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
    A. user_id SERIAL PRIMARY KEY,
    B. first_name VARCHAR,
    C. last_name VARCHAR,
    D. email VARCHAR,
    E. password VARCHAR
- User Profile – this table will hold profile data
    A. profile_id serial primary key,
    B. recipes_ INT NOT NULL REFFERENCES
    C. grocery_list_id INT NOT NULL REFERENCES grocery_list(users_grocery_list_id)
    D. occasions_id INT NOT NULL REFERENCES occasions(occasion_id)
    E. user_Id INT NOT NULL REFFERENCES profile(user_id),
- Occasions – this table will hold list of user's occasions
    A. Occasion_id SERIAL PRIMARY KEY
    B. profile_Id INT NOT NULL REFFERENCES profile(profile_id),
    C. Recipes_id NOT NULL REFERENCES,
- Grocery Lists – this table will hold user's grocery lists
    A. grocery_list_id SERIAL PRIMARY KEY,
    B. profile_Id serial primary key,

C. Ingredient_name
D. Ingredients_id INT NOT NULL REFFERENCES profile(ingredients_id),

Recipes

- Recipes – this table will hold recipe data
    A. Recipe_id SERIAL PRIMARY Key,
    B. Recipes_id INT NOT NULL REFFERENCES (users_recipes_id)
    C. Ingredient varchar
    D. Ingredient_id INT NOT NULL REFERENCES (ingredients_id)
    E. Instructions varchar
    F. Profile_id foreign key to profile
    G. Public? Boolean true or false; to set private recipes
- Ingredients – this table will hold all the ingredients
    A. Ingredients_id SERIAL PRIMARY KEY
    B. 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