**Brainstorming**

**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

**Data Needed**

* User id
* User email
* User password
* Username
* Recipes created
* Ingredients
* Ingredients needed
* Public Recipes
* Private recipes
* Grocery list
* Occasions
* Occasion date
* Recipes for each occasion

Table Ideas

User Table:

* User\_id
* User\_email
* User\_password
* Username
* User\_recipies

Recipe table:

* Recipe name
* Recipe tag
* Ingredients
* Instructions
* Recipe\_public
* comment

Occasion table:

* Occasion
* Occasion\_date
* Recipies\_for\_occasion

Grocery\_list table:

* Needed\_ingredients
* Saved\_recipes

Comment table:

* Comment\_user
* Comment\_content

Ingredient table:

* Ingredient
* Ingredient\_name

Followers table:

* Followers\_id
* Followers\_username

Favorite table:

* Favorite\_id
* Favorite\_recipie

Relationships

One to one

One to many

* User to recipes
* User to comments
* User to grocery\_list
* User to occasions
* User to followers

Many to many

* Recipe to ingredient
* Ingredients to grocery\_list
* Recipes to ingredients

Diagram

Description automatically generated

Columns

User:

* User\_id: Serial (ID for user)
* User\_email: VARCHAR (the user’s Email. A character since it needs to be a sting)
* User\_password: VARCHAR (the users password. A varchar since it’s a sting)
* Username VARCHAR (the displayed name for the user. It’s a string so varchar)
* User\_phone: INT (the users phone number. It’s a number, so interger)

Recipe:

* recipe\_id SERIAL PRIMARY KEY, (the recipe’s id, so serial)
* recipe\_name VARCHAR(50), (the name of the recipe. A string, so varchar)
* recipe\_time TIMESTAMP, (The time that the user uploads the recipe. A time, so timestamp)
* recipe\_bio TEXT, (an intro for the recipe. A bunch of text, so text)
* recipe\_instructions TEXT, (the step by step instructions for the recipe. A lot of sentences, so text)
* recipe\_ingredients INT NOT NULL REFERENCES item(item\_id), (the ingredients needed for the recipe. Uses item table, so references)
* recipe\_author INT NOT NULL REFERENCES user(user\_id), (the user that uploaded the recipe. Uses the user table, so references)
* is\_private boolean (determines if it’s a private recipe or not. True or false, so Boolean)

buy\_list:

* buy\_list\_id SERIAL PRIMARY KEY, (id for the buy list, so serial)
* buy\_list\_name VARCHAR(50), (the name of the buy list. A string, so varchar)
* buy\_list\_time TIMESTAMP, (the time the list is uploaded. A time, so timestamp)
* buy\_list\_contents INT NOT NULL REFERENCES item(item\_id), (the items to be purchased. Uses the items table, so references)
* buy\_list\_user INT NOT NULL REFERENCES user(user\_id) (the user that uploaded the lsiat. Uses user table, so references)

occ:

* occ\_id SERIAL PRIMARY KEY, (id for occasions, so serial)
* occ\_name VARCHAR(20), (the name of the occasion. String, so varchar)
* occ\_bio TEXT, (a description of the occasion. Several sentences, so text)
* occ\_user INT NOT NULL REFERENCES user(user\_id), (the user that uploaded the occasion. References user, so references)
* occ\_recipes INT NOT NULL REFERENCES recipe(recipe\_id) (the recipe that is assigned to the occasion. Uses recipe table, so references)

follow:

* follow\_id SERIAL PRIMARY KEY, (id for followers, so serial)
* follower\_id INT NOT NULL REFERENCES user(user\_id), (the users following you. Uses users, so references)
* following\_id INT NOT NULL REFERENCES user(user\_id) (the users you are following. Uses users, so references)

favorites:

* fav\_id SERIAL PRIMARY KEY, (id for favs, so uses serial)
* fav\_recipe INT NOT NULL REFERENCES recipe(recipe\_id), (the recipe that is favorited. Uses recipe table, so references)
* fave\_user INT NOT NULL REFERENCES user(user\_id) (The user that saved the favorite. Uses users, so refrerences)

comment:

* comment\_id SERIAL PRIMARY KEY, (the id for a comment, so serial)
* comment\_author INT NOT NULL REFERENCES user(user\_id), (the author of the comment. Uses user table, so references)
* comment\_recipe INT NOT NULL REFERENCES recipe(recipe\_id), (the recipe commented on. Uses recipe table, so references)
* comment\_time TIMESTAMP, (the time the comment was uploaded. Uses time, so timestamp)
* comment\_content TEXT (the comment itself. Uses several sentences, so text)

item:

* item\_id SERIAL PRIMARY KEY, (the id for an ingrident, so serial)
* item\_name VARCHAR(20), (the name of the ingrident. A string, so varchar)
* item\_amount VARCHAR(20) (the amount of the ingrident. Could be in cups, grams, teaspoons, etc, so a varchar instead of an interger)

SQL Code

CREATE TABLE user (

user\_id SERIAL PRIMARY KEY,

user\_email VARCHAR(30),

user\_password VARCHAR(50),

username VARCHAR(30),

user\_phone INT

);

CREATE TABLE follow (

follow\_id SERIAL PRIMARY KEY,

follower\_id INT NOT NULL REFERENCES user(user\_id),

following\_id INT NOT NULL REFERENCES user(user\_id)

);

CREATE TABLE buy\_list (

buy\_list\_id SERIAL PRIMARY KEY,

buy\_list\_name VARCHAR(50),

buy\_list\_time TIMESTAMP,

buy\_list\_contents INT NOT NULL REFERENCES item(item\_id),

buy\_list\_user INT NOT NULL REFERENCES user(user\_id)

);

CREATE TABLE recipe (

recipe\_id SERIAL PRIMARY KEY,

recipe\_name VARCHAR(50),

recipe\_time TIMESTAMP,

recipe\_bio TEXT,

recipe\_instructions TEXT,

recipe\_ingredients INT NOT NULL REFERENCES item(item\_id),

recipe\_author INT NOT NULL REFERENCES user(user\_id),

is\_private boolean

);

CREATE TABLE comment (

comment\_id SERIAL PRIMARY KEY,

comment\_author INT NOT NULL REFERENCES user(user\_id),

comment\_recipe INT NOT NULL REFERENCES recipe(recipe\_id),

comment\_time TIMESTAMP,

comment\_content TEXT

);

CREATE TABLE favorites (

fav\_id SERIAL PRIMARY KEY,

fav\_recipe INT NOT NULL REFERENCES recipe(recipe\_id),

fave\_user INT NOT NULL REFERENCES user(user\_id)

);

CREATE TABLE occ (

occ\_id SERIAL PRIMARY KEY,

occ\_name VARCHAR(20),

occ\_bio TEXT,

occ\_user INT NOT NULL REFERENCES user(user\_id),

occ\_recipes INT NOT NULL REFERENCES recipe(recipe\_id)

);

CREATE TABLE item (

item\_id SERIAL PRIMARY KEY,

item\_name VARCHAR(20),

item\_amount VARCHAR(20)

);