Brainstorming

- Email
- Password
- username
- Ingredients
- Instructions
- Public or private (boolean)
- View other's recipes
- Grocery list
- occasions (and assign recipes to occasions)
- Image of finished recipe

Table Ideas

Users : will contain all of the info of a user on the app

- user_id
- username
- email
- password

Recipe: will contain the img of the food, will determine if recipe is private or public, and is linked to a specific user

- recipe_id
- user_id
- image
- public/private

Ingredient: lists what the ingredient is

- ingredient_id
- ingredient

Recipe_Ingredient: links together an ingredient with a recipe and how many of the ingredient

- recipe_ingredient
- ingredient_id
- recipe_id

- Quantity

Instruction: holds an instruction

- instruction_id
- instruction

Recipe_Instructions: links the instruction to the recipe

- recipe_instruction_id
- instruction_id
- recipe_id

Grocery_list: gathers the ingredients for a specific recipe for the user

- grocery_id
- recipe_ingredient_id
- user_id

Occasion: is an occasion made by a user, and has the name of the occasion

- occasion_id
- user_id
- name

Occcasion_recipe: links what recipes the user put for the occasion

- occasion_recipe_id
- recipe_id
- occasion_id

Relationships

- one-to-one
  - a.
- one-to-many
  - a. Recipe => User: as one user will have what recipes they share, along with the ingredients they used and the instructions they put
  - b. Ingredient => Recipe_ingredient => Recipe => Grocery_list: As the grocery list is for one user while grabbing multiple ingredients from a recipe
- many-to-many

a. Recipe => Occasion_recipe => Occasion: as however many recipes decided by the user will be saved to the specific occasion
b. Instruction => Recipe_instruction => Recipe: Multiple recipes will be saved into this table, and each recipe will have multiple instructions
c. Ingredient => Recipe_ingredient => Recipe: Will contain multiple recipes, and each recipe will have multiple ingredients and an amount of each ingredient

Columns

Users :

user_id is the primary key as it'll keep track of who the user is in the system, and it was selected to represent the id of the user

username will represent the name the user picked to represent themselves as to display to other users, and I chose varchar as usernames uses characters

email will serve as half of how the user logins, and is varchar as it'll involve characters

password uses characters so its datatype is varchar, and serves as half of the login information for users

Recipe:

recipe_id is the serial primary key, created for each unique recipe

user_id is linking the recipe to a specific user, so it is an integer as the serial primary key are always integers

image_url is text, as you never know how many characters will be in a url, and is involved just if the app is designed to show an image of the finished food from the recipe so other users can see what it would look like

public will be a boolean to indicate if the recipe is public or private, for the user to make their recipe open to other users or not

Ingredient:

ingredient_id is the serial primary key

ingredient will just be the name of the ingredient so datatype varchar will suffice

Recipe_Ingredient:

recipe_ingredient_id is the serial primary key

ingredient_id is the foreign key from the ingredient table to collect the correct ingredient

recipe_id is the foreign key from the recipe table to select the correct recipe the ingredient should be involved in

quantity is an integer, serving to indicate how many of the ingredient is used in the recipe

Instruction:

instruction_id is the primary key of the table

instruction will be an instruction step, so varchar is used as instructions will be text

Recipe_Instructions:

recipe_instruction_id is the primary key of the table

instruction_id is the foreign key for the instruction table, meant to get the instruction indicated by the id

recipe_id is the foreign key for the recipe table, selecting the correct recipe

Grocery_list:

grocery_id is the primary key of the table

recipe_ingredient_id is an integer as its the foreign key of the recipe_ingredient_table since that's where it'll get the ingredients in their correct quantities from the recipe

user_id is the foreign key from the users table, so the grocery list is for the correct user

Occasion:

occasion_id is the primary key of the table

name is the name of the occasion, and should not be too long so varchar should suffice

user_id is the foreign key from the users table, to link the occasion to the user

Occcasion_recipe:

occasion_recipe_id is the primary key of the table

recipe_id is the foreign key from the recipe table so the correct recipes are linked

occasion_id is the foreign key from the occasion table so the correct occasion is linked

SQL

```
CREATE TABLE users (
        user_id SERIAL PRIMARY KEY,
        username VARCHAR(100) NOT NULL,
        email VARCHAR(255) NOT NULL,
        password VARCHAR(255) NOT NULL
)
CREATE TABLE recipes (
        recipe_id SERIAL PRIMARY KEY,
        user_id INTEGER NOT NULL REFERENCES users(user_id),
        image_url TEXT,
        public BOOLEAN
)
CREATE TABLE ingredients (
        ingredient_id SERIAL PRIMARY KEY,
        ingredient VARCHAR(100)
)
CREATE TABLE recipe_ingredients (
        recipe_ingredient_id SERIAL PRIMARY KEY,
        ingredient_id INTEGER NOT NULL REFERENCES ingredients(ingredient_id),
        recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id)
)
CREATE TABLE instructions (
        instruction_id SERIAL PRIMARY KEY,
        instruction VARCHAR(255)
)
CREATE TABLE recipe_instructions (
        recipe_instruction_id SERIAL PRIMARY KEY,
        instruction_id INTEGER NOT NULL REFERENCES instructions(instruction_id),
        recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id)
)
CREATE TABLE grocery_list (
        grocery_id SERIAL PRIMARY KEY,
        recipe_ingredient_id INTEGER NOT NULL REFERENCES
        recipe_ingredients(recipe_ingredient_id)
        user_id INTEGER NOT NULL REFERENCES users(user_id)
```

```
)
CREATE TABLE occasions (
        occasion_id SERIAL PRIMARY KEY,
        user_id INTEGER NOT NULL REFERENCES users(user_id),
        name VARCHAR(100)
)
CREATE TABLE occasion_recipes (
        occasion_recipe_id SERIAL PRIMARY KEY,
        recipe_id INTEGER NOT NULL REFERENCES recipes(recipe_id),
        occasion_id INTEGER NOT NULL REFERENCES occasions(occasion_id)
)
```

Intermediate (INSERT INTO)

```
INSERT INTO users(username,email, password) VALUES ('sonic', 'a123@yahoo.com', '123');

INSERT INTO users(username,email, password) VALUES ('tails', 'tailsmp@yahoo.com', 'tmp5');

INSERT INTO users(username,email, password) VALUES ('shadow', 'shad@yahoo.com', 'hedgehog');
```