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

Brainstorming

- Users can view other users recipes
    - Who is following whom

- Users can create occasions
    - Post text
    - Image url
    - Post author
    - Assign recipes to occasion
    - Date

- Users can sign into the app
    - Email
    - Password

- Users can create recipes (ingredients & instructions)
    - Post text
    - Image url
    - Post author
    - Date
    - Recipe can be marked public or private

- Users can create grocery lists
    - Post text
    - Date
    - Post author
    - Ingredients from recipes can be added to user grocery list

- Users can review recipes
    - Post text
    - Date
    - Post author
    - Image url
    - rating

Table Ideas

- Users
  - Holds user personal information, name, date, age
- Authorization
  - Holds login details
    - Email
    - Password
- Recipe posts
  - Holds information about recipes and who created it (text, image, date, author)
- Occasion posts
  - Holds information about occasions and who created them (text, image, date, author)
- Review posts
  - Holds information about who wrote the review, which recipe the review is about, image of their recipe, date of review, text of review, rating
- Grocery List
  - Holds information about the recipe ingredients (text, date, author)
- Followers
  - Holds information about who is following whom
    - Who is being followed
    - Who is the follower

Relationships

One-to-One

- User to Authorization-
  - there an only be one unique email per user

One-to-Many

- Users to occasion posts-
  - one user could create multiple occasions but and occasion can only be tied to one user
- Users to review posts-
  - one user can create multiple reviews but the review is only tied to one user
- Users to recipe posts-
  - one user can create multiple recipe posts but the recipe posts are only linked to the one posting user

- **Many-to-Many**
- User to grocery lists-
    - users can create multiple grocery lists and access recipe ingredients created by other users
- Users to followers-
    - one user can follow multiple users and other users can follow multiple others

Columns

- **Users -**
    - User_id, integer unique to each user
    - User_name, varchar, unique to each user so duplicates won't be possible for reviews, etc.
    - Age, integer because it's a number
    - Location, varchar, to let users see where other users are located
- Authorization
    - Auth_id,integer unique to each user
    - Email, varchar because its a string of text information
    - Password, varchar could be a string of numbers, text, or characters
- Recipe
    - Recipe_id,integer unique to each post
    - Post_text, varchar, sentences of steps for the recipe
    - Img_url - varchar, needs the link to the image url
    - Date - timestamp, so that the users can see when the recipe was posted
    - User_name - varchar, unique to each user so duplicates won't be possible for reviews, etc.
    -
- Occasion
    - Occasion_id- integer unique to each post
    - Post_text- varchar, sentences of steps for the unique occasion
    - Img_url- varchar, needs the link to the image url
    - Date- date, so the user can set the date of the occasion
    - User_name- - varchar, unique to each user so duplicates won't be possible for reviews, etc.
- Review
    - Review_id,integer unique to each post
    - Post_text - varchar, will contain the review and comments of the followers
    - Img_url- varchar, needs the link to the image url
    - Date - timestamp, so that the date the review was posted will appear
    - User_name - varchar, unique to each user so duplicates won't be possible for reviews, etc.
    - Rating- varchar, will contain the review and rating of the user
- Grocery List
    - Grocery_id- integer unique to each post

        - Date- date so the user can set the date of the grocery list
        - Post_text- varchar, list of ingredients needed
        - User_name - varchar, unique to each user so duplicates won't be possible for reviews, etc.
- Followers
    - Followers_id- integer unique to each post
    - User_name - varchar, unique to each user so duplicates won't be possible for reviews, etc.
    - Follower_id - varchar, unique to each user, so that the main user (user_name) can follow different followers and vice versa

```
CREATE TABLE user(
  user_id serial primary key,
  user_name varchar(60),
  age integer,
  location varchar(100)

 );

CREATE TABLE authorization(
  auth_id serial primary key,
  email varchar(255),
  password varchar(12)

  );

CREATE TABLE occasion(
  occasion_id serial primary key,
  post_text varchar(15000),
  img_url varchar(15000),
  occasion_date date,
  user_name varchar(60)

  );

CREATE TABLE recipe (
  recipe_id serial primary key,
  post_text varchar(150000),
  img_url varchar(100000),
  recipe_date timestamp,
  user_name varchar(60)

  );
```

```
CREATE TABLE grocery (
  grocery_id serial primary key,
  post_text varchar(10000),
  grocery_date date,
  user_name varchar(60)

  );

 CREATE TABLE review (
          review_id serial primary key,
  post_text varchar(500),
  img_url varchar(100000),
  review_table timestamp,
  user_name varchar(60),
  rating varchar(5)

  );

CREATE TABLE followers (
  followers_id serial primary key,
  user_name varchar(60),
  follower_id varchar(60)

          );



-- INSERT INTO account (user_name, age, location)
-- VALUES ('Sully', 8, 'Mesa');
-- INSERT INTO account c
-- VALUES ('Ku', 3, 'Wahiawa');
-- INSERT INTO account(user_name, age, location)
-- VALUES ('Sullivan', 7, 'Gilbert');


select * from account
```