

---

The Last Project: Brian Liu, Caden Khuu, Claire Song, Tracy Ye  
SoftDev  
PO5:  
2025-5-01  
Time Spent: 3 hrs  
TARGET SHIP DATE: 2025-6-08

## DESIGN DOCUMENT (VERSION 1)

---

### Project Description

Our project will be a multiplayer mafia game where players can create or join public lobbies and get a randomly assigned role: mafia, doctor, sheriff, or civilian. Once the game starts, it will alternate between day and night time phases. During the night, the mafia, doctor, and sheriff will pick someone to kill, save, or investigate respectively. During the day, players will type to each other in the chat and then vote on who they think the mafia is. The game ends when either all the mafia or all the innocent players are eliminated.

### Program Components

#### Front-end Components

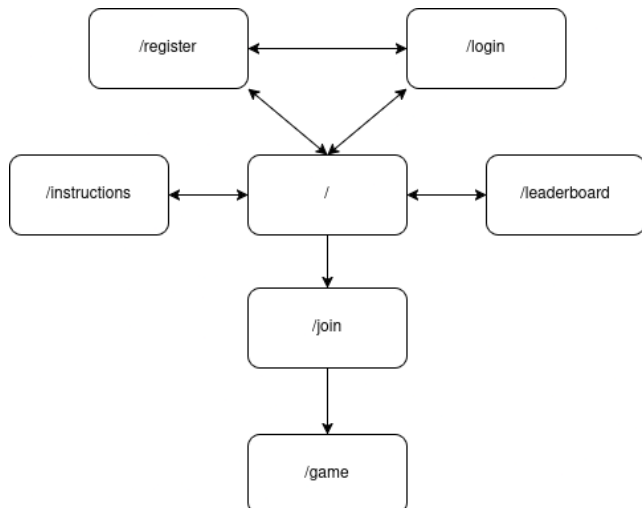
1. NodeJS Routes
  - a. /
    - i. Allows user to register, login, logout, view leaderboard, and view game instructions
    - ii. While logged in, users will be able to create, view, and join game rooms.
    - iii. Renders home.html template
  - b. /register
    - i. Registers user
    - ii. Must have a unique username
    - iii. Redirects to / -> renders home.html
  - c. /login
    - i. Cannot log in if the account is already in an active session
    - ii. Redirects to / -> renders home.html
  - d. /logout
    - i. Logs out session
    - ii. Redirects to / -> renders home.html
  - e. /join
    - i. Has the waiting room where the players can join the game
  - f. /game
    - i. Has the screen for users to play the game
  - g. /instructions
    - i. Contains the instructions of the game and a description of each role
  - h. /leaderboard
    - i. Contains a leaderboard that ranks all users based on the number of wins they have
2. Tailwind CSS - Frontend Framework
  - a. Tailwind provides user customization for accessibility, a clean user interface, and easy in-line styling that allows for different styling amongst the same HTML tags.
  - b. Features:

- i. Responsive design, animations, sidebars/navigation bars, and resizing based on screen size

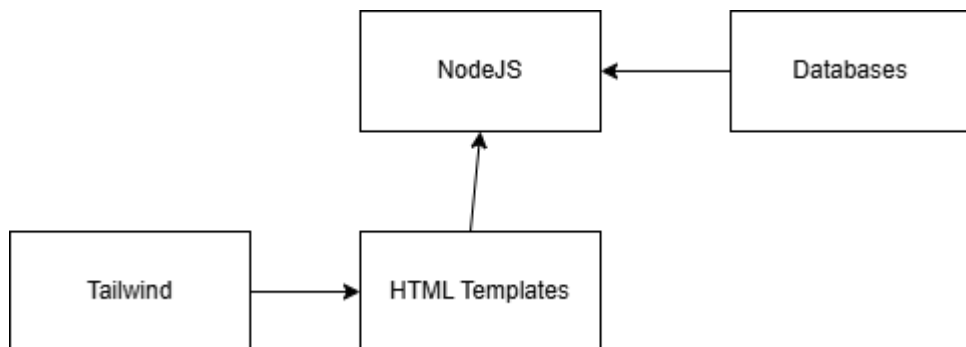
## Back-end Components

1. NodeJS
  - a. gameFunctions.js
    - i. Includes functions that facilitate players' actions in-game, such as killing other players (as the mafia), saving someone (as the doctor), voting someone out during the day, etc.
  - b. db.js
    - i. Creates db tables and inserts/updates values whenever necessary
  - c. app.js
    - i. Facilitates routing and calls functions from gameFunctions.py and db.py to allow for gameplay to work as intended. Also allows users to create and join parties.
2. SQLite Databases - Stores information from datasets
  - a. Store username and password and game states so that the program will know the roles of each player.

## Site Map



## Component Map



## Database Organization (SQL)

**User table (users)** - stores login information of users, their number of wins, and the party they are in.

username	TEXT	Unique identifier chosen by the user
password	TEXT	Password for account security

wins	int	The number of game wins the user has
party_id	int	ID of the party the user is in; -1 if not applicable

**Party table (parties)** - stores information of each party, including an ID, its owner, and whether they started a game

id	int	Unique identifier for each game room/party
owner	TEXT	Username of the player who created the party
started	boolean	Whether or not the party has started a game

**Game table (game)** - stores information about each player in the game, including their role and whether or not they are alive. Each player will be a row that is linked together by the game\_id.

game_id	int	Unique identifier for each game
user_id	int	ID of the user playing the game
role	TEXT	Role of the user
alive	boolean	Whether the user is alive

## APIs

We will not be using any APIs.

## Libraries

We will not be using any libraries or modules

## Task Breakdown

**Brian Liu (PM)** – Node.js

1. Implementing functions that help the game run using JS, such as role functionality.

**Caden Khuu** – Frontend

1. Create templates and make website look nice with Tailwind and CSS
2. Help out with NodeJS

**Claire Song** – SQLite Database

1. Creating tables to keep track of user information and player information

**Tracy Ye** – Node.js

1. Linking pages and backend using NodeJS
2. Implement multiplayer, party system, and real-time updating of information within the game.

Errors Warnings Info Logs Debug

[react-c](#)