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

Sign in function: tracks user_id, email, references a password hash list (separate). Also could include first/last name, user bio, cooking skill level (novice, home cook, professional chef, etc.), maybe a picture?

Recipe creation: allows user to input a list of ingredients and detailed instructions (text box). Ingredients stored individually in separate text field/list so easier to clip to grocery list later? Boolean value for public/private?

Recipe search to find other recipes: search by author or title or ingredient or category (season/good for kids/vegetarian/etc). ←This could pull from a table of tags. Recipe display includes info from author's account, picture(?), then ingredient list & instructions. A user can "pin" or "fav" a recipe, so they can get to it later?

Items from the recipe can be selected to be added to user's grocery lists. (Going to need ingredient IDs, then.) Grocery lists is for an individual user, includes ingredient IDs and text.

"Occasions" are basically groups of recipes & users. Can involve multiple users & multiple recipes. People can sign up as "attending" and then indicate recipes that they are planning to make for the Occasion.

BrainStorming:

Table Ideas:

- **Customers**
  Customer_id
  Customer_user
  email
  First_name
  Last_name

- **Customer Details**
  - Short bio
  - Skill level
  - Picture
  - Customer_id (from customer sheet)

- **Passwords**
  Customer_id (from customer sheet)
  Customer_password

- **Recipes**
  - Customer_id (from customer sheet)
  - Customer display name
  - Recipe id
  - List of ingredients (from Ingredient sheet)
  - Instructions (text)
  - Category tags
  - picture(s)
  - Is_private? (boolean)

- **Fav Recipe List**
  - Customer_id
  - Recipe id

- **Ingredients**

    - Ingredient_id
    - Ingredient name
    - Store link

- **Grocery Items**
    - Grocery list id
    - User id
    - Ingredient_id (from ingredients)
    - Ingredient name (from ingredients)
    - Store links (from ingredients)

- **Occasions**
    - Occasion_id
    - User ids
    - Recipe ids
    - Occasion description

Relationships:

- One to one
    - Customers & customer details, because each customer only has one set of details
    - Customer & passwords because each customer only has one password
    - Customer & Fav List
- One to many
    - Customers → Recipes, because one customer can post multiple recipes
    - Recipe → Ingredients because pretty much every recipe has more than one ingredient.
    - Fav List → Recipes, because it's a list of multiple recipes
- Many to many
    - Occasions ⇔ Users because multiple users attend an occasion, and users can sign up for multiple occasions.

- Occasions ⇔ Recipes because multiple recipes can be assigned to an occasion, and a recipe can be assigned to multiple occasions.

```
CREATE TABLE customers ( customer_id SERIAL PRIMARY KEY, username VARCHAR(25)
NOT NULL, email VARCHAR(75) NOT NULL, first_name VARCHAR(40), last_name
VARCHAR(40) ); CREATE TABLE passwords ( customer_id INTEGER NOT NULL
REFERENCES customers(customer_id), password_hash VARCHAR(500) NOT NULL );
CREATE TABLE customer_details( customer_id INTEGER NOT NULL REFERENCES
customers(customer_id), bio VARCHAR(250), skill_level VARCHAR(25), picture
VARCHAR(500) ); CREATE TABLE ingredients( ingredient_id SERIAL PRIMARY KEY,
ingredient_name VARCHAR(50), store_link VARCHAR(500) ); CREATE TABLE category(
category_id SERIAL PRIMARY KEY, category_name VARCHAR(25) ); CREATE TABLE
recipes ( customer_id INTEGER NOT NULL REFERENCES customers(customer_id),
username VARCHAR(25) REFERENCES customers(username), recipe_id SERIAL PRIMARY
KEY, recipe_name VARCHAR(50), ingredient_id INTEGER REFERENCES
ingredients(ingredient_id), ingredient_name VARCHAR(50) REFERENCES
ingredients(ingredient_name), store_link VARCHAR(500) REFERENCES
ingredients(store_link), instructions VARCHAR(500), category_tags VARCHAR(25)
REFERENCES category(category_name), picture VARCHAR(500), is_private BOOL NOT
NULL ); CREATE TABLE list ( customer_id INTEGER NOT NULL REFERENCES
customers(customer_id), recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id),
recipe_name VARCHAR(50) REFERENCES recipes(recipe_name) ); CREATE TABLE
grocery_list ( grocery_list_id SERIAL PRIMARY KEY, customer_id INTEGER NOT NULL
REFERENCES customers(customer_id), ingredient_id INTEGER REFERENCES
ingredients(ingredient_id), ingredient_name VARCHAR(50) REFERENCES
ingredients(ingredient_name), store_link VARCHAR(500) REFERENCES
ingredients(store_link) ); CREATE TABLE occasions( occasion_id SERIAL PRIMARY KEY,
customer_id INTEGER NOT NULL REFERENCES customers(customer_id), recipe_id
INTEGER NOT NULL REFERENCES recipes(recipe_id), occasion_desc VARCHAR(100) );
```