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

Brainstorm:

User profile

* Email
* Password
* Recipes – Public
* Recipes - Private
* Occasions
* Grocery lists

Recipe pages

* ingredients
* occasions

Grocery lists

* ingredients

Occasions

* recipes

Tables / Relationships:

* User Sign In – this table will hold user email and password/ sign up data
  1. user\_id SERIAL PRIMARY KEY,
  2. first\_name VARCHAR,
  3. last\_name VARCHAR,
  4. email VARCHAR,
  5. password VARCHAR
* User Profile – this table will hold profile data
  1. profile\_id serial primary key,
  2. recipes\_ INT NOT NULL REFFERENCES
  3. grocery\_list\_id INT NOT NULL REFERENCES grocery\_list(users\_grocery\_list\_id)
  4. occasions\_id INT NOT NULL REFERENCES occasions(occasion\_id)
  5. user\_Id INT NOT NULL REFFERENCES profile(user\_id),
* Occasions – this table will hold list of user’s occasions
  1. Occasion\_id SERIAL PRIMARY KEY
  2. profile\_Id INT NOT NULL REFFERENCES profile(profile\_id),
  3. Recipes\_id NOT NULL REFERENCES,
* Grocery Lists – this table will hold user’s grocery lists
  1. grocery\_list\_id SERIAL PRIMARY KEY,
  2. profile\_Id serial primary key,
  3. Ingredient\_name
  4. Ingredients\_id INT NOT NULL REFFERENCES profile(ingredients\_id),

Recipes

* Recipes – this table will hold recipe data
  1. Recipe\_id SERIAL PRIMARY Key,
  2. Recipes\_id INT NOT NULL REFFERENCES (users\_recipes\_id)
  3. Ingredient varchar
  4. Ingredient\_id INT NOT NULL REFERENCES (ingredients\_id)
  5. Instructions varchar
  6. Profile\_id foreign key to profile
  7. Public? Boolean true or false; to set private recipes
* Ingredients – this table will hold all the ingredients
  1. Ingredients\_id SERIAL PRIMARY KEY
  2. Ingredient\_name

“one-to-one”: user-sign-in to user profile; recipes to ingredients

* All of these have one relationship to one record in another table

“one-to-many”: profile-id -> (user recipes, user occasions, user grocery lists); ingredients-id -> (user grocery lists, recipe private, recipes

* On the flip side, the profile id and ingredients id are a foreign key for many tables

“many-to-many”:; user grocery list if my middle table, it connects to both user profile and ingredients

Columns:

User Sign in Table

1. User\_id: integer, primary key
2. First\_name: VARCHAR because first name is a string of characters; to customize profile
3. Last\_name: VARCHAR because last name is a string of characters; to customize profile
4. Email: VARCHAR because email is a string of characters; to associate account , unique values only
5. Password: VARCHAR because password is a string of characters; to protect the password

User Profile Table

1. Profile\_id: primary key, integer
2. recipe\_id: integer, to associate with list of user’s public recipes, foreign key
3. grocery\_id: integer, to associate with user’s grocery lists, foreign key
4. Occasions\_id: integer, to associate with user’s occasions, foreign key
5. user\_id: integer, foreign key, to associate with user

Recipes

1. recipe\_id: integer, primary key
2. Ingredient\_name: varchar, name of ingredient
3. Ingredient\_id: foreign key int to ingredients table
4. Instructions: text; how to make recipe
5. Profile\_id: integer, foreign key, to link back to profile
6. Public? Boolean, to set public or private

Occasions Table

1. Occasion\_id: primary key, integer
2. Profile\_id: foreign key, to link back to user profile, integer
3. Recipes\_id: foreign key, link to recipe

Ingredients Table

1. Ingredients\_id: integer, primary key
2. Ingredient\_name: varchar because name of ingredient