University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: Nov 29th 2024

Group Number: 40

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alex Dart	93792588	g1r1c	adart075@gmail.com
Griffin Velichko	74979287	w0n1r	griffin.velichko@gmail.com
Anna Friesen	33401860	w5h0a	annafriesen@shaw.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description

Project Description: This project allows users to interact with an application that generates meal plans and their associated recipes. For various recipes in a meal plan, users can view all ingredients, insert additional ingredients, filter ingredients by their total calories, and view highly rated recipes. Users also have the option to update their personal data, such as their name, country or preferred cuisine. For users who want to save time when prepping meals, there is functionality that allows them to find recipes that share ingredients with a chosen number of other recipes. Furthermore, the admin of this application can get a list of users with every allergy so that they can ensure these individuals are receiving diverse meal plans that fit within their dietary restrictions.

Final Schema: our schema remained unchanged from the version handed in for Milestone 3.

Query Descriptions

Query Type: INSERT
File Name: appService.js
Line Number: 82
Query: INSERT INTO RecipeHasIngredient (recipeID, ingredientName, quantity)
VALUES (:id, :name, :quantity)

Query Type: UPDATE
File Name: appService.js
Line Number: 341
Query:
UPDATE CLIENT SET FULLNAME=:newFullName, COUNTRY=:newCountry, CUISINE=:newCuisine, DIET=:newDiet, GROCERYSTORE=:newGroceryStore where USERID=:existingUserIDNum

Query Type: DELETE
File Name: appService.js
Line Number: 209
Query:
DELETE
FROM MEALPLAN
WHERE MEALPLANID = ${IDAsNumber}

Query Type: SELECTION
File Name: appService.js
Line Number: 96
Query: SELECT ra.overallRating, ra.userID, re.name, re.author
FROM rating ra, recipe re
WHERE ra.recipeID = re.ID AND overallRating >= :overallRating

Query Type: PROJECTION
File Name: appService.js
Line Number: 379
Query: SELECT ${fieldsString}
FROM INGREDIENTNUTRITIONALINFO

Query Type: JOIN
File Name: appService.js
Line Number: 241
Query:
SELECT r.NAME, SUM(rhi.QUANTITY * ini.CALORIES) AS TotalCalories
FROM RECIPE r
JOIN RECIPEHASINGREDIENT rhi ON r.ID = rhi.RECIPEID
JOIN INGREDIENTNUTRITIONALINFO ini ON rhi.INGREDIENTNAME = ini.NAME
GROUP BY r.NAME
HAVING SUM(rhi.QUANTITY * ini.CALORIES) > ${caloriesAsNumber}
Query Description: This query gets all of the recipes that have a Total Calories amount that is greater than the specified caloriesAsNumber value.

Query Type: Aggregation with GROUP BY
File Name: appService.js
Line Number: 297
Query:
SELECT r.NAME, SUM(rhi.QUANTITY * ini.CALORIES) AS TotalCalories, SUM(rhi.QUANTITY * ini.FAT) AS TotalFat, SUM(rhi.QUANTITY * ini.PROTEIN) AS TotalProtein
FROM MEALPLANCONTAINSRECIPE mp
JOIN RECIPE r on mp.RECIPEID = r.ID
JOIN RECIPEHASINGREDIENT rhi ON r.ID = rhi.RECIPEID
JOIN INGREDIENTNUTRITIONALINFO ini ON rhi.INGREDIENTNAME = ini.NAME
WHERE mp.MEALPLANID = ${mealPlanID}
GROUP BY r.NAME
Query Description: This query gets all of the recipes, along with their Total Calories, Total Fat and Total Protein that are in a provided meal plan, referenced by mealPlanID.

Query Type: Aggregation with HAVING
File Name: appService.js
Line Number: 392
Query: SELECT userID, COUNT(mealPlanID) as mealPlanCount
FROM UserCreatesMealPlan
GROUP BY userID
HAVING COUNT(mealPlanID) > :minMealPlansAsNumber
Query Description: This query, given a number of meal plans, finds all users that have created more than that number of meal plans, and returns their userID and the number of plans they've made.

Query Type: Nested aggregation with GROUP BY
File Name: appService.js
Line Number: 317
Query:

```
SELECT DISTINCT r.NAME
FROM RECIPE r
        JOIN RECIPEHASINGREDIENT rhi ON r.ID = rhi.RECIPEID
WHERE rhi.INGREDIENTNAME IN (
        SELECT INGREDIENTNAME
        FROM RECIPEHASINGREDIENT
        GROUP BY INGREDIENTNAME
        HAVING COUNT(DISTINCT RECIPEID) >= ${numRecipes})
```

Query Description: This query gets all of the recipes which contain at least one ingredient that appears in at least numRecipes other recipes.

Query Type: DIVISION
File Name: appService.js
Line Number: 172
Query:

```
SELECT c.fullName as "Full Name"
        FROM client c
        WHERE NOT EXISTS (
           (SELECT a.type
            FROM allergy a)
           MINUS
           (SELECT uha.allergyType
           FROM userHasAllergy uha
           WHERE uha.userID = c.userID)
                )
```

Query Description: This query returns all users that have every allergy in the database.