**Brainstorming**

user login information

* Name
* Email
* Username/ password

Search functionality

* Search by cuisines
* Search by food category (pizza, burger, etc.)
* Search by diet type ( Vegetarian, non vegetarian, low-carb, low-fat, etc)
* Search food by occasions

Access to recipes

* Access to edit or add their own recipes/grocery items
* Favorite their recipes
* Access to ingredients
* Add ingredients to grocery items list

Access to grocery list from other users

* Select measurement units
* Measurement quantity
* Ingredients
* Allows the user to select quantity of ingredients

Ability to set the recipes as private or public

* Review
* Comment on recipe
* Rate recipes

Create and assign occasions to recipes

**Tables:**

* User
  + This table will have information on user personal information/login information
  + First name
  + Last name
  + User id information
* Recipes
  + This table will hold information about the recipes selected
* Ingredients
  + This table will hold information about the ingredients required
  + This will hold ingredients\_id and ingredients\_name
* Measurement units
  + This table will hold information about measurement description
* Measurement qty
  + This table will hold information about quantity description
* Instruction table
* Grocery list
* Comment
* Occasions
* Cuisines
* Food tag
* Food type

**RELATIONSHIPS**

* I am planning a mobile recipe sharing app. In the many-to-many section, I could have occasions table and recipe table because there can be several recipes for several occasions.
* User to occasions can be one to many because each individual user can have access to multiple occasion
* User to grocery list is one to one where a user can create one grocery list
* Instruction table will hold one to one relationship with recipes table

Addition table

* Recipe ingredients
  + Recipe ingredients would be a join table for recipes table to measurement quantity, measurement unit and ingredients tables
  + For example: recipes table will hold recipe\_id, recipe\_name and recipe\_description, measurement units table will hold measurement id and measurement description, measurement quantity will hold measure qty id and qty amount and ingredients will hold ingriendients id and ingredients name whereas recipe ingredients table will hold the foreign keys for the previous mentioned tables.

SQL SYNTAX

CREATE TABLE user\_ data(

user\_id SERIAL PRIMARY KEY,

user\_name VARCHAR(50),

password VARCHAR(50)

);

CREATE TABLE search (

search\_id INT NOT NULL SERIAL PRIMARY KEY,

searchlist VARCHAR(50) NOT NULL);

CREATE TABLE food\_category (

foodcategory\_id NOT NULL SERIAL PRIMARY KEY,

food\_category\_name VARCHAR (50) NOT NULL,

search\_id INT);

CREATE TABLE diet\_type (

diet\_type\_id SERIAL PRIMARY KEY,

diet\_name VARCHAR (50),

diet\_description VARCHAR (50),

search\_id INT);

CREATE TABLE cuisines (

cuisine\_id SERIAL PRIMARY KEY,

cuisine\_name VARCHAR (50),

search\_id INT);

CREATE TABLE recipes (

recipe\_id SERIAL PRIMARY KEY,

title VARCHAR (50),

description VARCHAR(50),

cuisine\_id INT,

occasion\_id INT,

diet\_type\_id INT,

food\_category\_id INT);

CREATE TABLE recipe\_ingredients (

recipeingredients\_id SERIAL PRIMARY KEY,

recipe\_id INT,

measurement\_id INT,

ingredient\_id INT,

measurementqty\_id INT);

CREATE TABLE grocery\_list (

grocery\_list\_id SERIAL PRIMARY KEY,

recipe\_id INT,

measurement\_id INT,

ingredient\_id INT,

measurementqty\_id INT,

occasion\_id INT);

CREATE TABLE occasion (

occasions\_id SERIAL PRIMARY KEY,

occasions VARCHAR (50));

CREATE TABLE ingredients (

ingredient\_id SERIAL PRIMARY KEY,

ingredient\_name VARCHAR (50),

grocery\_list\_id INT);

CREATE TABLE measurement (

measurement\_id SERIAL PRIMARY KEY,

measurement\_description VARCHAR (50) );

CREATE TABLE measurement\_quantity (

measurementqty\_id SERIAL PRIMARY KEY,

measurementqty\_description VARCHAR (50) );

INSERT INTO user\_data (user\_name, password)

VALUES ('zenithshrestha', 'fdbfuefe43'), ('angelinajolie', 'kbf3i4432'), ('devmountain', 'nfdcds');

SELECT \* FROM user\_data;