend Rankings
Homepage
Flow 1
Rankings
BUFF GAMES
Daily Game
Social
Daily Game
Profile/Stats
Social
Send Challenge
Accept Challenge

Database


friends
user_username VARCHAR(20)
friend_username VARCHAR(20)
user_wins INT
friend_wins INT
ties INT
status VARCHAR(20)
PRIMARY KEY(user_username,friend_username)
users
username VARCHAR(20)
password VARCHAR(100) NN
first_name VARCHAR(50)
last_name VARCHAR(50)
email VARCHAR(100)
daily_plays INT
daily_wins INT
daily_total_guesses INT
daily_total_time INT
daily_total_score INT
challenge_plays INT
challenge_wins INT
scoreboard
id SERIAL
username VARCHAR(20)
score INT
completed_word VARCHAR(5)
daily_progress
username VARCHAR(20)
answer VARCHAR(5) NN
guesses JSONB
row INT
completed BOOLEAN
start_time BIGINT
last_score INT
challenge
id SERIAL
user_word VARCHAR(5) NN
friend_word VARCHAR(5)
user_username VARCHAR(20)
friend_username VARCHAR(20)
user_score INT
friend_score INT
user_start BIGINT
friend_start BIGINT
user_progress JSONB
friend_progress JSONB
status VARCHAR(20)
daily_word_cache
day VARCHAR(10)
word VARCHAR(5) NN

9. Test Results

Use Case 1: Playing and Completing a Challenge
Description: Test whether two users can complete a challenge and have the correct winner/loser recorded

Use Case 2: Viewing Stats After a Challenge
Description: Test whether the Stats page accurately displays updated wins, plays, and averages after a challenge is completed

Use Case 3: Viewing the Social Page Record
Description: Test whether the Social page correctly shows the head-to-head record between two users

Use Case 4: Comparing Stats Page vs. Social Page Data
Description: Test whether both pages stay consistent with one another after a challenge and whether any mismatched values appear

Observations - I tested the challenge system with my roommate acting as an external user. While he interacted with the app, I recorded the following observations:

1. User Actions
    a. He completed a full challenge against me several times
    b. He navigated between the Challenge page, the Stats page, and the Social page to check his results
    c. He expected both pages to reflect the same outcomes
2. User Reasoning
    a. He assumed that if he lost a challenge, his Stats page would show a loss
    b. He checked the Social page because he expected it to mirror the Stats page

    c. When he saw mismatches, he tried repeating the challenge to confirm whether or not it was just a one-time error
3. Consistency With Expected Behavior
    a. His behavior matched the intended use caseL complete a challenge -> check results on both pages
    b. The system behavior *did not* match the expected result: the Stats page repeatedly miscounted wins
4. Deviations From Expected Results
    a. The Stats page awarded wins to the wrong user, often giving a win to whoever guessed last, or simply guessed the word at all
    b. Play counts updated correctly, but wins did not
    c. The Social page, however, always showed the correct record (e.g., 1-0-0 vs 0-1-0).
5. Resulting Changes
    a. I reported these inconsistencies to the team
    b. The issue was confirmed and diagnosed as buggy challenge-result logic on the Stats page
    c. A teammate fixed the logic so wins and losses now update accurately and consistently across both pages
    d. This directly improved the reliability of the challenge feature for any future users

10. Deployment

Game: https://buff-games-group-project.onrender.com

Description: We used render as a cloud service to make our game accessible through the use of any internet browser. In order to play the game all you have to do is go to that website, create an account, and play the game.