User:
1.user_id
2.email
3.password

This table will hold the id of the user using the app, alongwith their password and email.

Recipes:
1.recipe_id
2.user_id
3.ingredients
4.instructions
5.public BOOLEAN false

This table with be holding the recipes of each user, alongwith the ingredients and instructions, and whether or not it's publicly available for viewing.

Grocery List:
1.grocery_id
2.recipe_id
3.user_id
4.vegetables
5.meats
6.fruits
7.wheats
8.seasonings

The grocery list will contain the id's of the user and specific list, alongwith 5 possible ingredient categories that the user will need to create the recipe, alongwith the instructions for putting it all together. The user will also use delimiter separated values via commas to list multiple objects for each property.

Occasions:
1.occasion_id
2.recipe_id
3.date
4.location

The occasion list will have the id's of the occasion and recipe being used, alongwith the date and location it's being taken at.

Viewable:
1.viewable_id
2.recipe_id

3.public
The viewable table merely states whether or not a user is allowed to access a recipe or not.

```
CREATE TABLE user (
user_id SERIAL PRIMARY KEY,
email TEXT,
password TEXT
)

CREATE TABLE recipe (
recipe_id SERIAL PRIMARY KEY,
User_id INTEGER NOT NULL REFERENCES user(user_id),
Ingredients TEXT,
Instructions TEXT,
public BOOLEAN NOT NULL DEFAULT FALSE
)

CREATE TABLE groceryList (
grocery_id SERIAL PRIMARY KEY,
recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),
user_id INTEGER NOT NULL REFERENCES user(user_id),
)

CREATE TABLE occasions (
occasion_id SERIAL PRIMARY KEY,
recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id),
date DATE,
location TEXT
)

CREATE TABLE viewable (
viewable_id SERIAL PRIMARY KEY,
recipe_id INTEGER NOT NULL REFERENCES recipe(recipe_id)
)
```

Relationships:

One-To-One:
Grocery List => Recipes
The grocery list will subdivide what is found in recipes into various categories.

One-to-many:
Occasions=> Recipes

Users will be referencing the recipe they are going to use at the location and time that it will occur.

User=>Viewable
It will see which user is currently trying to access it and which recipes are it's own.
Viewable=>Recipes
It will see if the recipe is public, and if it's not check to see if it belongs to them or not, and if not so it will not grant permission to see it.

Many-to-many:
Recipes => Grocery List
The recipe will be sent to the grocery list, and any ingredients in those categories will be listed for the user to know exactly what to buy.
User => Recipes, Grocery List, Viewable, Occasions
What recipes and groceries the user will be able to access will be displayed.

Columns

User:
User id and it's serial because it's the primary key.
Email and password for logging in as text.

Recipes:
Recipe id and it's serial because it's the primary key.
User_id for the recipes they have created, and ingredients and instructions for how to make that recipe, and whether it's publicly available or not.

Grocery List:
Grocery id and it's serial because it's the primary key.
Recipe and user id are used to ascertain who recipe they have, if they should be able to view it, so that they may go grocery shopping with it.
And all the ingredients that are filed into sub-categories to make shopping easier.

Viewable:
The viewable id to categorize it in the table and serial as it's the primary key.
Recipe id, user id and public are all needed to ascertain what recipes are viewable for the user.

Occasions:
The occasion id to categorize it in the table and serial as it's the primary key.
Recipe id is needed to see what recipes are going to be at the occasion.
And the date and location are also listed as it is necessary in order ot schedule and organize the vent.