Data Modeling
Part 1:
Step 1:

Brainstorming

- Users
  1. User ID
  2. Username
  3. Password
  4. Email Address
- Recipes
  1. Recipe ID
  2. Public or private
  3. Ingredients
  4. Text/picture
- Grocery List
  1. Links to recipes
  2. Adds ingredients
  3.
- Occasions
  1. Occasions ID
  2. Links to recipes
  3. Links to users
- Ingredients
  1. Ingredient ID
  2. Links to recipes
  3. Links to grocery list

Step 2:
Table Ideas

- Users
  This table will hold all of the user data, profile picture, username, Email, user ID, Recipes
- Recipes
  This will hold relevant ingredients, cooking instructions, recipe ID, public or private, food picture
- Grocery List
  This table will link ingredients and recipes
- Ingredients
  Ingredient ID, stores ingredients for recipes and grocery list
- Occasions
  Contains recipes suited for different events or celebrations
- Reviews
  Allows users to rate and comment on recipes

Step 3:
<u>Relationships</u>

One to one:
User to grocery list(one user only needs one grocery list)

One to many:
Users to Recipes, (one user can access many recipes)
Users to Occasions, (one user can create recipes for many different occasions)
recipes to reviews, (one recipe can have many reviews)
Recipe to ingredients (one recipe has many ingredients)

Many to many:
Recipes to occasions (one recipe can have several occasions, and an occasion can have many recipes)

<u>Columns</u>

- Users
    1. User ID (serial Primary Key, Integer)
    2. Username (varchar(255))
    3. Password (varchar(255))
    4. Email Address (varchar(255))
    5. Recipes (Foreign Key, Integer)
- Recipes
    1. Recipe ID (serial Primary Key, Integer)
    2. Public or private (Boolean)
    3. Ingredients (Foreign Key, Integer)
    4. instructions/picture (text)
    5. Occasions (Foreign Key, Integer)
- Grocery List
    1. List_id (serial Primary Key, Integer)
    2. Recipes (Foreign Key, Integer)
- Occasions
    1. Occasions ID (serial Primary Key, Integer)
    1. Recipes (Foreign Key, Integer)
    2. Users (Foreign Key, Integer)
- Ingredients
    1. Ingredient ID (serial Primary Key, Integer)
    2. Ingredients_name (varchar(255))
- Reviews
    1. Review_id (serial Primary Key, Integer)
    2. Users (Foreign Key, Integer)
    3. Review (text)

SQL create tables:

```
create table users (
  User_id serial primary key,
  Username varchar(255),
  Password varchar(255),
  email varchar(255),
  recipes_id int not null references recipes(recipes_id)
);

create table recipes (
  recipe_id serial primary key,
  Recipe_private boolean default true,
  ingredients_id int not null references Ingredients(ingredients_id),
  occasions_id int not null references Occasions(occasions_id),
  instructions text,
  picture text
);

create table grocery_list (
  grocery_list_id serial primary key,
  recipes_id int not null references recipes(recipes_id)
);

create table occasions (
  occasions_id serial primary key,
  recipes_id int not null references recipes(recipes_id),
  users_id int not null references users(users_id)
);

create table ingredients (
  ingredients_id serial primary key,
  name varchar(20)
);

create table reviews (
  review_id serial primary key,
  users_id int not null references users(users_id),
  user_review text,
  user_review_pic text
);
```

You have to alter each table to insert the foreign key after all the tables are made to create the relationships.