

## Brainstorming

- **Things to keep track of**

- User
  - Email
  - Name
  - Password
- Recipe
  - Ingredients
  - Prep time
  - Cooking time
  - Image
  - Public or Private
  - Steps to follow
- Grocery list
  - Ingredients
  - User\_id
- Occasions
  - Recipes

- **Tables**

- Users
  - Email
  - Name
  - Password
- Recipes
  - Name
  - user\_id
  - Ingredients
    - Using association table RecipeIngredients
  - Image
  - Private (one to many)
  - Public (many to many)
- Steps
- Grocery list (user to many lists)
  - Ingredients
  - user\_id
- Ingredients (many ingredients to many grocery lists)
  - Name
  - Description
- Occasions
  - Name
  - Date/Time of Year
  - Associated recipes

- RecipeIngredients
  - Association table
- UserOccasions
  - Association table
- GrocerylistIngredients
  - Association table
- RecipeOccassions
  - Association table
- **Relationships**
  - One-to-one
    - None
  - One-to-many
    - User to grocery lists ( a user may want multiple lists for multiple transactions or stores )
    - User to recipes ( a user may need to store multiple recipes on their account )
    - Recipe to steps (One recipe has many steps but each step is associated with only that unique recipe)
  - Many-to-many
    - Recipes to ingredients (many ingredients to a recipe, many recipes w/ the same ingredients)
    - Users to occasions (many users may want to keep track of the same occasion(s))
    - Grocery list to ingredients (same as recipes to ingredients)
    - Recipe to occasion (many recipes for many occasions)
- **Columns**
  - User
    - user\_id SERIAL PRIMARY KEY
    - email VARCHAR(60)
      - For login
      - emails should be 60 chars or less. Longer may be a bot.
    - password VARCHAR(256)
      - For login
      - hashed password may follow some (more secure) flavor of SHA256
    - name VARCHAR(60)
      - So we can refer to a user by their name
        - It would probably be good to split this up into first\_name, last\_name in practice
      - Names should not surpass 60 chars
  - UserOccasion
    - user\_occassion\_id SERIAL PRIMARY KEY
    - user\_id REFERENCES user

- creates one to many relationship between UserOccasion and User
  - occasion\_id REFERENCES occasion
    - creates one to many relationship between UserOccasion and occasion
- GroceryList
  - grocerylist\_id SERIAL PRIMARY KEY
  - name VARCHAR(40)
    - Allows a user to easily find a specific list from their set of lists
    - Names need not be overly long. 40chars should suffice
  - user\_id REFERENCES user
    - Attaches a list to a given user
- Recipe
  - recipe\_id SERIAL PRIMARY KEY
  - user\_id REFERENCES user
    - Defines ownership of the recipe
  - name VARCHAR(60)
    - Allows recipe to have a name for users to find quickly
    - Name need not be overly long. 60chars should suffice.
  - description TEXT
    - Allows user to describe their recipe, making it more appealing to other users
    - Description may be fairly long (up to a couple thousand chars)
  - imageURL TEXT
    - Allows user to upload a picture
    - Store image resource and refer to it using URL, which could be quite long
      - Is there a better way?
  - private BOOL
    - Allows user to mark recipe as public or private
    - Default is true
    - If true, recipe is private. If false, recipe is public.
- RecipeOccasion
  - recipe\_occasion\_id SERIAL PRIMARY KEY
  - recipe\_id REFERENCES recipe
    - Establish many to one relationship between RecipeOccasion and Recipe
  - occasion\_id REFERENCES occasion
    - Establish many to one relationship between RecipeOccasion and Occasion
- Occasion
  - occasion\_id SERIAL PRIMARY KEY
  - name VARCHAR(60)

- Allows occasion to have a name
    - In this app, we assume public occasions only, for simplicity. Another column could be added to make it private and the association table UserOccasion could be used to subquery occasions for private use.
    - Alternatively, occasions may function as tags on the UI, and be somewhat generic so they don't have to be stored multiple times (eg. birthday, Thanksgiving, Christmas, etc. would be applicable to most users). This would minimize data storage requirements while still ensuring the end user functionality that I presume occasion will enable (that is, allowing recipe planning for a specific occasion)
- GrocerylistIngredient
  - grocerylist\_ingredient\_id SERIAL PRIMARY KEY
  - grocerylist\_id REFERENCES grocerylist
    - Establish one to many relationship between grocerylistIngredient and groceryList
  - ingredient\_id REFERENCES ingredient
    - Establish one to many relationship between grocerylistIngredient and ingredient
- RecipeIngredient
  - recipe\_ingredient\_id SERIAL PRIMARY KEY
  - recipe\_id REFERENCES recipe
    - Establish one to many relationship between recipeIngredient and recipe
  - ingredient\_id REFERENCES ingredient
    - Establish one to many relationship between recipeIngredient and ingredient
- Step
  - step\_id SERIAL PRIMARY KEY
  - recipe\_id REFERENCES recipe
    - Establishes one to many relationship between recipe and step
  - step\_number INT
    - Associates the step with a certain order so all steps belonging to a recipe may be sorted accordingly
    - Steps are part of an ordered list, and could have many steps for a complex recipe, so INT makes the most sense here
  - description TEXT
    - This is the instructions specific to the step
    - Steps could be quite complicated, and length of description is unknown so TEXT is best
- Ingredients
  - ingredient\_id SERIAL PRIMARY KEY

- name VARCHAR(60)
  - All ingredients have names
  - Names should not exceed 60 chars
- description
  - Allows people who may be browsing the ingredients associated with public recipes to understand ingredients they are unfamiliar with
  - Description, although presumed short and optional, is of unknown length so TEXT is best
- imageURL
  - Allows the ingredient to have a picture attached
  - Resource may have a long URL so text is best