Brainstorming

- ~~Users~~
- ~~Grocery lists~~
- ~~Sign-in (email/password)~~
- ~~posts~~
- ~~Public or private posts~~
- vegetarian/non-vegetarian
- ~~Recipes(ingredients/instructions)~~
- ~~Photos~~
- ~~Time-stamps~~
- ~~UpDoots~~

Table Ideas

- Users
  - User ID
  - Email (I figured the Users table will contain all the sign in info here.)
  - Password
  - Occasion
  - Personal Grocery List

- Public Posts
  - User ID
  - Photo
  - Post ID

- UpDoots
  - UserID
  - Post_ID

- Private Posts
  - User ID (after writing it all out i decided the user can just include if the dish
  - Photo is vegetarian or not in the post. It was very repetitive.)
  - Post ID (I decided it was best to combine private and public tables into one)

- Timestamps
  - Date and Time of posts (decided timestamps and recipes belong in the posts table)
- Recipes
  - Ingredients
  - Recommended grocery list

- Relationships:
    - One-to-One:
        - Recipes (Recipes will only be one per post)
        - Timestamps (The Timestamps will only be one per post.)
    - One-to-Many:
        - UpDoots (UpDoots will store Many UserIDs)
    - Many-to-Many:
        - User ID (Many User IDs that are used in Many other Tables)
        - Public/Private Posts (Posts will have Many relationships with other tables)

Hopefully I did this relationship thing correctly.. I'm a little confused on breaking them down into the different categories.

CREATE TABLE users (id SERIAL PRIMARY KEY, name VARCHAR(50), email VARCHAR(50), password VARCHAR(50), occasion VARCHAR(50), grocery_list VARCHAR(100));

INSERT INTO users (name, email, password, occasion, grocery_list)
VALUES ('tevin', 'tpulley@gmail.com', '1234' , 'ThanksGiving' , 'Turkey')

CREATE TABLE posts (id INT PRIMARY KEY, photos VARCHAR(500), post_id SERIAL, timestamps TIMESTAMP WITH TIME ZONE, ingredients VARCHAR(50), private_posts VARCHAR(500), public_posts VARCHAR(50));

CREATE TABLE updoots (user_id INT PRIMARY KEY, post_id INT);