HoMePIT Updates

By Erik Anderson, Jerome Chestnut, and Maxwell Fugette

Outline

- Introduction
- Current Changes
  - App changes
  - Database changes
- Considered Changes
  - App changes
  - Database changes
- Accomplishments
- What needs implementation
- Issues

Current Changes, App

- Early on, it was decided to switch from Replit to VSCode as our primary IDE due to familiarity and freedom of development.
  - Using the Next.js template with Replit forces the user to use TypeScript, which we did not want.
  - Replit seemed to complicate things such as version control, user authentication, and using Supabase.
- Replit will still be used for hosting purposes, but only for the demo.

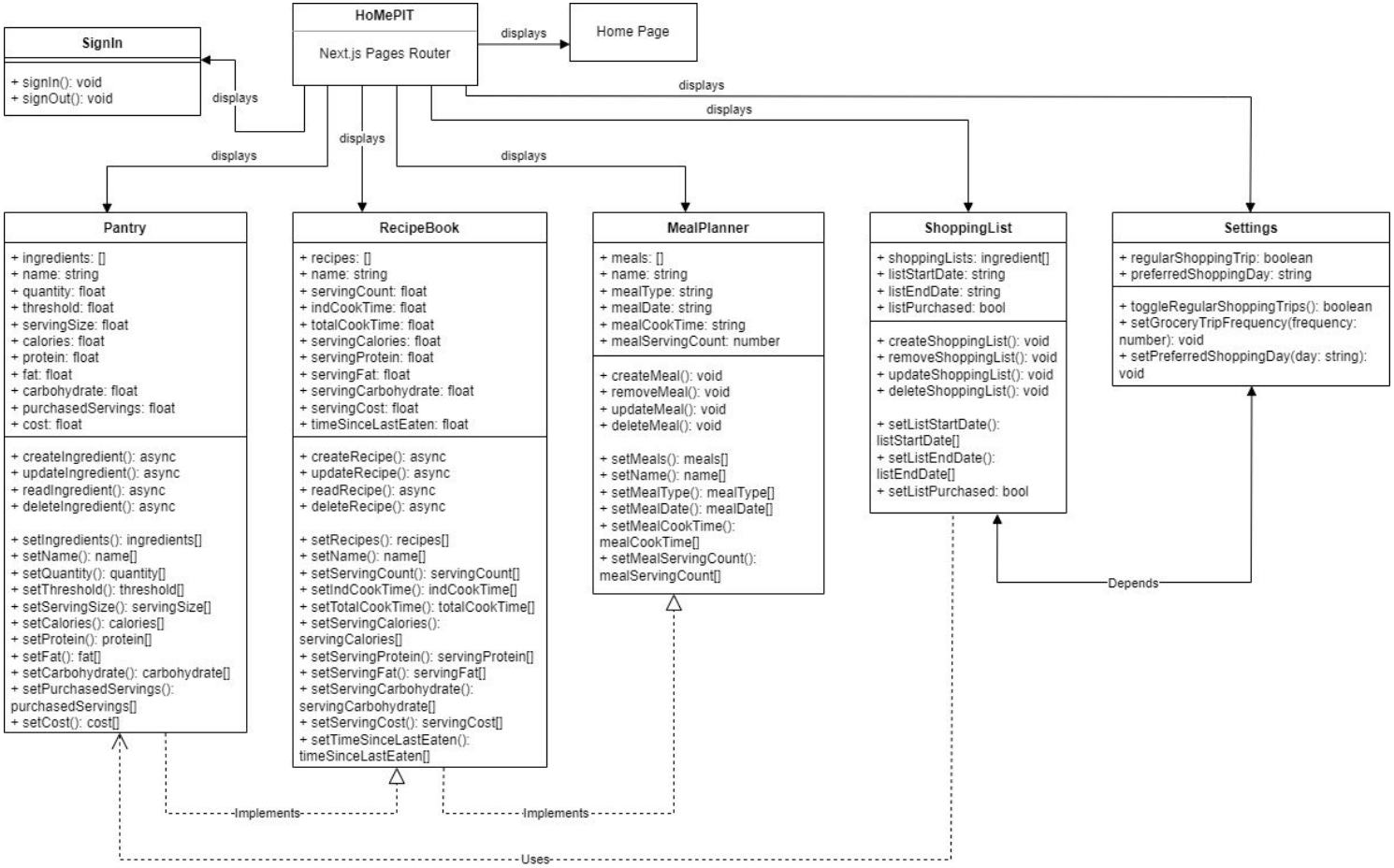
Current Changes, App pt.2

- Using React's useState() hook simplified assigning attributes to ingredients.
- The Pantry, Recipes, and Meal pages were overhauled to have essentially the same format, with state hooks, CRUD methods, and the HTML display.

```
export default function Pantry() {

  const [ingredients, setIngredients] = React.useState([]);
  const [name, setName] = React.useState('');
  const [quantity, setQuantity] = React.useState();
  const [threshold, setThreshold] = React.useState();
  const [servingSize, setServingSize] = React.useState();
  const [calories, setCalories] = React.useState();
  const [protein, setProtein] = React.useState();
  const [fat, setFat] = React.useState();
  const [carbohydrate, setCarbohydrate] = React.useState();
  const [purchasedServings, setPurchasedServings] = React.useState();
  const [cost, setCost] = React.useState();

  const createIngredient = async (e) => {
    e.preventDefault();
    try {
      const { data } = await supabase
        .from('Ingredients')
        .insert([
          {
            ing_name: name,
            ing_qnt: quantity,
            ing_threshold_qnt: threshold,
            ing_serv_qnt: servingSize,
            ing_serv_cal: calories,
            ing_serv_prot: protein,
            ing_serv_fat: fat,
            ing_serv_carb: carbohydrate,
            ing_purchase_serv_cnt: purchasedServings,
            ing_purchase_cost: cost,
          }
        ]);
      return data;
    } catch (error) {
      console.error(error);
    }
  };
```

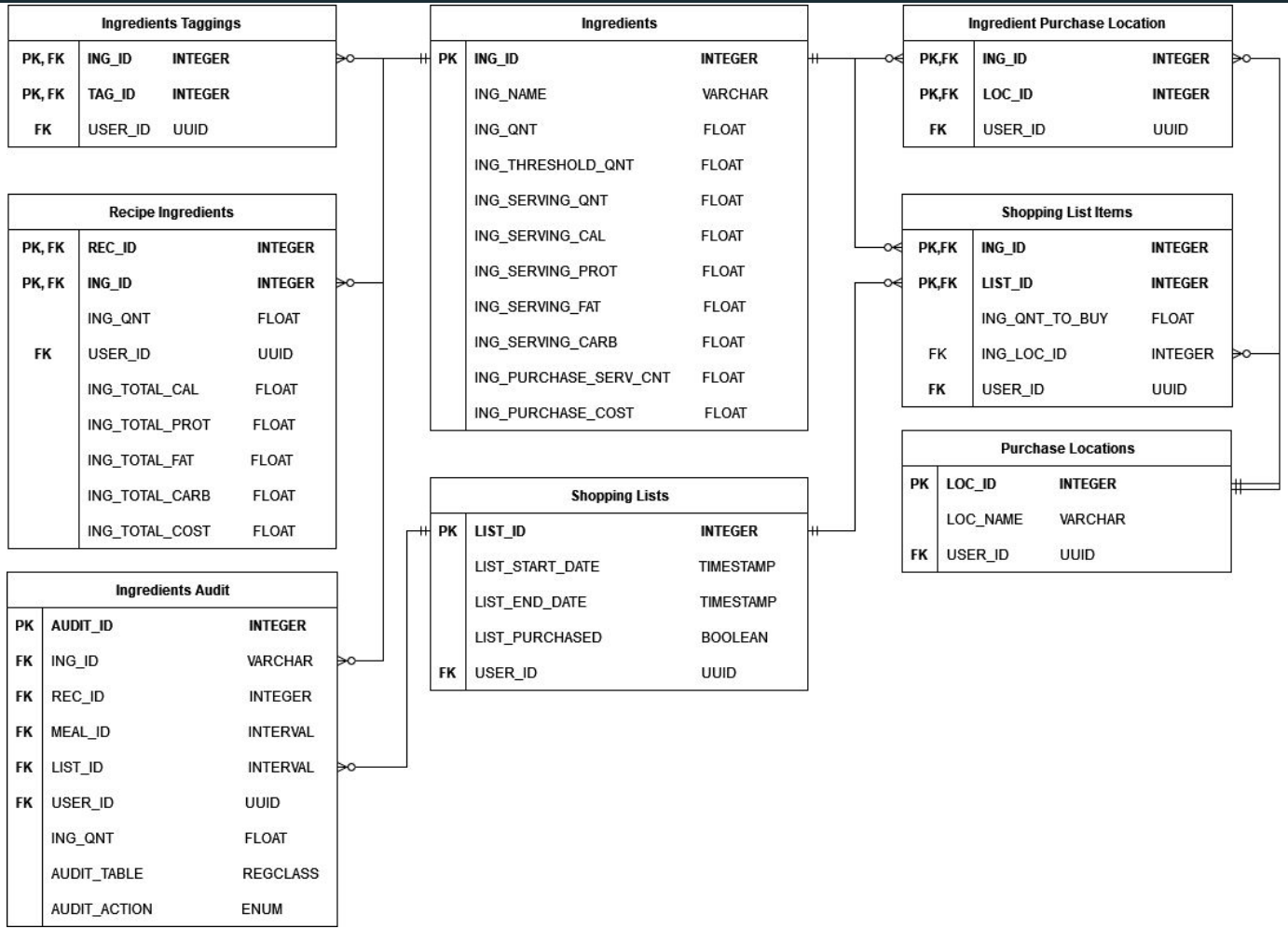

HoMePIT
Next.js Pages Router
displays
Home Page
SignIn
+ signIn(): void
+ signOut(): void
displays
displays
displays
displays
displays
displays
Pantry
+ ingredients: []
+ name: string
+ quantity: float
+ threshold: float
+ servingSize: float
+ calories: float
+ protein: float
+ fat: float
+ carbohydrate: float
+ purchasedServings: float
+ cost: float
+ createIngredient(): async
+ updateIngredient(): async
+ readIngredient(): async
+ deleteIngredient(): async
+ setIngredients(): ingredients[]
+ setName(): name[]
+ setQuantity(): quantity[]
+ setThreshold(): threshold[]
+ setServingSize(): servingSize[]
+ setCalories(): calories[]
+ setProtein(): protein[]
+ setFat(): fat[]
+ setCarbohydrate(): carbohydrate[]
+ setPurchasedServings(): purchasedServings[]
+ setCost(): cost[]
RecipeBook
+ recipes: []
+ name: string
+ servingCount: float
+ indCookTime: float
+ totalCookTime: float
+ servingCalories: float
+ servingProtein: float
+ servingFat: float
+ servingCarbohydrate: float
+ servingCost: float
+ timeSinceLastEaten: float
+ createRecipe(): async
+ updateRecipe(): async
+ readRecipe(): async
+ deleteRecipe(): async
+ setRecipes(): recipes[]
+ setName(): name[]
+ setServingCount(): servingCount[]
+ setIndCookTime(): indCookTime[]
+ setTotalCookTime(): totalCookTime[]
+ setServingCalories(): servingCalories[]
+ setServingProtein(): servingProtein[]
+ setServingFat(): servingFat[]
+ setServingCarbohydrate(): servingCarbohydrate[]
+ setServingCost(): servingCost[]
+ setTimeSinceLastEaten(): timeSinceLastEaten[]
MealPlanner
+ meals: []
+ name: string
+ mealType: string
+ mealDate: string
+ mealCookTime: string
+ mealServingCount: number
+ createMeal(): void
+ removeMeal(): void
+ updateMeal(): void
+ deleteMeal(): void
+ setMeals(): meals[]
+ setName(): name[]
+ setMealType(): mealType[]
+ setMealDate(): mealDate[]
+ setMealCookTime(): mealCookTime[]
+ setMealServingCount(): mealServingCount[]
ShoppingList
+ shoppingLists: ingredient[]
+ listStartDate: string
+ listEndDate: string
+ listPurchased: bool
+ createShoppingList(): void
+ removeShoppingList(): void
+ updateShoppingList(): void
+ deleteShoppingList(): void
+ setListStartDate(): listStartDate[]
+ setListEndDate(): listEndDate[]
+ setListPurchased: bool
Settings
+ regularShoppingTrip: boolean
+ preferredShoppingDay: string
+ toggleRegularShoppingTrips(): boolean
+ setGroceryTripFrequency(frequency: number): void
+ setPreferredShoppingDay(day: string): void
Depends
Implements
Implements
Uses

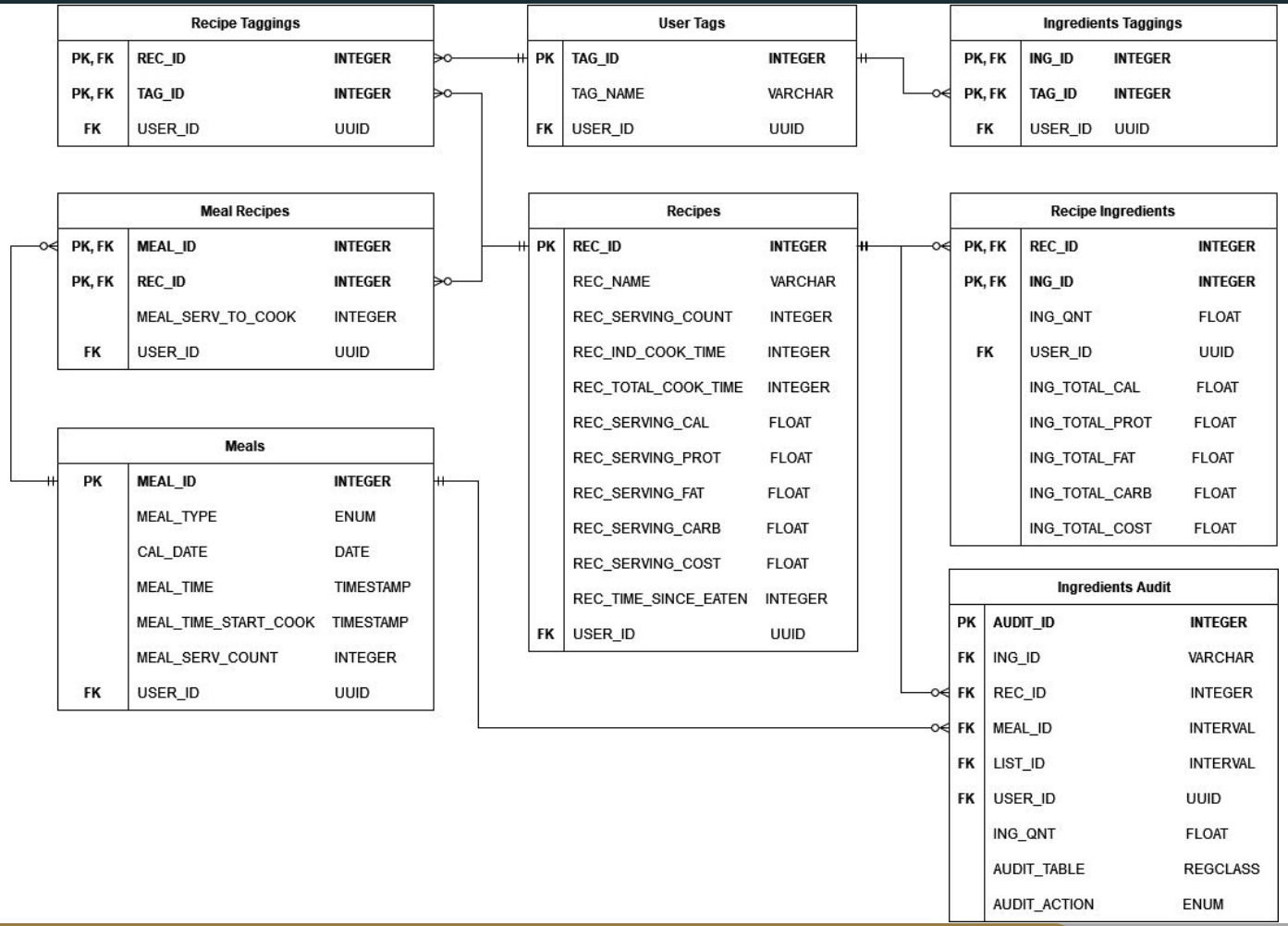
Current Changes, Database

- User ID added to every table
- Row Level Security policies added to every table
  - Only the owner of info can interact with it
- Added nutrient stats & cost to “Recipe Ingredients”
- “Recipe” nutrient stats & cost update on trigger
- Added “Ingredient Audit” table and related functions

Current Changes, Database pt. 2

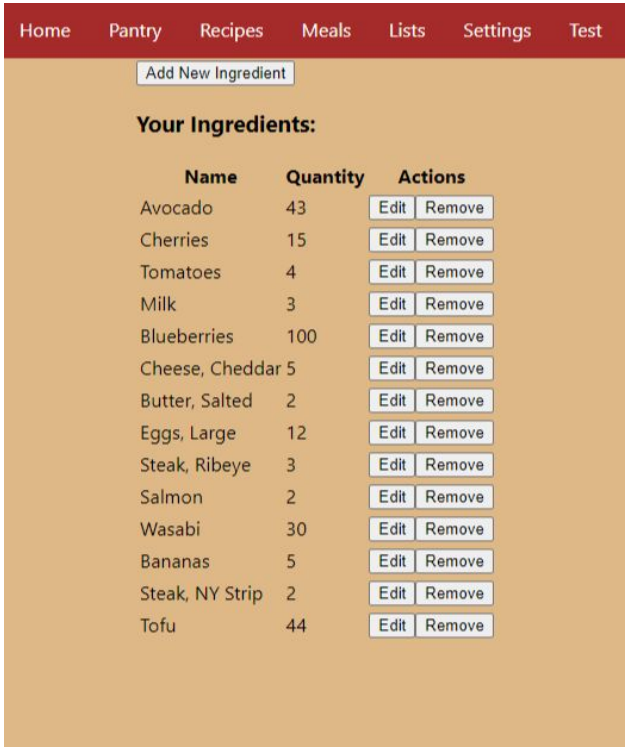
- Cutting substitution, sub-recipe, leftovers, and expiry date functionality
- Removed “Units” table
- Likely cutting shopping list purchase location optimization
- May cut purchase location functionality overall


Ingredients Taggings
PK, FK ING_ID INTEGER
PK, FK TAG_ID INTEGER
FK USER_ID UUID
Ingredients
PK ING_ID INTEGER
ING_NAME VARCHAR
ING_QNT FLOAT
ING_THRESHOLD_QNT FLOAT
ING_SERVING_QNT FLOAT
ING_SERVING_CAL FLOAT
ING_SERVING_PROT FLOAT
ING_SERVING_FAT FLOAT
ING_SERVING_CARB FLOAT
ING_PURCHASE_SERV_CNT FLOAT
ING_PURCHASE_COST FLOAT
Ingredient Purchase Location
PK,FK ING_ID INTEGER
PK,FK LOC_ID INTEGER
FK USER_ID UUID
Recipe Ingredients
PK, FK REC_ID INTEGER
PK, FK ING_ID INTEGER
ING_QNT FLOAT
FK USER_ID UUID
ING_TOTAL_CAL FLOAT
ING_TOTAL_PROT FLOAT
ING_TOTAL_FAT FLOAT
ING_TOTAL_CARB FLOAT
ING_TOTAL_COST FLOAT
Shopping List Items
PK,FK ING_ID INTEGER
PK,FK LIST_ID INTEGER
ING_QNT_TO_BUY FLOAT
FK ING_LOC_ID INTEGER
FK USER_ID UUID
Purchase Locations
PK LOC_ID INTEGER
LOC_NAME VARCHAR
FK USER_ID UUID
Shopping Lists
PK LIST_ID INTEGER
LIST_START_DATE TIMESTAMP
LIST_END_DATE TIMESTAMP
LIST_PURCHASED BOOLEAN
FK USER_ID UUID
Ingredients Audit
PK AUDIT_ID INTEGER
FK ING_ID VARCHAR
FK REC_ID INTEGER
FK MEAL_ID INTERVAL
FK LIST_ID INTERVAL
FK USER_ID UUID
ING_QNT FLOAT
AUDIT_TABLE REGCLASS
AUDIT_ACTION ENUM


Recipe Taggings
PK, FK REC_ID INTEGER
PK, FK TAG_ID INTEGER
FK USER_ID UUID
User Tags
PK TAG_ID INTEGER
TAG_NAME VARCHAR
FK USER_ID UUID
Ingredients Taggings
PK, FK ING_ID INTEGER
PK, FK TAG_ID INTEGER
FK USER_ID UUID
Meal Recipes
PK, FK MEAL_ID INTEGER
PK, FK REC_ID INTEGER
MEAL_SERV_TO_COOK INTEGER
FK USER_ID UUID
Recipes
PK REC_ID INTEGER
REC_NAME VARCHAR
REC_SERVING_COUNT INTEGER
REC_IND_COOK_TIME INTEGER
REC_TOTAL_COOK_TIME INTEGER
REC_SERVING_CAL FLOAT
REC_SERVING_PROT FLOAT
REC_SERVING_FAT FLOAT
REC_SERVING_CARB FLOAT
REC_SERVING_COST FLOAT
REC_TIME_SINCE_EATEN INTEGER
FK USER_ID UUID
Recipe Ingredients
PK, FK REC_ID INTEGER
PK, FK ING_ID INTEGER
ING_QNT FLOAT
FK USER_ID UUID
ING_TOTAL_CAL FLOAT
ING_TOTAL_PROT FLOAT
ING_TOTAL_FAT FLOAT
ING_TOTAL_CARB FLOAT
ING_TOTAL_COST FLOAT
Meals
PK MEAL_ID INTEGER
MEAL_TYPE ENUM
CAL_DATE DATE
MEAL_TIME TIMESTAMP
MEAL_TIME_START_COOK TIMESTAMP
MEAL_SERV_COUNT INTEGER
FK USER_ID UUID
Ingredients Audit
PK AUDIT_ID INTEGER
FK ING_ID VARCHAR
FK REC_ID INTEGER
FK MEAL_ID INTERVAL
FK LIST_ID INTERVAL
FK USER_ID UUID
ING_QNT FLOAT
AUDIT_TABLE REGCLASS
AUDIT_ACTION ENUM

Completed Accomplishments, App

- Authentication (almost) successfully implemented.
- User can create, update, and delete ingredients, recipes, and meals, and see those changes reflected in the page they are on.
- Site is responsive and updates accordingly, given changes in the database.



Completed Accomplishments, Database

- General database tables added
- “Recipe” and “Recipe Information” automatically update nutrition & cost info
  - Caused by certain actions in “Ingredients”, “Recipe Ingredients”, and “Recipes”
- “Ingredients Audit” table added

Partially Complete, App

- User authentication works, however after sign in the user is incapable of altering the contents of the database.
- Users can input data for ingredients/recipes/meals, but that data is unique to the instance of that object.
    - i.e. if the user were to create two ingredients with the same attributes, they would be represented as separate instead of being the same.
- A calendar component has been implemented for the Meal Planner, but has no functionality.

Partially Complete, Database

- Some “Ingredients Audit” auditing functions added
  - Can be triggered by actions on most tables
    - “Ingredients”, “Recipe Ingredients”, “Recipes”, “Meal Recipes”, “Meals”, “Shopping List Ingredients”, and “Shopping Lists”
    - Each trigger has unique potential impact - not generalizable

Still to be Implemented, App

- Data validation has yet to be implemented in the frontend.
  - If the user were to create an ingredient named “Tomatoes” and set the quantity as “Four”, this would result in an error in the console but not to the user.
- Unit conversion has yet to be implemented, which is a core component.
  - Inputting “5” as the user’s quantity of Salmon does not make sense. “5 lbs” would be more applicable.
- The entire Meal Planning component needs to be implemented, along with cost and nutrition data tracking.
- The Shopping List component has not yet been implemented, along with the associated Settings page.

Still to be Implemented, Database

- Shopping List generation
- Shopping List updating
- Shopping List purchase location optimization
  - Assuming this doesn’t get cut due to time
- General database access utility functions

Issues

- Minor / major: shopping list functionality may be cut due to time constraints.
  - First to be cut: purchase location optimization
  - Second: overall purchase location functionality
- Minor: cutting of substitution, sub-recipes, leftovers, and expiry dates
  - Impacts quality of life, not core functions
- Minor: issues with some css elements
  - Most borders are not appearing
  - The alignment of content is off on some pages