Brainstorming

users can sign into the app with their email and password

- User id
- User email
- User Password
- User name
- User phone
- User address

users can create recipes with ingredients and instructions

- Recipe id
- Recipe name
- Recipe instructions
- Recipe ingredients list
- Recipe created date
- Recipe created-by user id

recipes can be marked as public or private

- Recipe public boolean
- List of private recipes
- List of public recipes

users can view other people's recipes
ingredients from recipes can be added to user's grocery lists

- Ingredients id
- Ingredients names
- Ingredients price?
- Grocery list id
- Grocery list user id
- Grocery list items
- Grocery list items quantities

users can create their own occasions and assign recipes to occasions

- Occasion id
- Occasion user id
- Occasion name
- Occasion recipes list
- Occasion description

Table Ideas

Users
This table will hold information about the user. Each row will be an individual user.

Recipes
This table will hold information about recipes that users post.

Ingredients
This table will hold information about the ingredients needed for a recipe.

Grocery list
This table will hold a list of the ingredients in a user's grocery cart. Each row will be an ingredient.

Occasions
This table will hold a list of recipes associated with different occasions.

Users_Groceries
This table will contain the relationship between a user (user id) and their grocery list (ingredients ids)

Users_Recipes
This table will contain the relationship between a user (user id) and their recipes (recipe id)

Users_Occasions
This table will contain the relationship between a user (user id) and their occasions (occasion id)

Occasions_Recipes
This table will contain the relationship between an occasion (occasion id) and its recipes (recipe id)

Grocery_List_Ingredients
This table will contain the relationship between a grocery list (list id) and its ingredients (ingredient id)

Recipes_Ingredients
This table will contain the relationship between a recipe (recipe id) and its ingredients (ingredients id)

Relationships

One-to-One

One-to-Many

- Users => Recipes - each user will have many recipes and each recipe belongs to one user

- Users => Occasions - each user will have many occasions and each occasion will have one user

Many-to-Many

- Users => Ingredients - each user will have many ingredients and each ingredient will have many users
- Grocery_List => Ingredients - each grocery list will have many recipe ingredients
- Recipes => Ingredients - each recipe will have many ingredients and each ingredient can be in many recipes
- Occasions => Recipes - each occasion will have many recipes and there will be only one of each occasion

Columns

Users

- id - Needs a primary key. Primary keys should be integers.
- name - Need to know the user's name. Needs to be text, but length should be limited.
- email - Need to know the user's email. Needs to be text, but length should be limited.
- password - Need to know the user's password. Needs to be text, not sure how long hashes are.
- address - Can know the user's address. Needs to be text, but length should be limited.
- phone - Can know the user's phone number. Needs to be an integer with a length of 10.

Recipes

- id - Needs a primary key. Primary keys should be integers.
- created_by - Need to know who created the recipe. Needs to match user id data type.
- name - Need to know the recipe's name. Needs to be text, but length should be limited.
- instructions - Need to know the recipe's instructions. Needs to be text because it will contain letters and numbers and could be quite long.
- public - Need to know if the recipe is public or private. Should be a boolean because public flag will be true or false.

Occasions

- id - Needs a primary key. Primary keys should be integers.
- user_id - Need to know who created the occasion. Needs to match user id data type.
- name - Need to know the occasion's name. Needs to be text, but length should be limited.
- description - Optional description of the occasion. Needs to be text, but length should be limited.

Ingredients

- id - Needs a primary key. Primary keys should be integers.
- name - Need to know the ingredient's name. Needs to be text, but length should be limited.

Occasions_Recipes

- occasion_id - Need to know which occasion has which recipes. Needs to match occasion id data type.
- recipe_id - Need to know which occasion has which recipes. Needs to match recipe id data type.

Recipes_Ingredients

- recipe_id - Need to know which recipe has which ingredients. Needs to match recipe id data type.
- ingredient_id - Need to know which recipe has which ingredients. Needs to match ingredients id data type.

Users_Ingredients

- user_id - Need to know which user has which ingredients in their cart. Needs to match user id data type.
- ingredient_id - Need to know which user has which ingredients in their cart. Needs to match ingredients id data type.

SQL Statements

Create Tables

```
CREATE TABLE users (
   id SERIAL PRIMARY KEY NOT NULL UNIQUE,
   name VARCHAR(100) NOT NULL,
   email VARCHAR(50) NOT NULL UNIQUE,
   password VARCHAR(50) NOT NULL,
   address VARCHAR(100),
   phone INTEGER
);

CREATE TABLE recipes (
   id SERIAL PRIMARY KEY NOT NULL UNIQUE,
   created_by INTEGER NOT NULL REFERENCES users(id),
   name VARCHAR(100) NOT NULL,
   instructions VARCHAR(2000) NOT NULL,
   public BOOLEAN
);

CREATE TABLE occasions (
   id SERIAL PRIMARY KEY NOT NULL UNIQUE,
   user_id INTEGER NOT NULL REFERENCES users(id),
   name VARCHAR(100) NOT NULL UNIQUE,
```

```
    description VARCHAR(255)
);

CREATE TABLE ingredients (
    id SERIAL PRIMARY KEY NOT NULL UNIQUE,
    name VARCHAR(50) NOT NULL
);

CREATE TABLE users_ingredients (
    user_id INTEGER NOT NULL REFERENCES users(id),
    ingredient_id INTEGER NOT NULL REFERENCES ingredients(id)
);

CREATE TABLE recipes_ingredients (
    recipe_id INTEGER NOT NULL REFERENCES recipes(id),
    ingredient_id INTEGER NOT NULL REFERENCES ingredients(id)
);

CREATE TABLE occasions_recipes (
    occasion_id INTEGER NOT NULL REFERENCES occasions(id),
    recipe_id INTEGER NOT NULL REFERENCES recipes(id)
);
```

Insert Data

```
INSERT INTO users (name, email, password, address, phone)
VALUES
('Kyle Kocherhans', 'kylesEmail@gmail.com', 'kylesPassword', '123 Street', 1234567890),
('Mike Wazowski', 'mikesEmail@yahoo.com', 'mikesPassword', '456 Monster Ln', 0987654321);

INSERT INTO occasions (user_id, name, description)
VALUES (1, 'Thanksgiving', 'some description for Christmas'),
(1, 'Christmas', 'some description for Christmas'),
(2, 'Wedding', 'some description for wedding'),
(2, 'Monster Party', 'some description for monster party');

INSERT INTO recipes (created_by, name, instructions, public)
VALUES (1, 'Garlic Herb Butter Roast Turkey', 'put instructions here', true),
(1, 'The BEST Mashed Potatoes!', 'put instructions here', false),
(1, 'Minute Steak Stacks with Herbed Anchovy Butter', 'put instructions here', true),
(1, 'Harissa Sweet Potato Tarte Tatin', 'put instructions here', true),
(2, 'Southeast Asian Bites', 'put instructions here', true),
(2, 'Chicken Tacos', 'put instructions here', true),
(2, 'BBQ Chicken Wings', 'put instructions here', true),
```

```
(2, 'Secret Recipe Punch', 'put instructions here', false);

INSERT INTO occasions_recipes (occasion_id, recipe_id)
VALUES (1, 1),
(1, 2),
(2, 3),
(2, 4),
(3, 5),
(3, 6),
(3, 8),
(4, 7),
(4, 8);

INSERT INTO ingredients (name)
VALUES ('whole turkey'),
('garlic herb butter'),
('butter'),
('potatoes'),
('turkey gravy'),
('steak'),
('anchovy butter'),
('sweet potatoes'),
('harissa'),
('cumin'),
('red kimchi'),
('lo-dough'),
('lean tenderloin pork fillet'),
('chicken breast'),
('taco shells'),
('BBQ sauce'),
('chicken wings'),
('fruit punch mix'),
('Sprite');

INSERT INTO recipes_ingredients (recipe_id, ingredient_id)
VALUES (1, 1),
(1, 2),
(2, 4),
(2, 3),
(2, 5),
(3, 6),
(3, 7),
(4, 8),
(4, 9),
```

```
(4, 10),
(5, 17),
(5, 18),
(5, 19),
(6, 11),
(6, 12),
(7, 13),
(7, 14),
(8, 15),
(8, 16);

INSERT INTO users_ingredients (user_id, ingredient_id)
VALUES (1, 1),
(1, 2),
(1, 3),
(1, 4),
(1, 5),
(1, 6),
(1, 7),
(1, 8),
(1, 9),
(1, 10),
(2, 11),
(2, 12),
(2, 13),
(2, 14),
(2, 15),
(2, 16),
(2, 17),
(2, 18),
(2, 19),
(2, 3),
(1, 15);
```