Features

- users can sign into the app with their email and password
- users can create recipes with ingredients and instructions
- recipes can be marked as public or private
- users can view other people's recipes
- ingredients from recipes can be added to user's grocery lists
- users can create their own occasions and assign recipes to occasions

Brainstorming

- Unique User id
- Unique grocery list
- Email
- Password
- Recipe ID
- Occasion ID
- Ingreadient table
- Ingredient ID
- Recipe table boolean true false

Table Ideas

- User table
    - Be able o store user id with their email and password
- Ingredient table
    - Store all of the ingredients
- Recipe table
    - Store all of the recipes with the date created
- Recipe_ID
    - Store all of the individual recipes's ingredients and instructions
- Grocery list table
    - Have a palace to add recipe ingredients to
- Occasions
    - List out all of the occasions

Relationships

One-to-one

- One user can make a recipe, one recipe is made by one person.

one-to -many

-

Many-to-many

- Grocery list can have many ingredients, one ingredient can be on many grocery lists.
- A recipe can have many ingredients, one ingredient can be in many recipes.
- One occasion can have many recipes, one recipe can have many occasions

- User table
  - User_id (SPK)
  - Username (varchar)
  - Email (varchar)
  - Password (varchar)
    - I need a place to reference each user by their ID, give them a username for their interaction through the app, and store the email and password.
- Ingredient table
  - ingredient_ID (SPK)
  - Ingredient Name (varchar)
    - Having on egiant database of ingredients is better than listing them out in each individual recipe.
- Recipe table
  - User_ID (FK)
  - Recipe ID (SPK)
  - Instructions (text)
  - Recipe Name (varchar)
  - Public (boolean)
    - If we can have a recipe list with each recipe, who created it, and the name, it would make sorting through it much easier. We can also check to see if the recipe is public or not.
- Recipe_ingredients
  - Recipe ID (FK)
  - Ingredient ID (FK)
    - This list will allow the recipe list and ingredient list to communicate with each other better. This way we can list out each ingredient in each recipe.
- Grocery list table
  - Ingredient_ID (FK)
  - Recipe ID (FK)
  - User ID (FK)
    - This can communicate with the other tables (mostly the recipe_ingredients table), allowing the user to put all of the ingredients for a specific recipe in here, or just pick their own ingredients.
- Occasions
  - Occaision_id (SPK)

    - Occasions Name (varchar)
        - This will hold all of our occasions that we choose to create.
- Recipe_occasions
    - Recipe_id (FK)
    - Occasion_id (FK)
        - This will allow the recipe and occasions tables to communicate to each other, assigning multiple recipes to multiple occasions.

Postgres Code

```
create table user_list (
  user_id serial primary key,
  username varchar(20),
  email varchar(100),
  password varchar(255)
);

create table recipe_list (
  recipe_id serial primary key,
  recipe_created_by integer,
  recipe_name varchar(50),
  public boolean,
  instructions text,
  FOREIGN KEY(recipe_created_by) REFERENCES user_list(user_id)
);

create table ingredients (
  ingredient_id serial primary key,
  ingredient_name varchar(30)
);

create table recipe_ingredients (
  recipe_id integer,
  ingredient_id integer,
  quantity varchar(20),
  FOREIGN KEY(recipe_id) REFERENCES recipe_list(recipe_id),
  FOREIGN KEY(ingredient_id) REFERENCES ingredients(ingredient_id)
);

create table grocery_list (
  user_id integer,
  recipe_id integer,
  ingredient_id integer,
  quantity varchar(20),
  FOREIGN KEY(user_id) REFERENCES user_list(user_id),
  FOREIGN KEY(recipe_id) REFERENCES recipe_list(recipe_id),
  FOREIGN KEY(ingredient_id) REFERENCES ingredients(ingredient_id)
);

create table occasions (
  occasions_id serial primary key,
  occasions_name varchar(50)
);

create table recipe_occasions (
  recipe_id integer,
  occasions_id integer,
  FOREIGN KEY(recipe_id) REFERENCES recipe_list(recipe_id),
  FOREIGN KEY(occasions_id) REFERENCES occasions(occasions_id)
)
```