Milestone 1: Game Decision & Wireframes

By Team Project2

Project Overview – Leslie

Our project is an online multiplayer version of the card game UNO, where players can join game rooms, draw and play cards, and compete to reach zero cards first.

The application will feature user authentication, turn-based gameplay interactions, and a visually engaging interface that mimics the original UNO experience.

Players will be able to create or join rooms, track turns, and use game logic such as skips, reverses, and wild cards.

Features - Griscelda

Required Features

- Render Deployment
  - Game should be running on Render without crash
- Node.js + Express.js
  - Server-rendered pages should be valid, handle errors, and follow correct routing
    - For example: Home page → sign up → queue → game → end scoreboard
- Authentication with SQL Database
  - Users → User Id, email, password, display name
  - Rooms → room id, host id, status
  - Room members → room id, user id
  - Matches → started at, ended at, winner id (for scoreboard)
- Game should support multiple players at the same time without crashing.(simultaneous connections)

Features - Griscelda

Game Specific Features

- Home Page
  - Will be a minimal screen with title and play button
- Authentication
  - After hitting the play button users will be prompted to sign in / sign up
- Lobby & Invite
  - Once signed in the users will be able to share invite links with friends who will land in the same queue room. Users will be able to see scoreboard and their user profile
- Basic UNO Gameplay
  - Once in a game users will be able to Skip, Reverse, Play Wild card, and draw +2/4
  - Basic UNO Gameplay
- UI/UX
  - Responsive layout with highlighted playable cards (popped-out cards)
  - End game scoreboard (if time allows)

Technologies - Diego

Back End

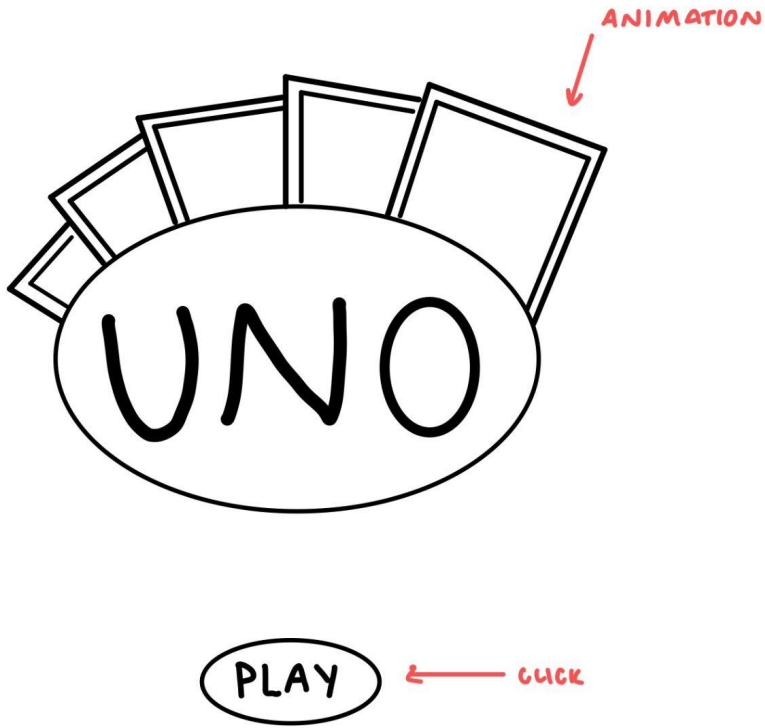
1) **Render** - used to host our full-stack application, including our server and database
2) **Node.js** - will be used to run the server-side code and handle game logic
3) **Express.js** - manages the server routes, API communication, and user authentication.
4) **PostgreSQL**- Create our database and store user accounts, match results, and game rooms
5) **Socket.IO -** used for real-time communication between front end and back end

Front End

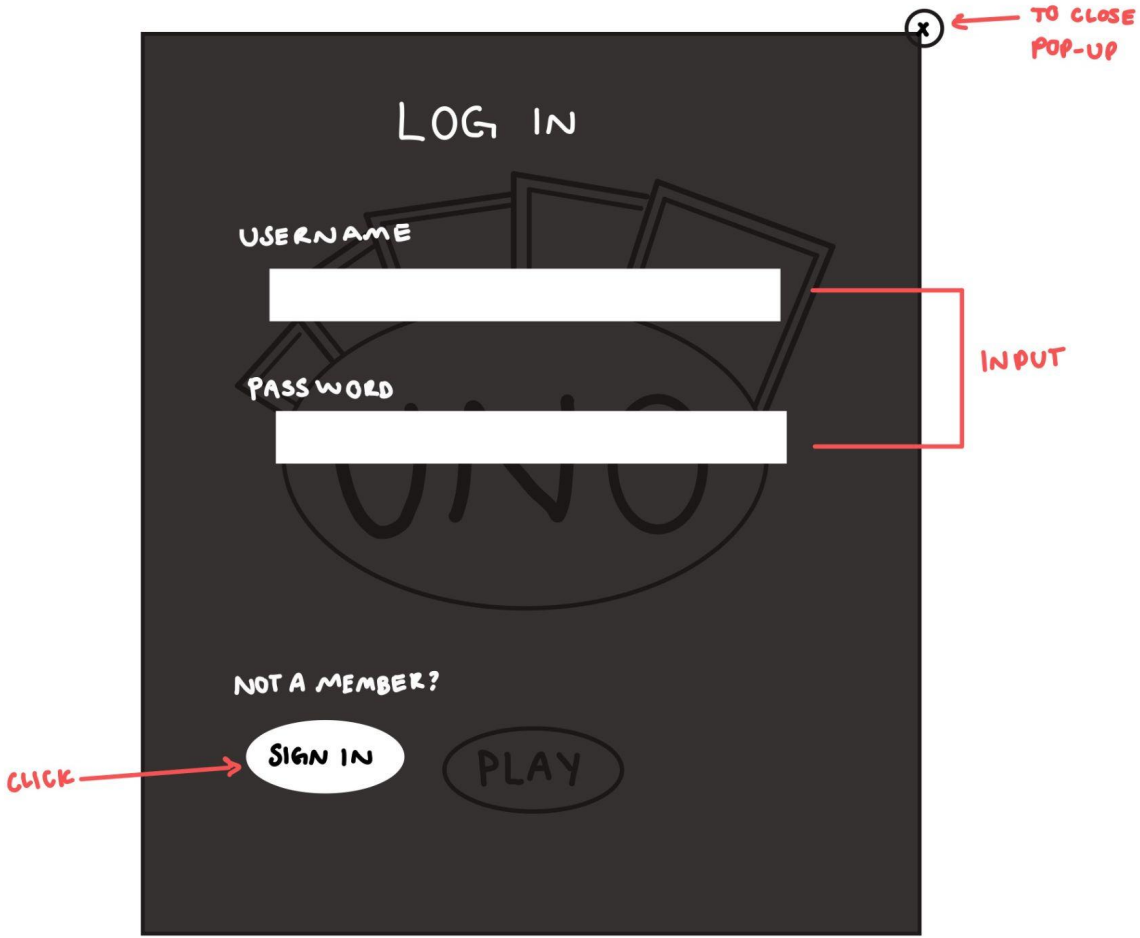
1) **HTML** - used to build the structure of our web pages
2) **CSS** - style the pages and help create a more visual layout for the project
3) **JavaScript** - will communicate with our backend code and will update the game in real time.

Wireframes – Chrissa

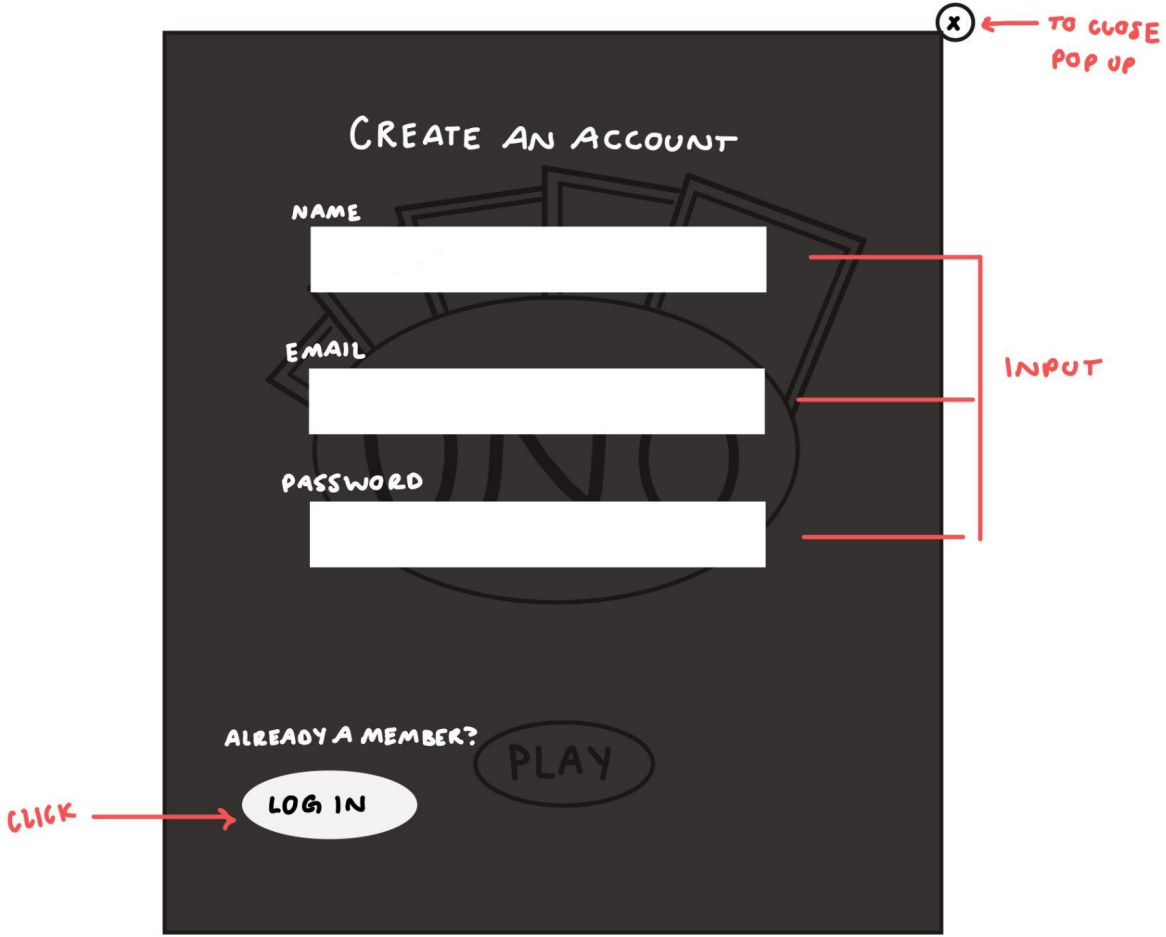
LANDING PAGE



LOG IN


TO CLOSE POP-UP
LOG IN
USERNAME
PASSWORD
INPUT
NOT A MEMBER?
CLICK
SIGN IN
PLAY

SIGN UP


TO CLOSE POP UP
CREATE AN ACCOUNT
NAME
EMAIL
PASSWORD
INPUT
ALREADY A MEMBER?
LOG IN
PLAY
CLICK

SIGNED-IN LANDING

READY TO PLAY!

[USERNAME]

[#] GAMES WON
[#] FRIENDS
[#] STREAK

← COLLAPSE-ABLE

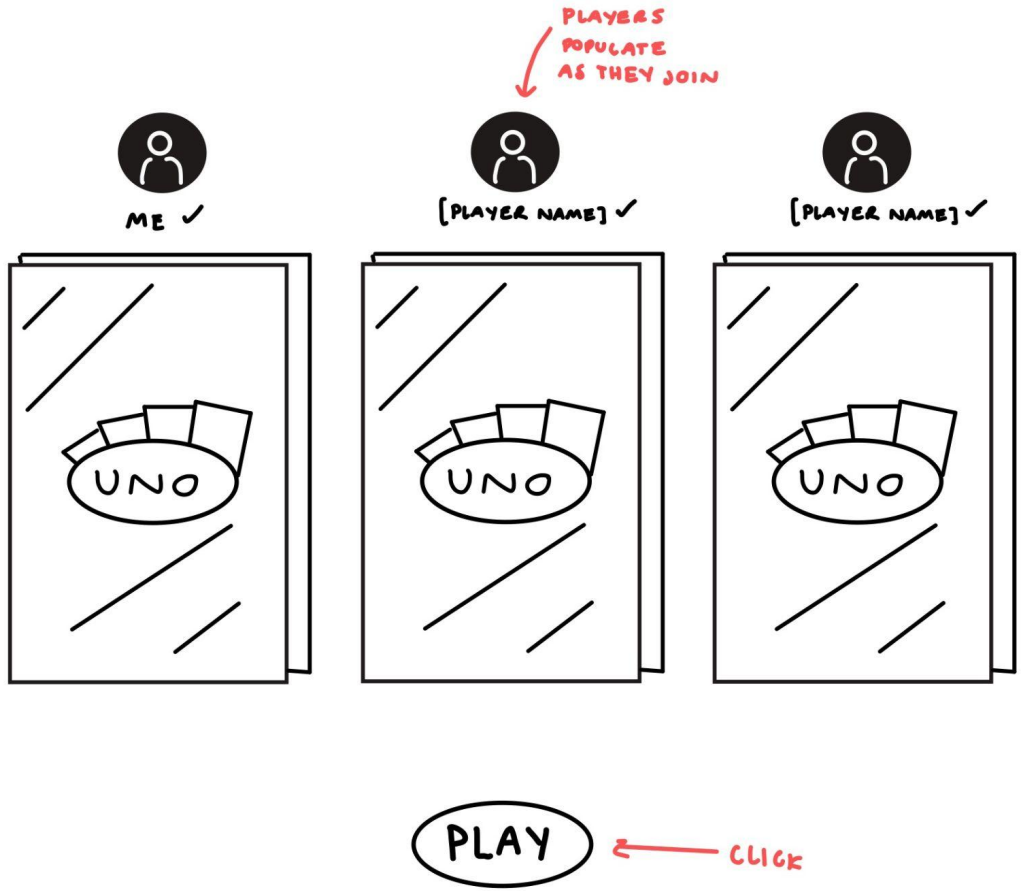
INVITE YOUR FRIENDS!

[INVITE LINK]

COPY LINK

NAVIGATE TO GAME ROOM

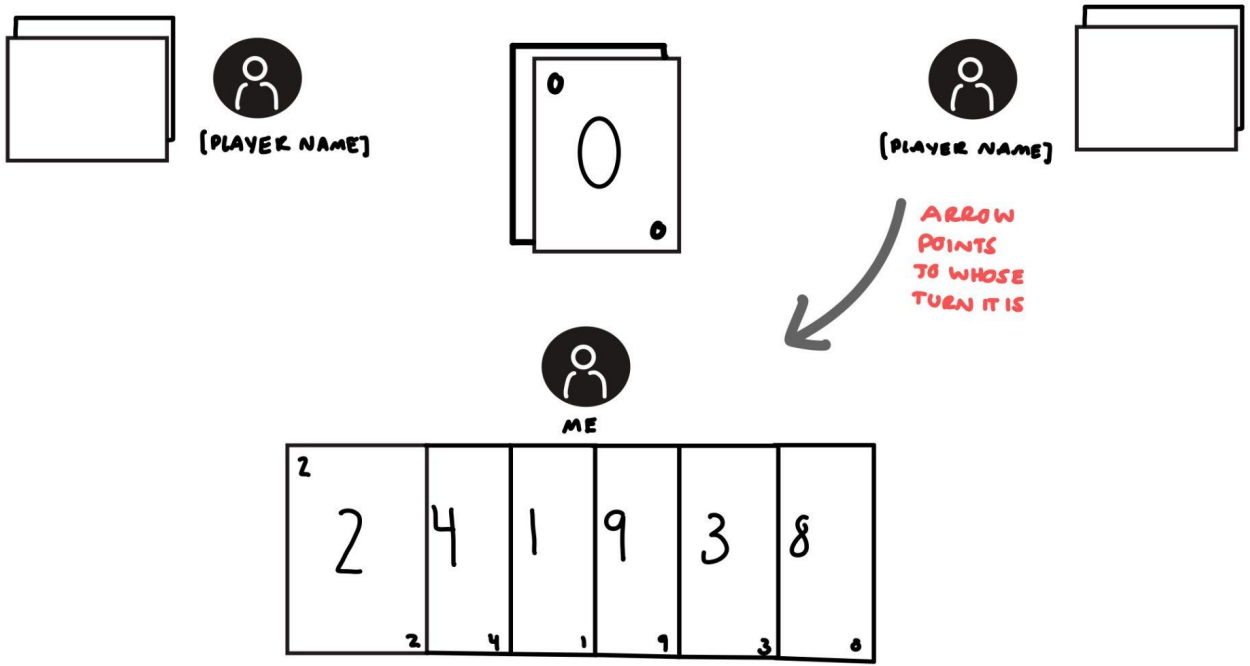
GAME WAITING ROOM



GAME ROOM

? ← INSTRUCTIONS

TO EXIT → ⊗



RESULTS

TO EXIT →

1 [PLAYER NAME]

2 [PLAYER NAME]

3 [PLAYER NAME]

CLICK → PLAY AGAIN