**Brainstorming:**

* **Users need to be able to interact with other user’s recipes**
* **Recipes need to contain specific ingredients with ids that can be added to grocery lists**
* **Users need to be able to create occasions with lists of recipes for that occasion pulled from a recipe table**
* **Shopping list that holds the user id and the ingredient Id to keep track of ingredients**

**Tables:**

* Users – List of users in the database
  + Passwords
  + Username
  + Email
  + Id
  + Recipes
  + Shopping lists
  + Occasions
* Admins – list of admins
  + Admin name
  + Admin password
  + Admin email
  + Admin id
* Recipes- List of recipes that also lists the user that created the recipe and the ingredients inside the recipe
  + Recipe id
  + Recipe name
  + User id
  + Ingredient id
  + Instructions
* Ingredients- List of ingredients that can be used inside recipes and added to shopping lists
  + Ingredient id
  + Ingredient name
* Shopping lists- List of gathered ingredients by a specific user containing ids for the ingredients they wish to buy
  + Shopping list id
  + User id
  + Ingredients id
* Occasions- List of recipes created by one user and accessible to others for occasions that require multiple recipes
  + Occasion name
  + Occasion id
  + User id
  + Recipe id
* Reviews – List of reviews for recipes
  + Review id
  + User id
  + Review text
  + Recipe id
* Post – posts by user
  + Post\_id
  + User\_id
  + Public\_value
  + Recipe\_id
  + Occasion

**Relationships:**

* Users = One to Many: Only one part of users will be used in other tables
* Recipes = Many to Many: Many parts of recipes will be used in other tables
* Ingredients = One to many: Only one value of ingredients will be used in other tables
* Shopping List = One to One: Only one value in shopping lists will relate to only one other value in another table
* Occasions = One to one: One part of occasions will be used in one other table

**Columns:**

* Users
  + Passwords – Users passwords should be store for log in
  + User\_name – keep track of a user’s identity
  + Email – for log in and notifications
  + Id – to log each user as an individual in the database
  + Recipes
  + Shopping lists
  + Occasions
* Admins
* Recipes
* Ingredients
* Shopping Lists
* Occasions
* Reviews
* Posts

**SQL CODE:**

CREATE TABLE users (

user\_id SERIAL PRIMARY KEY,

user\_name TEXT,

user\_email TEXT,

user\_password TEXT,

recipe\_id INTEGER NOT NULL REFERENCES recipe(recipe\_id),

shopping\_list\_id INTEGER NOT NULL REFERENCES shopping\_list(shopping\_list\_id),

occasion\_id INTEGER NOT NULL REFERENCES occasions(occasion\_id)

)

CREATE TABLE admins(

admin\_id SERIAL PRIMARY KEY,

admin\_name TEXT,

admin\_password TEXT,

admin\_email TEXT,

user\_id INTEGER NOT NULL REFERENCES users(user\_id)

)

CREATE TABLE ingredients(

ingredients\_id SERIAL PRIMARY KEY,

ingredient\_name TEXT

)

CREATE TABLE recipes(

recipe\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(user\_id),

ingredient\_id INTEGER NOT NULL REFERENCES ingredients(ingredients\_id),

recipe\_name TEXT

)

CREATE TABLE shopping\_list(

shopping\_list\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(user\_id),

ingredients\_id INTEGER NOT NULL REFERENCES ingredients(ingredients\_id),

shopping\_list\_name TEXT

)

CREATE TABLE occasions(

occasion\_id SERIAL PRIMARY KEY,

occasion\_name TEXT,

user\_id INTEGER NOT NULL REFERENCES users(user\_id),

recipe\_id INTEGER NOT NULL REFERENCES recipes(recipe\_id)

)

CREATE TABLE posts(

post\_id SERIAL PRIMARY KEY,

occasion\_id INTGER NOT NULL REFERENCES occasions(occasion\_id),

comment TEXT REFERENCES reviews(comment),

recipe\_id INTEGER NOT NULL REFERENCES recipes(recipe\_id),

public\_value BOOLEAN,

user\_id INTEGER NOT NULL REFERENCES users(user\_id)

)

CREATE TABLE reviews(

review\_id SERIAL PRIMARY KEY,

user\_id INTEGER NOT NULL REFERENCES users(user\_id),

post\_id INTEGER NOT NULL REFERENCES posts(post\_id),

comment TEXT

)