Summary

We want to create a recipe creating/sharing and grocery list app. You'll be planning out what tables we'll need, what information they'll store, and how the data will relate to each other.

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

Step 1: Brainstorming

User profile:
User name
Email
Password
My grocery list
My recipes
My occasions

Create recipes
Share recipes
Make public/private
Instructions
Ingredients
Grocery list
Create occasions
View: my recipes, other's recipes, my grocery list, my occasions
Lists

Step 2: Table Ideas

Users:
user name
email
password
~~*grocery list*~~
~~*recipes*~~
~~*occasions*~~

Recipe:
name of recipe
ingredients
instructions
public/private setting

Grocery List:
ingredients
quantity

Occasion:
names
recipe

Ingredients:
name
quantity

Step 3: Relationships

“One-to-one”

-User table and Grocery List table because that list is specific to each individual user.

“One-to-many”

- User table and Occasion table because the user can choose multiple different occasions but that occasion is specific only to the one user.

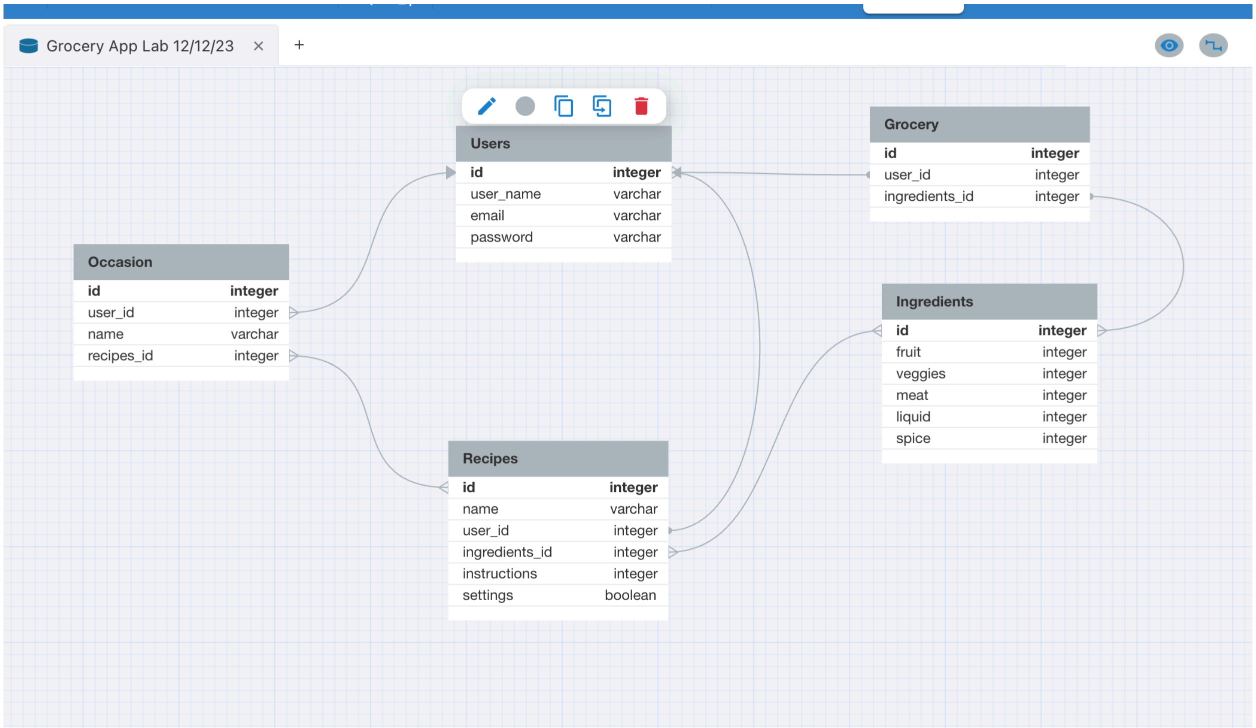
-Recipe table and Ingredients table because only 1 set of ingredients is used for each recipe, but each ingredient can be used in many different recipes.

-Grocery table and Ingredients table because every grocery list requires 1 specific set of ingredients but each ingredient can be used for many peoples’ lists.

“Many-to-many”

-User table and Ingredients table because users can use different ingredients and the same ingredients can be used by different users.

-Recipe table and Occasion table because the recipe can be used for any occasion and there’s many recipe options for every occasion.



Users

id	integer
user_name	varchar
email	varchar
password	varchar

Grocery

id	integer
user_id	integer
ingredients_id	integer

Occasion

id	integer
user_id	integer
name	varchar
recipes_id	integer

Ingredients

id	integer
fruit	integer
veggies	integer
meat	integer
liquid	integer
spice	integer

Recipes

id	integer
name	varchar
user_id	integer
ingredients_id	integer
instructions	integer
settings	boolean

“Columns”

Users table:

-user_name to identify person using app, varchar max 20 characters
-email for login, varchar max 50 char
-password for login, varchar max 500 (for long hashed password)

Grocery table:

-user_id to link the list to the user that created it (integer for their assigned ID in users table)
-ingredients_id that was added from where they selected the ingredients in the app (integer from the specific ingredient’s ID)

Ingredients table:

-listed variety of ingredients and used integer to show quantity needed. Could make table longer with every possible ingredient listed for specificity.

Recipes table:

-name of recipe varchar for limited characters
-user_id shows who posted that recipe
-ingredients_id: integer of specific ingredients chosen from ingredients table
-instructions (should be type text as is a text block of prep/cook info)
-settings are either public or private (boolean)

Occasion table:

-user_id to who selected the occasion (integer from their user table)
-name of occasion (Christmas, birthday, Thanksgiving, etc)
-recipe_id of the specific recipe that the user wants to assign to the occasion (integer)

pgAdmin SQL code:

```
CREATE TABLE users (
id SERIAL PRIMARY KEY,
username VARCHAR(20),
password VARCHAR(500),
email VARCHAR(50)
);

CREATE TABLE Ingredients (
id SERIAL PRIMARY KEY,
   fruit INTEGER,
   veggies INTEGER,
   meat INTEGER,
   liquid INTEGER,
   spice INTEGER
);

CREATE TABLE Grocery (
id SERIAL PRIMARY KEY,
user_id INTEGER NOT NULL REFERENCES users(id),
ingredients_id INTEGER NOT NULL REFERENCES ingredients(id)
);

CREATE TABLE recipes (
id SERIAL PRIMARY KEY,
name VARCHAR(100),
user_id INTEGER NOT NULL REFERENCES users(id),
ingredients_id INTEGER NOT NULL REFERENCES ingredients(id),
instructions TEXT,
   settings BOOLEAN
);

CREATE TABLE occasion (
id SERIAL PRIMARY KEY,
name VARCHAR(100),
user_id INTEGER NOT NULL REFERENCES users(id),
recipes_id INTEGER NOT NULL REFERENCES recipes(id)
);
```