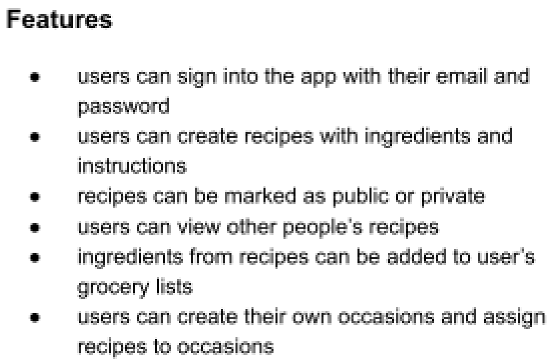
Brainstorming

- Users:
  - User_id
  - User_password
  - User_email
  - First_name
  - Last_name
  - Grocery_id
  - occasion_id
- Ingredients
  - Ingredient_id
  - ingredient_name
- Recipes
  - recipes_id
  - Instructions
  - Public/Private
  - ingredient_id
- Grocery Lists
  - Grocery_id
  - Ingredient_id
- Occasions
  - occasion_id
  - recipes_id



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

Table Ideas

- User Table:
  - User_id
  - User_password
  - User_email
  - First_name
  - Last_name

    - Grocery_id
    - occasion_id
- Ingredient Table
    - Ingredient_id
    - ingredient_name
- Recipe Table
    - recipe_id
    - Instructions
    - Public/Private
    - Ingredient_id
    - user_id
- Grocery Table
    - Grocery_id
    - Ingredient_id
    - user_id
- Occasion Table
    - occasion_id
    - Recipe_id
    - user_id

Relationships

One to One

One to Many

- Recipes => Ingredients
- User => Occasion
- User => Grocery
- Grocery => Ingredients
- Occasion => Recipes
- Users => Recipes

Many to Many

- Ingredients ⇔ Users

Columns

- List out each table's respective columns in the table's sub-section and explain for each column:
  - why you'll be storing that data
  - and why you chose the data type you did
- User Table:
  - User_id: unique user identifier, serial primary key so it's always unique
  - User_password: so user can log in, VARCHAR so it limits characters, but can be any kind of character
  - User_email: so user can log in, VARCHAR so it limits characters, but can be any kind of character
  - First_name: to personalize user profile, VARCHAR so it limits characters, but can be any kind of character
  - Last_name: to personalize user profile, VARCHAR so it limits characters, but can be any kind of character
  - Grocery_id: to attach the grocery list to the user, references the already created grocery id
  - Occasion_id: to attach the occasion to the user, references the already created occasion id
- Ingredient Table
  - Ingredient_id: unique identifier for ingredients.
  - Ingredient_name: every ingredient needs a name.
- Recipe Table
  - recipe_id: unique recipe identifier, serial primary key so it's always unique
  - Instructions: need to know the order or adding ingredients
  - Public/Private: Users can set it public or private
  - Ingredient_id: the recipes table needs the ingredients table
  - User_id: each user needs to be associated with recipe(s).
- Grocery Table

    - Grocery_id: unique grocery identifier, serial primary key so it’s always unique
    - Ingredient_id: the recipes table needs the ingredients table
    - User_id: each user needs to be associated with recipe(s)
- Occasion Table
    - occasion_id: unique occasion identifier, serial primary key so it’s always unique
    - Recipe_id: unique recipe identifier, serial primary key so it’s always unique. Occasions call for certain recipes
    - User_id: each user needs to be associated with recipe(s)

Create Tables

```
CREATE TABLE user (
user_id SERIAL PRIMARY KEY,
user_password VARCHAR(500),
user_email VARCHAR(255),
first_name VARCHAR(50),
last_name VARCHAR(50)
);


CREATE TABLE ingredient (
ingredient_id SERIAL PRIMARY KEY,
ingredient_name VARCHAR(250)
);


CREATE TABLE recipe (
recipe_id SERIAL PRIMARY KEY,

instructions TEXT,

private BIT,

ingredient_id INTEGER NOT NULL REFERENCES ingredient(ingredient_id),

user_id INTEGER NOT NULL REFERENCES user(user_id)

);




CREATE TABLE grocery (

grocery_id SERIAL PRIMARY KEY,

ingredient_id INTEGER NOT NULL REFERENCES ingredient(ingredient_id),

user_id INTEGER NOT NULL REFERENCES user(user_id)

);




CREATE TABLE occasion (

occasion_id SERIAL PRIMARY KEY,

recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),

user_id INTEGER NOT NULL REFERENCES user(user_id)

);



INSERT INTO user (user_password, user_email, first_name, last_name)

VALUES ('password', 'email@email.com', 'First', 'Last');
```