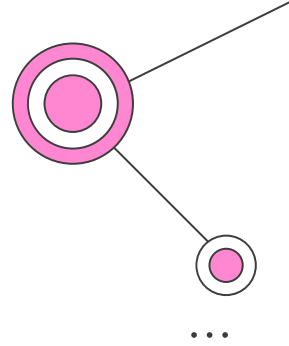
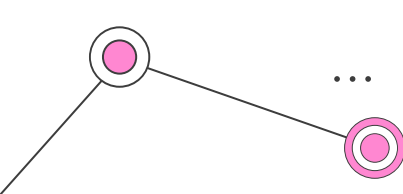


# Rinder

(R-in-der)



# TEAM MEMBERS

**Justin Goh**

Front/Backend  
Developer

**Austin Schene**

Front/Backend  
Developer

**Jonny Lunney**

Front/Backend  
Developer

**Kevin Yang**

UI/Backend Developer

**Benjamin Hyde**

UI/UX/Frontend  
Developer

**Alexis Marez**

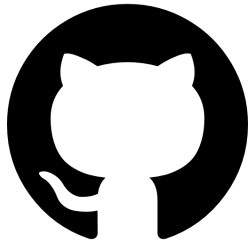
Backend Developer



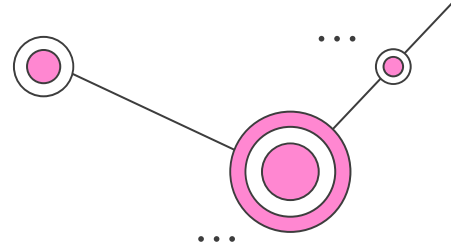
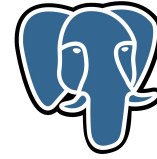
**“To preserve the fantasy of mere mortal kin  
dynamically cognizant”**

Finding roommates is hard. We just made it slightly easier.

...



## Tools Used



VCS: GitHub - Version control, stored our branches - Rating: 5

Database: PostgreSQL - Database for our backend - Rating: 5

IDE: VSCode - Where code was developed - Rating: 4

UI Tools: EJS - UI tools, Routing data to pages - Rating: 4

Application Server: Docker - Hosted Front and Backend - Rating: 5

Deployment Environment AWS: Azure - Rating: 5

External APIS: Random User Generator API Documentation: <https://randomuser.me/documentation> - Rating: 5

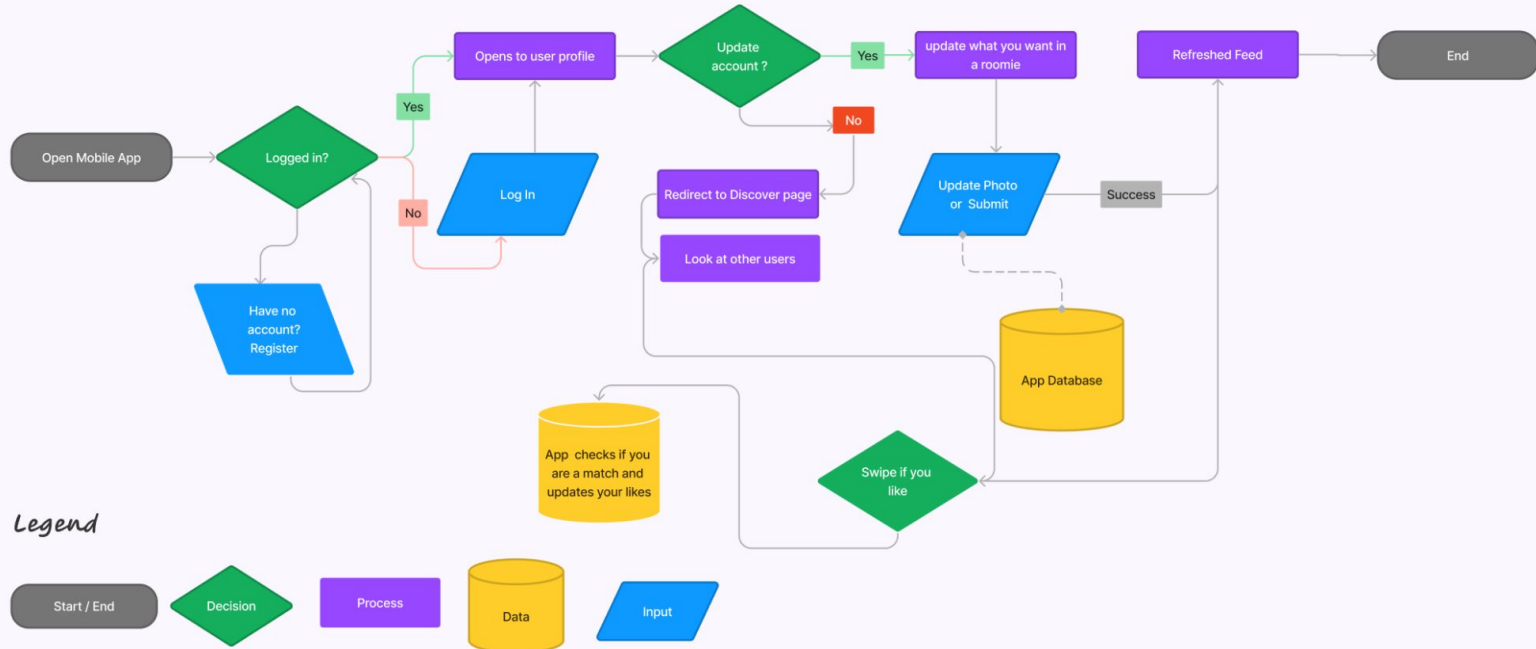
Additional Libraries: Anime.js - Javascript animation library - Rating: 5

Wireframe and Prototyping: Figma - Rating: 5

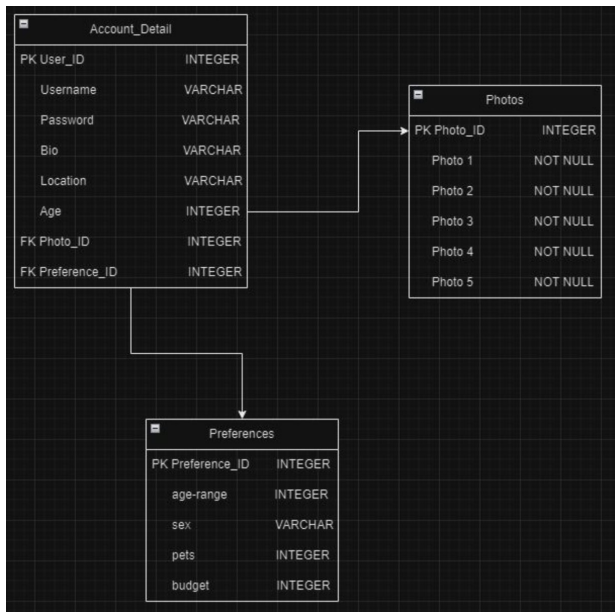


# Architecture Diagram

Use-Case  
Diagram



# Database Setup



```
1 CREATE TABLE IF NOT EXISTS preferences(  
2     preferences_id SERIAL PRIMARY KEY,  
3     age_range INTEGER NOT NULL,  
4     sex VARCHAR NOT NULL,  
5     pets INTEGER NOT NULL,  
6     budget INTEGER NOT NULL  
7 );  
8  
9 CREATE TABLE IF NOT EXISTS photos(  
10    photo_id SERIAL PRIMARY KEY,  
11    photo_1 VARCHAR NOT NULL,  
12    photo_2 VARCHAR NOT NULL,  
13    photo_3 VARCHAR NOT NULL,  
14    photo_4 VARCHAR NOT NULL,  
15    photo_5 VARCHAR NOT NULL  
16 );  
17  
18 CREATE TABLE IF NOT EXISTS users (  
19    user_id SERIAL PRIMARY KEY,  
20    username VARCHAR NOT NULL,  
21    password VARCHAR NOT NULL,  
22    first_name VARCHAR NOT NULL,  
23    last_name VARCHAR NOT NULL,  
24    bio VARCHAR NOT NULL,  
25    location VARCHAR NOT NULL,  
26    age INTEGER NOT NULL,  
27    photo_id INTEGER REFERENCES photos(photo_id),  
28    preferences_id INTEGER REFERENCES preferences(preferences_id)  
29 );  
30  
31 CREATE TABLE IF NOT EXISTS matches (  
32    liked_user_id INT,  
33    liker_user_id INT  
34 );
```

# Challenges

01

...

## Backend Structure Changed

While developing, backend structure became difficult to traverse

02

...

## Github Issues

Github became a mess at the beginning with merging issues.

## Combining Preferences and Users

Struggling to pass preferences and users through EJS. Made inner join.

03

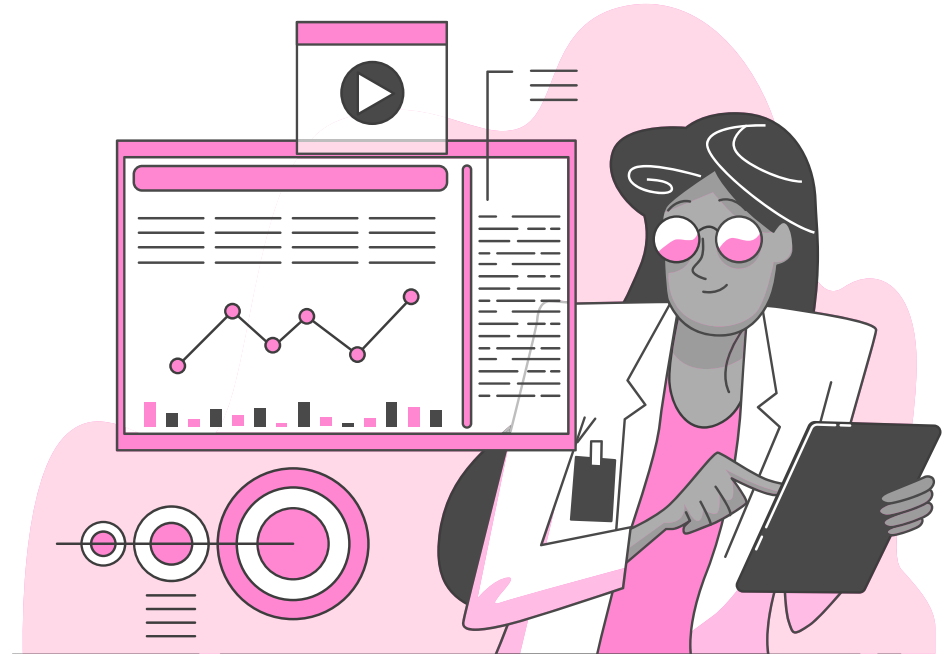
...

## Preferences and Using Local Images

Preferences wouldn't update, made use of sub-query. Used built-in middleware function in Express to insert logo.

04

...



# Frontend animations

## Purpose

More immersive and attention grabbing.

## Library: anime.js

Open source javascript library with a robust set of targets; CSS properties, SVG, DOM attributes and Javascript objects.

## More polished front end

Using this additional library allowed the front end of Rinder to be delivered with a more complete capacity to manipulate the users experience.

## Challenges

Creating a personalized object to a specified shape and orientation.

Compatibility as a partial, to be easily added or removed from any page.





# Demo Time