Grant Roszhart

Meal Master

Re-Design Thoughts

I have decided not to make any changes to my database selection or tables that I've created. I received good feedback and am happy where everything is at this time. Obviously, this is still not set in stone, as removals or additions may be necessary down the road.

(Please see bottom of this document for Stretch Goal idea).

Database Style

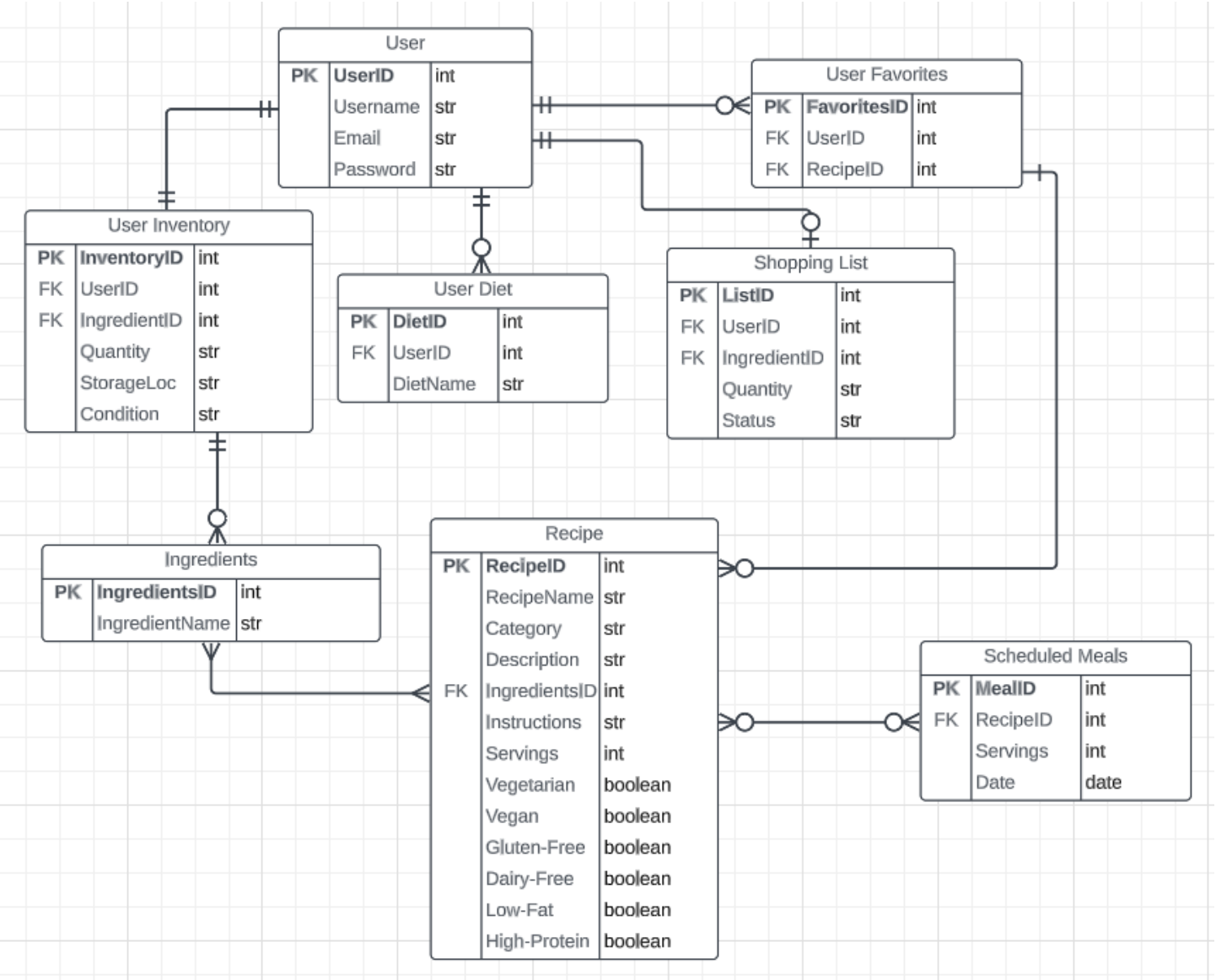
For this project, I have elected to use a relational database. A relational database is the right choice for the scope of my project for a few reasons. First, this app will use structured data like user info, recipes, ingredients, etc., and relational databases excel at handling and manipulating structured data. I also selected relational over other types for the data integrity aspect that comes with it. The usage of constraints, primary and foreign keys, and transactions will guarantee the data remains consistent when being pulled around to different locations. The reliability this offers is difficult to justify passing up for another database type. I will require the use of complex queries in order for my users to sort and filter through recipes, which again is a major benefit of a relational database style. I have also considered the idea of scaling this application to a larger project, and relational databases are great at handling large datasets with high levels of user traffic.

Program Selection

I will be using PostgreSQL as my relational database application. PostgreSQL offers many perks to its users. I have found it easy to use, and with pgAdmin, I can easily make changes to all levels of my database. Like I mentioned in the previous section, data integrity is crucial when working with large sets of data to ensure consistency and reliability. PostgreSQL makes it easy to view data and manipulate accordingly. This service is also open source, and there are dozens of resources online for referencing and expanding my database when necessary. PostgreSQL follows the principles of ACID (Atomicity, Consistency, Isolation, Durability) which is a quality standard that ensures data consistency, which this program will need. Users will be relying heavily on their inventory data being stored and manipulated consistently. This software also shines in its security, requiring authentication and access controls, as well as passwords to access any database.

Grant Roszhart

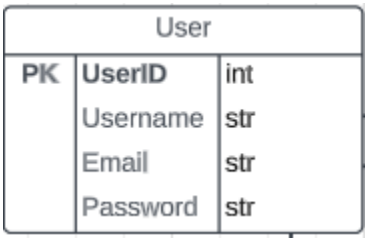
Entity Relationship Diagram



Grant Roszhart

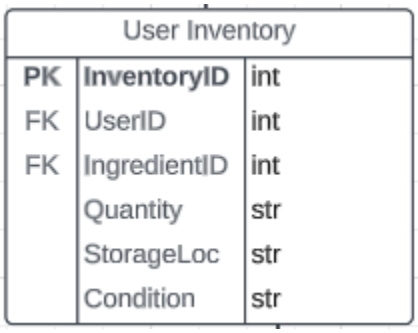
These are the tables required for my MVP. They include all of the necessary relationships and data for the application to function. In this diagram, you can see the relationship types (one-to-one, etc.) as well as each attribute and type for each table. Now let's dive into each table and see what they're all about.

User Table



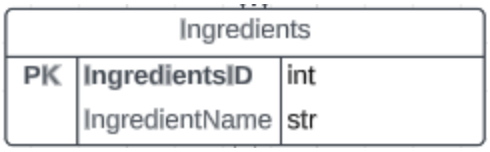
The User table is critical for the function of my application. It will contain the information necessary for users to log into their accounts with the username, email, and password attributes. The UserID is the primary key which is referenced by many other tables in this database.

User Inventory Table



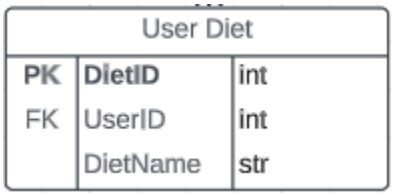
The User Inventory table will keep track of the user's ingredients they have in their inventory. This table references the User table and the Ingredients table with foreign keys. It will store information on the quantity of the user's ingredient, storage location (fridge, pantry, freezer, etc.), and condition (canned, fresh, frozen, etc.).

Ingredients Table



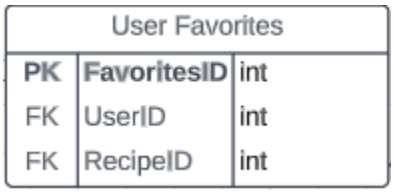
The Ingredients table is a small one, but is responsible for relating the ingredient ID to its name. This is important for users to be able to input ingredients as their names and the database will convert that to an ID to use.

User Diet Table



The User Diet table is crucial for the user to associate a diet with their account so that the app can recommend appropriate recipes.

User Favorites Table



The User Favorites table will keep track of the recipes that the user has favorited. This will help them easily find their favorite recipes from the past if they want to make them again. The table references the User table and Recipe table, as it ties the recipe to the User.

Recipe Table

Grant Roszhart

Recipe		
PK	**RecipeID**	int
	RecipeName	str
	Category	str
	Description	str
FK	IngredientsID	int
	Instructions	str
	Servings	int
	Vegetarian	boolean
	Vegan	boolean
	Gluten-Free	boolean
	Dairy-Free	boolean
	Low-Fat	boolean
	High-Protein	boolean

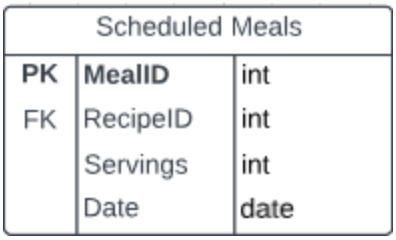
The Recipe table is one of the most important tables in the database. It will keep track of each recipe's data. The category attribute will refer to the recipe's "genre" (Asian, Seafood, American, etc.) and can have multiple. The Instructions attribute will contain the list of directions in order to create the recipe. Servings will be an integer that counts the number of people that the recipe will feed. The next 6 attributes are boolean, true or false, for different dietary restrictions. Making these booleans will help users filter recipes by these options.

Shopping List Table

Shopping List		
PK	**ListID**	int
FK	UserID	int
FK	IngredientID	int
	Quantity	str
	Status	str

The Shopping List table will keep track of the ingredients that the user needs to acquire based on the recipes they select. It will contain the quantity they need to get from the store, as well as the status (which I may change to "purchased" and make the type boolean) which indicates whether the user has obtained the ingredient.

Scheduled Meals Table



The Scheduled Meals table will store the recipe information that the user has stored when adding a recipe. It contains the RecipeID, which links to the Recipe table in order to reference all ingredients and related information, and will also show the date that the user intends to cook that recipe.

Stretch Goal: Upload Recipes

A stretch goal I have for this project would be for users to be able to upload their own recipes that they enjoy or have found useful. This would allow the app to grow organically, and also open up the opportunity for a more social community.

As far as additional tables are concerned, there would not need to be another table added. Rather, the Recipe table would need an additional column “UserID” to signify if a user uploaded this Recipe, and which user it was.