**FEATURES:**

1. Users can sign into the app with their email and password
2. Users can create recipes with ingredients and instructions
3. Recipes can be marked as public or private
4. Users can view other user’s recipes
5. Ingredients from recipes can be added to users grocery lists
6. Users can create their own occasions and assign recipes to occasions

**BRAINSTORMING:**

1. Username
2. Password
3. Recipe
4. Grocery lists
5. Occasions
6. Public/private

**TABLE IDEAS:**

1. Users: this table will hold usernames and passwords, each row will be a user\_id with username and password
2. Recipe: this table will hold recipes, each row will be a list of ingredients and instructions
3. Grocery lists: this table will hold ingredients users have chosen, each row will be the ingredients assigned to a users grocery list
4. Occasions: this table will be special occasions users have created, each row will be an occasion with recipes related to it
5. Public/private: this table will hold whether a recipe is public, each row will be a recipe

**RELATIONSHIPS:**

1. One-one:

* Recipes table and public/private table because each recipe is either private or public

1. One-many:
2. Many-many:

* Users table and recipes table because many users can create and view recipes and many recipes can belong to many users

**COLUMNS:**

1. Users:

user\_id, integer

username, varchar 25

password, text

1. Recipes:

recipe\_id, integer

user\_id, integer

name, varchar 100

ingredients, varchar 255

instructions, varchar 500

1. GroceryLists:

grocery\_list\_id, integer

user\_id, integer

ingredients, varchar 225

1. Occasions:

occasions\_id, integer

user\_id, integer

name, varchar 100

ingredients, varchar 225

1. Private:

private\_id, integer

recipe\_id, integer

1. Public:

public\_id, integer

recipe\_id, integer