Final Project Documentation

SFSU **CSC 667-02 / 887-02**
Due: May 17, 2021

TEAM C

SFSU ID	Name	Email
918599427	Jose Gonzales	jgonzalez34@mail.sfsu.edu
920051012	Frederick White	fwhite2@mail.sfsu.edu
920600366	Suman Basalua	sbasaula@mail.sfsu.edu
920184756	John Freirez	jfreirez@mail.sfsu.edu

GitHub:

https://github.com/sfsu-csc-667-spring-2021-roberts/term-project-team-plus-four

Link to application:

https://uno-game-csc667.herokuapp.com/ (app is not deployed)

Table of Contents

Project Overview

For our final project we were tasked to implement an online multiplayer game. Our team has picked UNO to make. Using frontend frameworks like Node.js, Express, pug view engine (used to be jade) to render HTML/CSS, JavaScript, Postgresql for database, and pg-promise and node package sequelize for creation/migration of table to our database.

Summarized Technical Work

Our team used Figma to design the wireframe/prototype of the application. Used Node.js (asynchronous javascript) Using pug view engine to render. Used Bootstrap templates for a more refined front end visual/design. Used passport.js to authenticate,socket.io for chat. Used Heroku to deploy our application. Used postgres as a local database to test the backend.

Scope of work

We followed the migration set up that was given during milestone 2. For front end technical work: Jose did UI for the sign-in/register/game pages and Suman worked on the lobby, win and endgame and the protected routes. Back end work was done by Frederick and setting up the structure of the code base. John implemented passport authentication and lobby chat.

John: My duty was to implement the Passport-local strategy for authentication during sign in and establish a connection with chat inside the lobby and game. The rest of my time was an attempt to establish connection with the database using pg-promise to query data from postgresql. I got through inserting users from the sign in page and using the database to check for authentication. Implemented lobby chat using socket.io to render in the messages in the lobby. The big roadblock for me was the socket that was being used was constant and I didn't know how to set up new sockets used for the game room. Also Promise objects produced in pg-promise from grabbing data from DB -> top level was difficult to figure out and I couldn't properly use the Promise data as JSON to be able to use properly in the scripts and pug.

Suman: I was assigned the task of creating the pug template for lobby page, game win and game lost page and popups. I was assigned the task of protected routes so that the user can only visit the page if the user is logged in.

Command Line Instructions to Compile and Execute:

Check **package.json**

Look for **“start:dev”**

Change for

Mac: "start:dev": "NODE_ENV=development DEBUG=APP_NAME:* nodemon ./bin/www"

Windows: "start:dev": "SET \"NODE_ENV=development\" \"SET DEBUG=myapp:*\"& nodemon ./bin/www"

Check **.env**

Add this line

SESSION_SECRET=secret

Used for express-session

Then

&npm i

Will also migrate and create table from your own postgresql database

Make sure you have your postgres db on

Then

$npm run start:dev

Assumptions

John: I didn’t really have any assumptions because most of the technologies like postgres / figma / express were new to me. My skills in Javascript are pretty basic. I'm okay with SQL query syntax but the hardest part to understand and still now is the pg-promise package and how to process the Promises. I just wish we had more time, or I shouldn’t have taken 3 CS classes.

Suman: Most of the technologies and framework used in the project were new to me. Pug templates, postgres, figma,node, express. Pug templates were fun to do but sometimes could be tedious because it was hard to figure out the cause of the errors.

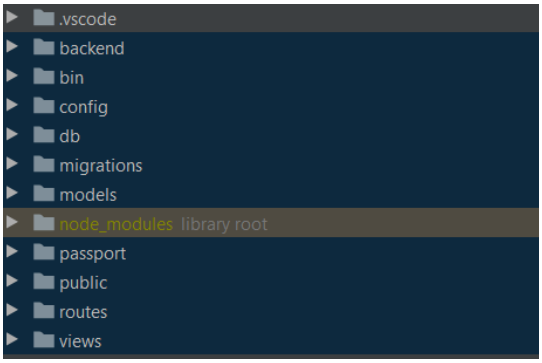
Implementation Discussion

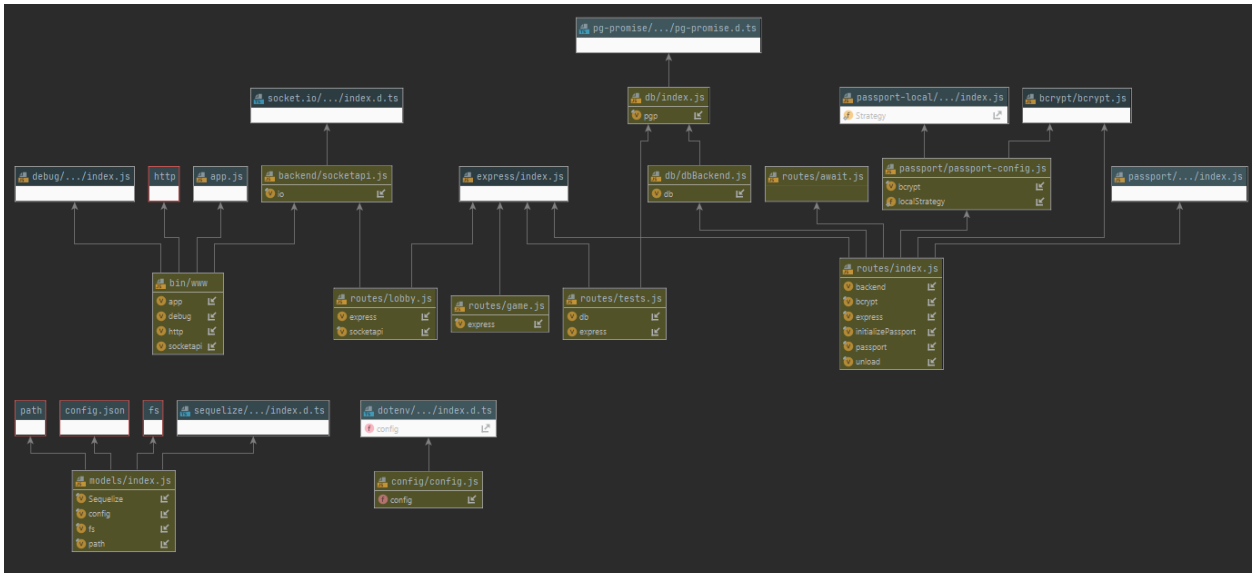
Features Planned

- User Authentication
    - Login
        - Sign in with Username and password
    - Registration
        - Sign up for account with unique username
        - Password confirmation
- Logout
    - Logout from account
    - Sent to login screen
- Lobby Chat
    - Realtime chat
    - Everyone in lobby is logged in
    - Everyone in lobby can see chat
- Game chat
    - Realtime chat
    - No logs
- Game creation
    - Choose number of players
        - 2-4
    - Game name
        - Game name is displayed in games available list
    - Make game public
    - Game id is generated
        - Players can join off id
- Game screen
    - Game is played in realtime
    - Holds game chat
    - Uno rules
        - Challenge for +4
        - Calling uno on player
- User can participate in multiple games
    - Play from multiple tabs
- User can rejoin game in progress they left
    - Retakes old hand
    - Fitted into new rotation

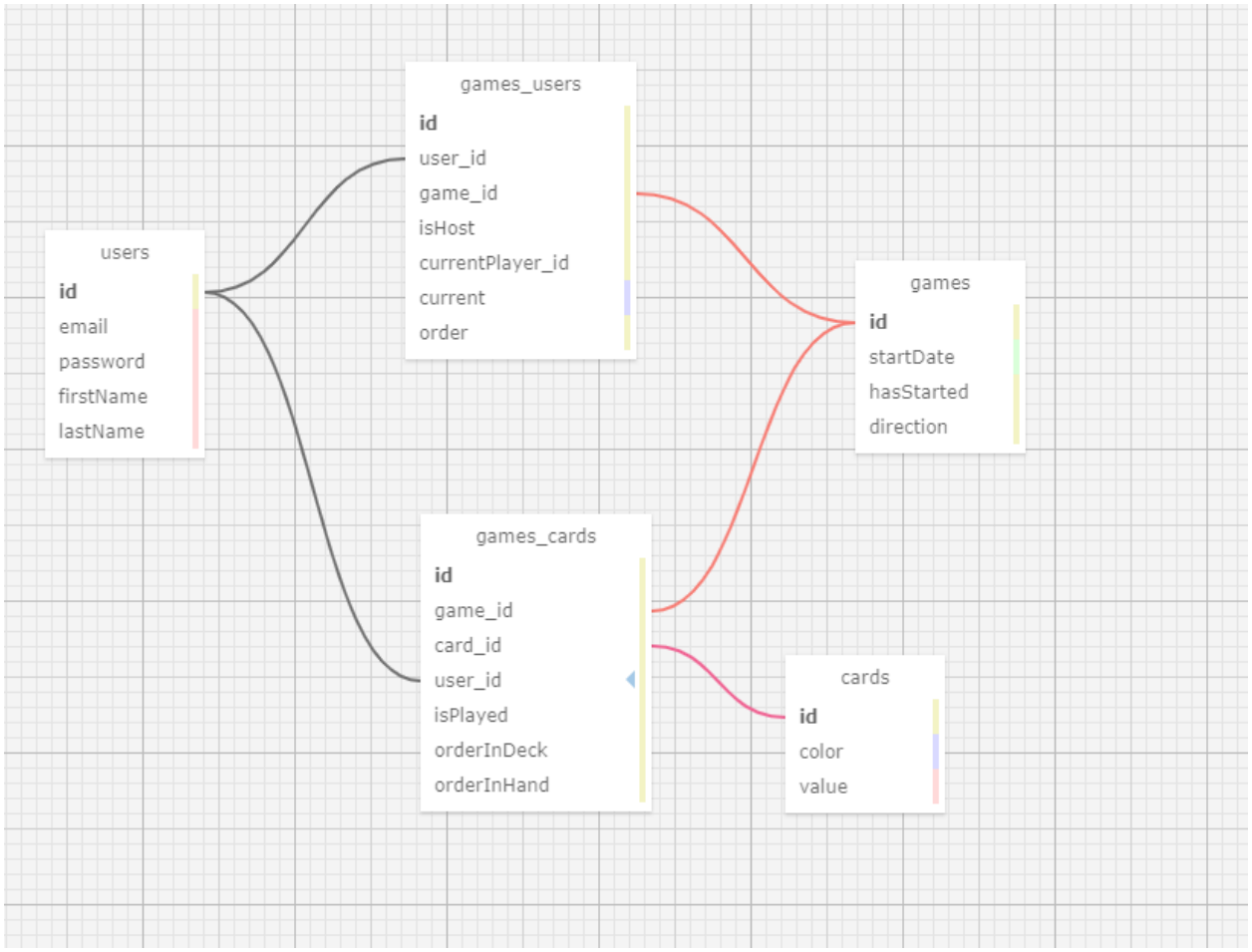
- Winner/defeat screen on game completion
    - Users can restart game with same players
    - Leave game and go to lobby
- Game is removed playable game list after completion

Code structure relationship diagram



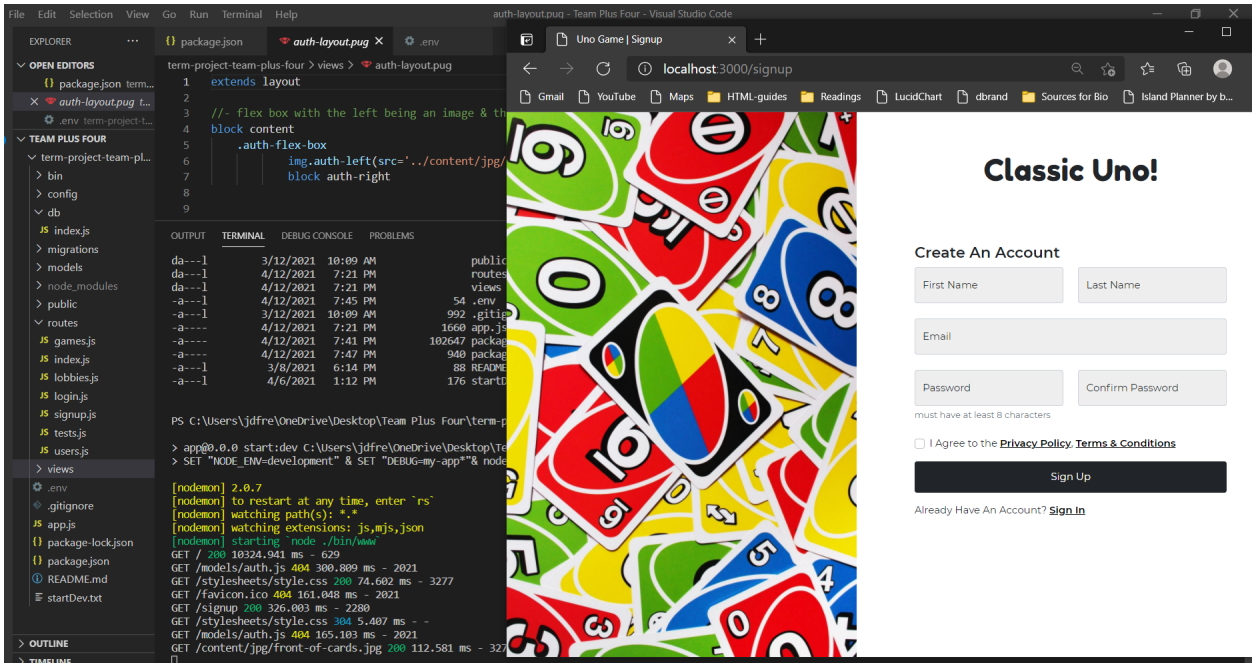


DB Relationship Diagram

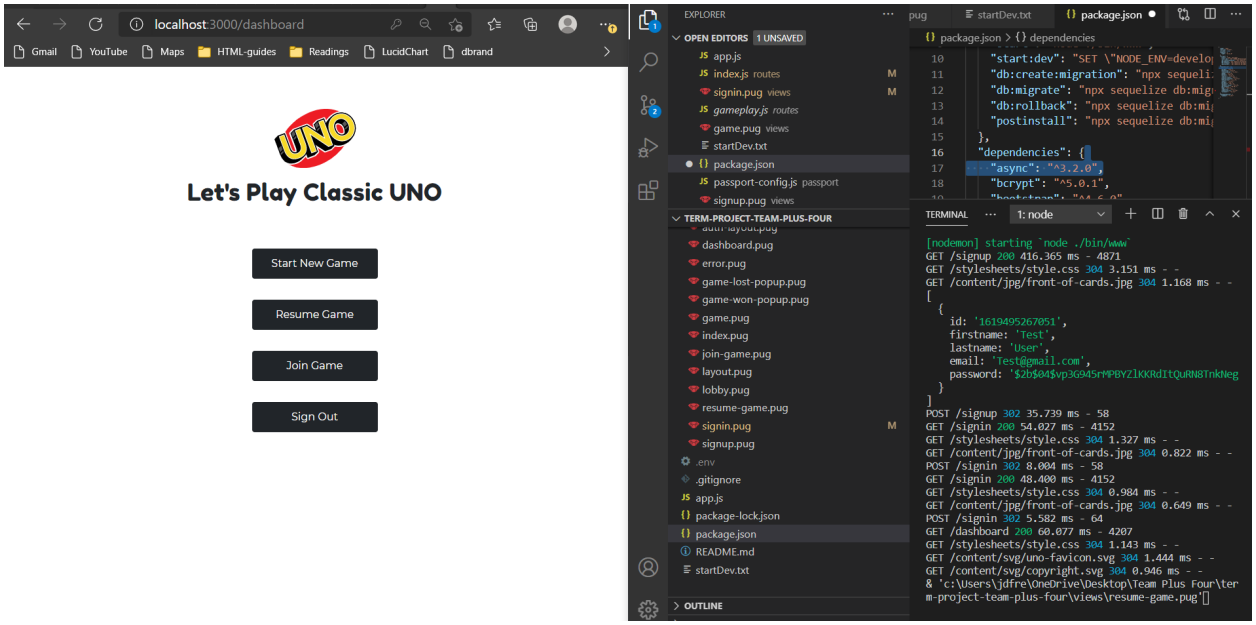


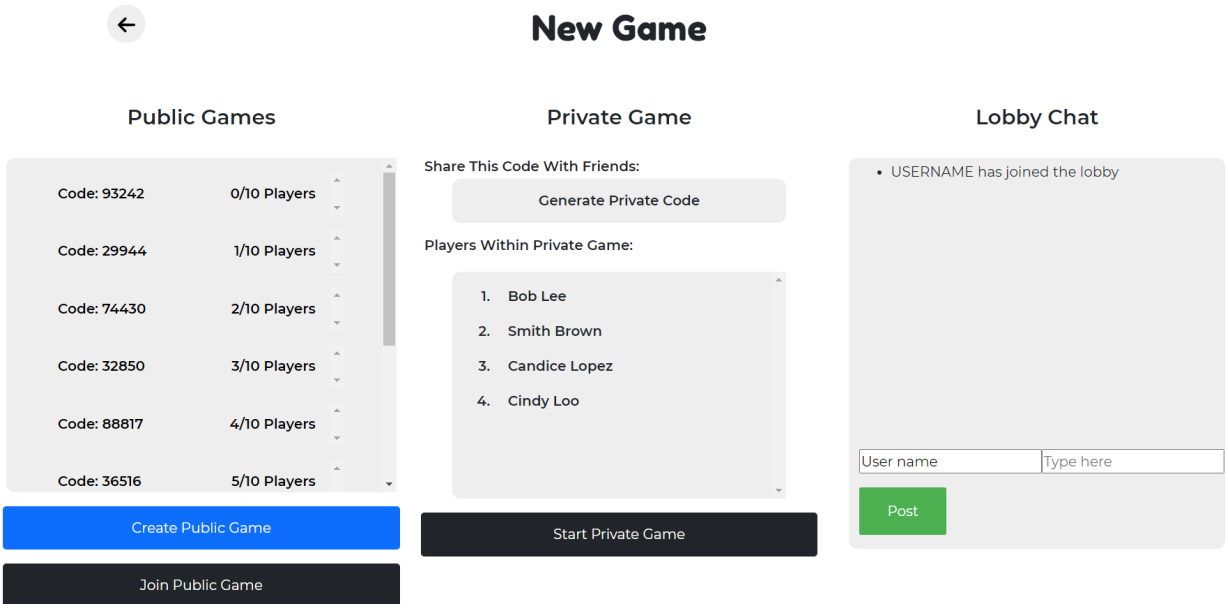
Images of application

Sign-up page

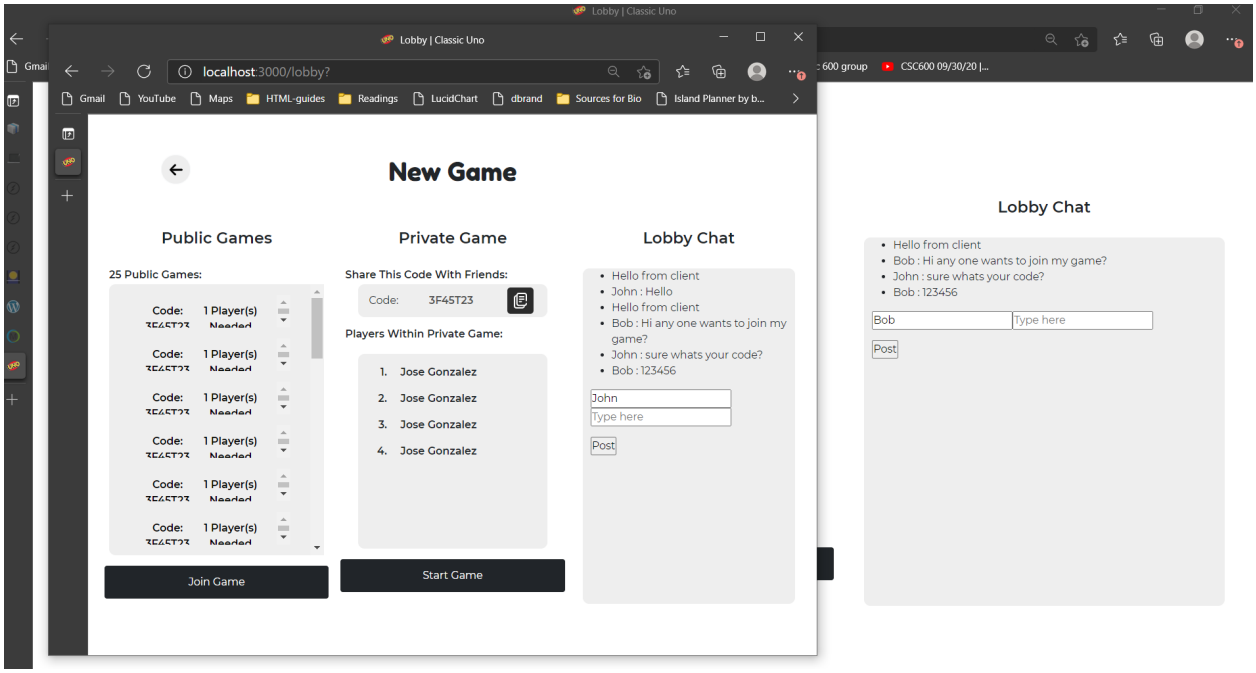


After sign in/sign up -> dash board





Lobby chat working



Results and Conclusions

What did we learn?

Suman: Honestly, I learnt a lot of things. I was familiar with git but I got more experience with github workflow and experience of working together in a git project. I learnt about Postgresql which was a totally new database to me. Pug templates were also totally new to me and they were sort of fun to do. I also learnt a few things about node and express through the project.

John: I learned that a project managing role is a must or if possible one must be tasked to keep in check of the project's progress constantly throughout development. Everyone is more than capable of working and programming but once we lose track of the project's progress the team becomes out of sync. Plus it's hard to reach out without feeling like I'm disturbing. For future reference have someone take care of the scheduling and mini milestones and bi-weekly project updates.

What challenges did you encounter and how did you overcome them?

Suman: The main challenge or sort of problem that I faced was to make the modal popup when a user wins or loses the game in the game page to work. It bugged me a little, but the problem was due to improper import of the javascript file. Another major challenge was to figure out what was causing errors in pug template. Also, it took me a while to figure out to what's protected routes and how to implement that but in the end it seemed fairly simple.

John: As I'm writing this and looking back what unexpectedly challenging was communication and just checking up with each other. We were missing a lead or someone that took the role or something similar to it. I see now that role is key to success and to keep everyone accountable and on schedule. Unfortunately we did not overcome this, and I hope this is a learning lesson to the others as much as it impacted me this semester. We did try at the beginning but the team died down and struggled to get back up.