University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ____2____

Date: __2024-03-01_____

Group Number: ____49_____

Name	Student Number	CS Alias (Userid)	Preferred Email Address
Ian Qin	10687325	p6e5b	qinyi2333@outlook.com
Marvin Wu	76668078	r7d3b	mwu584@student.ubc.ca
Winston Shin	35537142	y1c2b	winstonshin@gmail.com

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

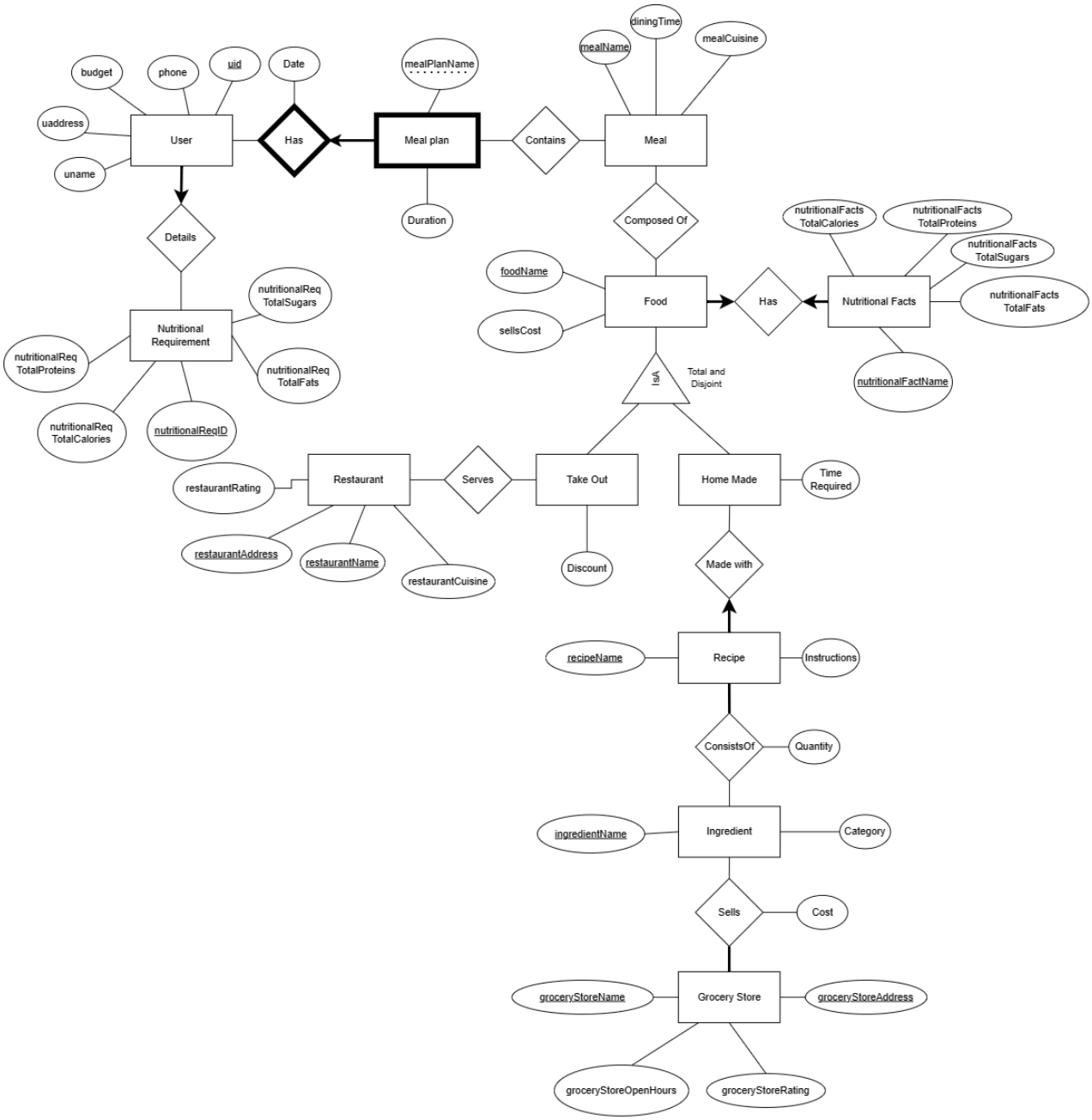
Section 2 - Summary

Our project is an app that allows its users to plan their meals in much detail over a set period of time. The users would initially input their budget, location, nutritional preferences, and preferences to eat in or out as a query. The app would help the user build a set meal schedule best accommodating their preferences, such as by allowing them to see each food choice's nutritional facts, which restaurant to buy their food, which grocery store to buy certain ingredients, and which recipe for making each food/meal.

Section 3 - Changes made on ER Diagram

- Added the ISA constraint: "Total and Disjoint" (TA suggestion: missing ISA constraint)
- Added a new entity called "Nutritional Requirement," replacing the nutritional requirement attribute which was originally part of the user entity. We felt that no domain would be appropriate, so we pulled out this attribute as an entity.
- Added a new relationship called "Details" between the "User" entity and "Nutritional Requirement" entity.
- Changed the relationship set name from "Contains" (between "Recipe" and "Ingredient") to "ConsistOf" to make every relationship name unique.
- Many-to-1 participation constraints for the new relation from "User" to "Nutritional Requirement" as every user would have a nutritional requirement, but a nutritional requirement doesn't have to belong to a specific user (TA suggestion).
- 1-to-1 participation constraints for the "Has" relationship between "Food" and "Nutritional Facts" as every food item has a nutritional fact, and every nutritional fact belongs to a specific food item (TA suggestion).
- 1-to-many participation constraint for the "MadeWith" relationship between "Recipe" and "HomeMade" as every recipe makes a specific food item but every food item could be made with different recipes (TA suggestion).
- Total participation between "Recipe" to "ConsistsOf" as every recipe should contain at least one ingredient (TA suggestion).
- Total participation between "GroceryStore" and "Sells" as every grocery store should sell one ingredient (TA suggestion).
- Renamed "Ingredients/Items" to "Ingredient" for precise semantic purposes (TA suggestion).
- Renamed "GroceryStores" to "GroceryStore" for precise semantic purposes.

All TA suggestions have been implemented. Please see updated ER Diagram on the next page.


budget
phone
uid
Date
uaddress
User
Has
Meal plan
mealPlanName
Contains
diningTime
mealName
mealCuisine
Meal
uname
Duration
Details
Composed Of
nutritionalFacts TotalCalories
nutritionalFacts TotalProteins
nutritionalFacts TotalSugars
foodName
Food
Has
Nutritional Facts
nutritionalFacts TotalFats
sellsCost
nutritionalFactName
nutritionalReq TotalSugars
Nutritional Requirement
nutritionalReq TotalProteins
nutritionalReq TotalFats
IsA
Total and Disjoint
nutritionalReq TotalCalories
nutritionalReqID
restaurantRating
Restaurant
Serves
Take Out
Home Made
Time Required
restaurantAddress
restaurantName
restaurantCuisine
Discount
Made with
recipeName
Recipe
Instructions
ConsistsOf
Quantity
ingredientName
Ingredient
Category
Sells
Cost
groceryStoreName
Grocery Store
groceryStoreAddress
groceryStoreOpenHours
groceryStoreRating

Section 4 - Schemas

Many of our entities have identical attribute names. To distinguish between identical attribute names from different entities, we added the name of the entity in front of the attribute names (i.e. the label for the attribute “name” from “MealPlan” is changed to “mealPlanName”).

Note: Attributes which can act as candidate keys are labeled with an asterisk.

- NutritionRequirement(nutritionalReqID: int, nutritionalReqTotalProteins: int, nutritionalReqTotalSugars: int, nutritionalReqTotalFats: int, nutritionalReqTotalCalories: int);
- User_Details(uid: int, **nutrionalReqID**: int, uname: varchar, uaddress: varchar, budget: int, phone: char[13]);
- MealPlan_Has(**uid**: int, mealPlanName: varchar, date: varchar, duration: int);
- Contains(**uid**: int, **mealPlanName**: varchar, **mealName**: varchar);
- Meal(mealName: varchar, diningTime: int, mealCuisine: varchar);
- ComposedOf(**uid**: int, **mealName**: varchar, **foodName**: varchar);
- NutritionalFacts_Has(nutritionalFactName: varchar, ***foodName**: varchar, calories: int, nutritionalFactTotalProteins: int, nutritionalFactTotalSugars: int, nutritionalFactTotalFats: int);
- Food(foodName: varchar, foodCost: int, ***nutritionalFactName**: varchar);
- TakeOut(**foodName**: varchar, discount: int);
- Restaurant(restaurantAddress: varchar, restaurantName: varchar, restaurantRating: float, restaurantCuisine: varchar);
- Serves(**restaurantAddress**: varchar, **restaurantName**: varchar, **foodName**: varchar);
- HomeMade(**foodName**: varchar, timeRequired: int);
- Recipe_MadeWith(recipeName: varchar, **foodName**: varchar, *instructions: varchar);
- ConsistsOf(**recipeName**: varchar, **ingredientName**: varchar);
- Ingredient(ingredientName: varchar, category: varchar);
- Sells(**ingredientName**: varchar, **groceryStoreName**: varchar, **groceryStoreAddress**: varchar, sellsCost: int);
- GroceryStore(groceryStoreName: varchar, groceryStoreAddress: varchar, openHours: varchar, groceryStoreRating: float);

Section 5 - Functional Dependencies

Explicit Functional Dependencies (Derived from PKs and CKs)

- nutritionalReqID → nutritionalReqTotalSugars, nutritionalReqTotalFats, nutritionalReqTotalProteins, nutritionalReqTotalCalories
- uid → uname, uaddress, budget, phone, nutritionalReqID
- uid, mealPlanName → duration, date

- mealName → diningTime, mealCuisine
- foodName → foodCost, nutritionalFactName, discount, timeRequired
- nutritionalFactName → foodCost, foodName, discount, timeRequired
- nutritionalFactName → nutritionalFactTotalCalories, nutritionalFactTotalProteins, nutritionalFactTotalSugars, nutritionalFactTotalFats, foodName
- foodName → nutritionalFactTotalCalories, nutritionalFactTotalProteins, nutritionalFactTotalSugars, nutritionalFactTotalFats, nutritionalFactName
- restaurantAddress, restaurantName → restaurantRating, restaurantCuisine
- recipeName → foodName, instructions
- ingredientName → category
- ingredientName, recipeName → quantity
- groceryStoreName, groceryStoreAddress → openHours, groceryStoreRating
- groceryStoreName, groceryStoreAddress, ingredientName → sellsCost

Implicit Functional Dependencies (Derived from attributes which are not PKs or CKs)

- restaurantCuisine → mealCuisine
- instructions → timeRequired
- instructions → recipeName
- nutritionalReqTotalSugars, nutritionalReqTotalFats, nutritionalReqTotalProteins → nutritionalReqTotalCalories
- nutritionalFactTotalSugars, nutritionalFactTotalFats, nutritionalFactTotalProteins → nutritionalFactTotalCalories

Section 6 - Normalization

We did a 3NF decomposition using a minimal cover and 3NF synthesis. Please see Appendix A for the work.

- R1 (nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats, nutritionalReqTotalProteins, nutritionalReqTotalCalories)
- R2 (uid, uname, uaddress, budget, phone, **nutritionalReqID**)
- R3 (**uid**, mealPlanName, duration, date)
- R4 (mealName, diningTime, mealCuisine)
- R5 (foodName, foodCost, **nutritionalFactName**, discount, timeRequired)
- R6 (nutritionalFactName, nutritionalFactTotalProteins, nutritionalFactTotalSugars, nutritionalFactTotalFats, nutritionalFactsTotalCalories, **foodName**)
- R7 (recipeName, instructions, **foodName**)
- R8 (restaurantName, restaurantAddress, restaurantRating, restaurantCuisine)
- R9 (ingredientName, category)
- R10 (**ingredientName**, **recipeName**, quantity)
- R11 (groceryStoreName, groceryStoreAddress, openHours, groceryStoreRating)
- R12 (**groceryStoreName**, **groceryStoreAddress**, **ingredientName**, sellsCost)
- R13 (**restaurantCuisine**, mealCuisine)

- R14 (**uid**, **mealName**, **mealPlanName**, **restaurantAddress**, **restaurantName**, **foodName**)
- R15 (**uid**, **mealName**, **mealPlanName**, **ingredientName**, **recipeName**, **groceryStoreName**, **groceryStoreAddress**, **foodName**)

Section 7 and 8 - SQL DDL / INSERT

```
CREATE TABLE R1(
        nutritionalReqID            INT PRIMARY KEY,
        nutritionalReqTotalSugars   INT,
        nutritionalReqTotalFats     INT,
        nutritionalReqTotalProteins INT,
        nutritionalReqTotalCalories INT
);

INSERT
INTO R1(nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats,
nutritionalReqTotalProteins, nutritionalReqTotalCalories)
VALUES(156, 250, 40, 48, 1700)

INSERT
INTO R1(nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats,
nutritionalReqTotalProteins, nutritionalReqTotalCalories)
VALUES(460, 300, 55, 70, 2400)

INSERT
INTO R1(nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats,
nutritionalReqTotalProteins, nutritionalReqTotalCalories)
VALUES(416, 483, 65, 280, 3400)

INSERT
INTO R1(nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats,
nutritionalReqTotalProteins, nutritionalReqTotalCalories)
VALUES(233, 483, 46, 211, 2400)

INSERT
INTO R1(nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats,
nutritionalReqTotalProteins, nutritionalReqTotalCalories)
VALUES(521, 280, 44, 120, 2000)

CREATE TABLE R2(
        uid         INT         PRIMARY KEY,
        uname       VARCHAR,
```

```
        uaddress        VARCHAR,
        budget          INT,
        phone           CHAR(13),
        nutritionalReqID INT,
        FOREIGN KEY (nutritionalReqID) REFERENCES R1(nutritionalReqID),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R2(uid, uname, uaddress, budget, phone, nutritionalReqID)
VALUES(131, 'Jennifer', '335 Elbing street', 150, '778-365-7145', 156)

INSERT
INTO R2(uid, uname, uaddress, budget, phone, nutritionalReqID)
VALUES(168, 'David', '1743 Grey Avenue', 120, '778-563-9865', 460)

INSERT
INTO R2(uid, uname, uaddress, budget, phone, nutritionalReqID)
VALUES(189, 'Chadwick', '8987 Webber street', 400, '778-451-2333', 416)

INSERT
INTO R2(uid, uname, uaddress, budget, phone, nutritionalReqID)
VALUES(457, 'Mathew', '133 14th street', 200, '236-446-3535', 233)

INSERT
INTO R2(uid, uname, uaddress, budget, phone, nutritionalReqID)
VALUES(381, 'Victoria', '8763 cross drive', 600, '778-867-9025', 521)


CREATE TABLE R3(
        uid                     INT,
        mealPlanName            VARCHAR,
        duration                INT,
        date                    VARCHAR,
        PRIMARY KEY (uid, mealPlanName),
        FOREIGN KEY (uid) REFERENCES R2(uid),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R3(uid, mealPlanName, duration, date)
VALUES(131, 'regular daily', 49, '2023-06-07'')
```

```
INSERT
INTO R3(uid, mealPlanName, duration, date)
VALUES(168, 'regular daily', 42, '2023-06-07'')

INSERT
INTO R3(uid, mealPlanName, duration, date)
VALUES(189, 'bulking', 30, '2023-11-11'')

INSERT
INTO R3(uid, mealPlanName, duration, date)
VALUES(457, 'low fat', 27, '2024-01-12'')

INSERT
INTO R3(uid, mealPlanName, duration, date)
VALUES(381, 'regular daily', 30, '2023-08-15'')


CREATE TABLE R4(
        mealName    VARCHAR     PRIMARY KEY,
        diningTime  INT,
        mealCuisine VARCHAR
);

INSERT
INTO R4(mealName, diningTime, mealCuisine)
VALUES('Steak with asparagus and potatoes', 40, 'Italian')

INSERT
INTO R4(mealName, diningTime, mealCuisine)
VALUES('Caesar salad with chicken breast', 25, 'American')

INSERT
INTO R4(mealName, diningTime, mealCuisine)
VALUES('Beef stir fry noodles', 30, 'Chinese')

INSERT
INTO R4(mealName, diningTime, mealCuisine)
VALUES('Kimchi fried rice', 20, 'Korean')

INSERT
INTO R4(mealName, diningTime, mealCuisine)
VALUES('Burger and Fries', 25, 'American)
```

```
CREATE TABLE R5(
        foodName            VARCHAR     PRIMARY KEY,
        foodCost            INT,
        nutritionalFactName VARCHAR,
        discount            INT,
        timeRequired        INT,
        UNIQUE (nutritionalFactName)
        FOREIGN KEY (nutritionalFactName) REFERENCES R6(nutritionalFactName),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R5(foodName, foodCost, nutritionalFactName, discount, timeRequired)
VALUES('Steak', 20, 'Steak', NULL, 30)

INSERT
INTO R5(foodName, foodCost, nutritionalFactName, discount, timeRequired)
VALUES('greek yogurt', 7, 'greek yogurt', NULL, 0)

INSERT
INTO R5(foodName, foodCost, nutritionalFactName, discount, timeRequired)
VALUES('lasagna ', 25, 'lasagna', 10, NULL)

INSERT
INTO R5(foodName, foodCost, nutritionalFactName, discount, timeRequired)
VALUES('Burger & Fries', 35, 'Burger & Fries, 20, NULL)

INSERT
INTO R5(foodName, foodCost, nutritionalFactName, discount, timeRequired)
VALUES('Kimchi fried rice', 20, 'kimchi fried rice', NULL, 50)


CREATE TABLE R6(
        nutritionalFactName          VARCHAR     PRIMARY KEY,
        nutritionalFactTotalProteins INT,
        nutritionalFactTotalSugars   INT,
        nutritionalFactTotalFats     INT,
        nutritionalFactTotalCalories INT,
        foodName                     VARCHAR,
        UNIQUE (foodName),
        FOREIGN KEY (foodName) REFERENCES R5(foodName)
```

```
        ON DELETE CASCADE
        ON UPDATE CASCADE);

INSERT
INTO R6(nutritionalFactName, nutritionalFactTotalProteins, nutritionalFactTotalSugars,
nutritionalFactTotalFats, foodName);
VALUES('Steak with Asparagus and potatoes', 10, 10, 10, 1000, 'Steak with Asparagus and
potatoes')

INSERT
INTO R6(nutritionalFactName, nutritionalFactTotalProteins, nutritionalFactTotalSugars,
nutritionalFactTotalFats, foodName);
VALUES('Caesar Salad Recipe', 11, 11, 11, 1001, 'Caesar Salad Recipe')

INSERT
INTO R6(nutritionalFactName, nutritionalFactTotalProteins, nutritionalFactTotalSugars,
nutritionalFactTotalFats, foodName);
VALUES('Greek yogurt and oats Recipe', 12, 12, 12, 1002, 'Greek yogurt and oats Recipe')

INSERT
INTO R6(nutritionalFactName, nutritionalFactTotalProteins, nutritionalFactTotalSugars,
nutritionalFactTotalFats, foodName);
VALUES('Burger & fries Recipe', 13, 13, 13, 1003, 'Burger & fries Recipe')

INSERT
INTO R6(nutritionalFactName, nutritionalFactTotalProteins, nutritionalFactTotalSugars,
nutritionalFactTotalFats, foodName);
VALUES('French toast Recipe", 14, 14, 14, 1004, "French toast Recipe')

CREATE TABLE R7(
        recipeName    VARCHAR     PRIMARY KEY,
        instructions    VARCHAR,
        foodName      VARCHAR,
        FOREIGN KEY (foodName) REFERENCES R5(foodName),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R7(recipeName, instructions, foodName);
VALUES("Steak with Asparagus and potatoes", "Grill steaks in a hot pan for 8 minutes then
cover steak with tin foil to rest, cook the asparagus in the steak pan and bake potatoes in the
oven at 425 F for 20 mins then cover with cheese to serve.", "Steak with Asparagus and
potatoes")
```

```
INSERT
INTO R7(recipeName, instructions, foodName);
VALUES("Caesar Salad Recipe", "Wash and cut lettuce into bite size; add croutons and ranch dressing and mix well. Optional topping of olives, bacon bits, cheese.", "Caesar Salad Recipe")

INSERT
INTO R7(recipeName, instructions, foodName);
VALUES("Greek yogurt and oats Recipe", "Slightly roast oats in the oven then in a bowl put in jam, greek yogurt, diced fruits of choice and oats when cooked to golden brown and rested to room temperature.", "Greek yogurt and oats Recipe")

INSERT
INTO R7(recipeName, instructions, foodName);
VALUES("Burger & fries Recipe", "Form a ball with ground beef before flattening, then put patty on pan at medium heat until brown. Turn off the stove and place a piece of american cheese and some onions on top of the patty. Put desired condiments & sliced vegetables on a bun and serve with fries. Wash and peel potatoes then cut them into strips. Dry with a towel then sprinkle some oil on top before putting in the air fryer,fry for 20 min then sprinkle salt to serve.", "Burger & fries Recipe")

INSERT
INTO R7(recipeName, instructions, foodName);
VALUES("French toast Recipe", "Create an egg wash by adding two eggs, vanilla extract and cinnamon into a bowl. Heat up the pan and place a small slice of butter. Dip toast in egg wash and fry until golden brown. Add condensed milk as topping before serving.", "French toast Recipe")

CREATE TABLE R8(
        restaurantName      VARCHAR,
        restaurantAddress   VARCHAR,
        restaurantRating    FLOAT,
        restaurantCuisine   VARCHAR,
        PRIMARY KEY (restaurantName, restaurantAddress, restaurantCuisine),
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

INSERT
INTO R8(restaurantName,restaurantAddress, restaurantRating, restaurantCuisine)
VALUES('Togo Sushi', '3380 Shrum Lane, Vancouver, BC', 3.7, 'Japanese');

INSERT
INTO R8(restaurantName,restaurantAddress, restaurantRating, restaurantCuisine)
```

```
VALUES(‘Mercante’’, ‘6488 University Blvd, Vancouver, BC’, 4.0, ‘Italian’’);

INSERT
INTO R8(restaurantName,restaurantAddress, restaurantRating, restaurantCuisine)
VALUES(My Home’, ‘5728 University Blvd B9, Vancouver, BC’, 4.1, Chinese’);

INSERT
INTO R8(restaurantName,restaurantAddress, restaurantRating, restaurantCuisine)
VALUES(‘The Corner Kitchen’’, ‘115-5743 Dalhousie Rd, Vancouver, BC’, 4.1, ‘Korean’’);

INSERT
INTO R8(restaurantName,restaurantAddress, restaurantRating, restaurantCuisine)
VALUES(‘McDonald’s’’, ‘5728 University Blvd #101, Vancouver, BC’, 3.4, ‘American’);

CREATE TABLE R9(
        ingredientName      VARCHAR     PRIMARY KEY,
        category            VARCHAR
);

INSERT
INTO R9(restaurantName, category)
VALUES(‘asparagus’, ‘vegetable’);

INSERT
INTO R9(restaurantName, category)
VALUES(‘greek yogurt’, ‘dairy’);

INSERT
INTO R9(restaurantName, category)
VALUES(‘steak’, ‘meat’);

INSERT
INTO R9(restaurantName, category)
VALUES(‘potato’, ‘vegetable’);

INSERT
INTO R9(restaurantName, category)
VALUES(‘lettuce’, ‘vegetable’);

INSERT
INTO R9(restaurantName, category)
VALUES(‘ranch’, ‘dressing’);

CREATE TABLE R10(
```

```
        ingredientName      VARCHAR,
        recipeName          VARCHAR,
        quantity            INT,
        PRIMARY KEY (ingredientName, recipeName),
        FOREIGN KEY (ingredientName) REFERENCES R9(ingredientName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (recipeName) REFERENCES R7(recipeName)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R10(ingredientName, recipeName, quantity)
VALUES ('ground beef ', ' Burger & Fries', 100)

INSERT
INTO R10(ingredientName, recipeName, quantity)
VALUES ('potato', ' Burger & Fries', 150)

INSERT
INTO R10(ingredientName, recipeName, quantity)
VALUES ('bread ', ' French Toast', 100)

INSERT
INTO R10(ingredientName, recipeName, quantity)
VALUES ('lettuce ', ' Caesar Salad', 200)

INSERT
INTO R10(ingredientName, recipeName, quantity)
VALUES ('tomato', ' Burger & Fries', 50)


CREATE TABLE R11(
        groceryStoreName    VARCHAR,
        groceryStoreAddress VARCHAR,
        openHours           VARCHAR,
        groceryStoreRating  FLOAT,
        PRIMARY KEY(groceryStoreName, groceryStoreAddress)
);

INSERT
INTO R11(groceryStoreName, groceryStoreAddress, openHours, groceryStoreRating)
VALUES ('Costco', '9151 Bridgeport Rd, Richmond, BC', '9:00-20:30', 4.1)
```

```
INSERT
INTO R11(groceryStoreName, groceryStoreAddress, openHours, groceryStoreRating)
VALUES (‘Costco’, ‘605 Expo Blvd, Vancouver, BC‘, ‘9:00-20:30’, 4.3)

INSERT
INTO R11(groceryStoreName, groceryStoreAddress, openHours, groceryStoreRating)
VALUES (‘Save On Foods’, ‘5945 Berton Ave, Vancouver, BC‘, ‘7:00-22:00’, 3.9)

INSERT
INTO R11(groceryStoreName, groceryStoreAddress, openHours, groceryStoreRating)
VALUES (‘Real Canadian Superstore’, ‘3185 Grandview Hwy, Vancouver, BC‘, ‘7:00-23:00’, 4.1)

INSERT
INTO R11(groceryStoreName, groceryStoreAddress, openHours, groceryStoreRating)
VALUES (‘Safeway’, ‘2733 W Broadway, Vancouver, BC‘, ‘7:00-23:00’, 4.1)

CREATE TABLE R12(
        groceryStoreName    VARCHAR,
        groceryStoreAddress VARCHAR,
        ingredientName      VARCHAR,
        sellsCost           INT,
        PRIMARY KEY(groceryStoreName, groceryStoreAddress, ingredientName),
        FOREIGN KEY(groceryStoreName, groceryStoreAddress) REFERENCES
        R11(groceryStoreName, groceryStoreAddress),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (ingredientName) REFERENCES R9(ingredientName),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R12(groceryStoreName, groceryStoreAddress, ingredientName, sellsCost)
VALUES(‘Costco’, ‘9151 Bridgeport Rd, Richmond, BC’, ‘asparagus’, ‘ ‘vegetable’);

INSERT
INTO R12(groceryStoreName, groceryStoreAddress, ingredientName, sellsCost)
VALUES(‘Costco’, ‘605 Expo Blvd, Vancouver, BC’, ‘greek yogurt’, ‘dairy’);

INSERT
INTO R12(groceryStoreName, groceryStoreAddress, ingredientName, sellsCost)
VALUES(‘Save On Foods’, ‘5945 Berton Ave, Vancouver, BC’, ‘steak’, ‘ ‘meat’);
```

```
INSERT
INTO R12(groceryStoreName, groceryStoreAddress, ingredientName, sellsCost)
VALUES('Real Canadian Superstore', '3185 Grandview Hwy, Vancouver, BC', 'potato', '
'vegetable');

INSERT
INTO R12(groceryStoreName, groceryStoreAddress, ingredientName, sellsCost)
VALUES('Safeway', '2733 W Broadway, Vancouver, BC', 'lettuce', ' 'vegetable');

INSERT
INTO R12(groceryStoreName, groceryStoreAddress, ingredientName, sellsCost)
VALUES('Costco', '9151 Bridgeport Rd, Richmond, BC', 'ranch', ' 'dressing');

CREATE TABLE R13(
        restaurantCuisine       VARCHAR     PRIMARY KEY,
        mealCuisine             VARCHAR,
        FOREIGN KEY(restaurantCuisine) REFERENCES R8(restaurantCuisine),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R13(restaurantCuisine, mealCuisine)
VALUES ('French', 'French')

INSERT
INTO R13(restaurantCuisine, mealCuisine)
VALUES ('Italian', 'Italian')

INSERT
INTO R13(restaurantCuisine, mealCuisine)
VALUES ('American', 'American')

INSERT
INTO R13(restaurantCuisine, mealCuisine)
VALUES ('Chinese', 'Chinese')

INSERT
INTO R13(restaurantCuisine, mealCuisine)
VALUES ('Korean', 'Korean')

CREATE TABLE R14(
        uid                     INT,
        nutritionalReqID        INT,
```

```
        mealName            VARCHAR,
        mealPlanName        VARCHAR,
        restaurantAddress   VARCHAR,
        restaurantName      VARCHAR,
        foodName            VARCHAR,
        FOREIGN KEY(uid) REFERENCES R2(uid),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(mealName) REFERENCES R4(mealName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(mealPlanName) REFERENCES R3(mealPlanName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(restaurantName, restaurantAddress) REFERENCES
        R8(restaurantName, restaurantAddress),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(foodName) REFERENCES R5(foodName),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT
INTO R14(uid, nutritionalReqID, mealName, mealPlanName, restaurantAddress,
restaurantName, foodName)
VALUES (131, 156, 'burger and fries', 'regular daily', '5728 University Blvd #101, Vancouver, BC',
'McDonald's', 'burger')

INSERT
INTO R14(uid, nutritionalReqID, mealName, mealPlanName, restaurantAddress,
restaurantName, foodName)
VALUES (168, 460, 'beef stir fry noodles ', 'regular daily', '5728 University Blvd B9, Vancouver,
BC', 'My Home Cuisine', 'noodles')

INSERT
INTO R14(uid, nutritionalReqID, mealName, mealPlanName, restaurantAddress,
restaurantName, foodName)
VALUES (131, 156, 'burger and fries', 'regular daily', '5728 University Blvd #101, Vancouver, BC',
'McDonald's', 'fries')

INSERT
INTO R14(uid, nutritionalReqID, mealName, mealPlanName, restaurantAddress,
restaurantName, foodName)
```

VALUES (189, 416, ‘beef stir fry noodles ’, ‘regular daily’, ‘5728 University Blvd B9, Vancouver, BC’, ‘My Home Cuisine’, ‘noodles’)

INSERT
INTO R14(uid, nutritionalReqID, mealName, mealPlanName, restaurantAddress, restaurantName, foodName)
VALUES (189, 416, ‘burger and fries’, ‘regular daily’, ‘5728 University Blvd #101, Vancouver, BC’, ‘McDonald’s’, ‘burger’)

```
CREATE TABLE R15(
        uid                     INT,
        nutritionalReqID        INT,
        mealName                VARCHAR,
        mealPlanName            VARCHAR,
        ingredientName          VARCHAR,
        recipeName              VARCHAR,
        groceryStoreName        VARCHAR,
        groceryStoreAddress     VARCHAR,
        foodName                VARCHAR,
        PRIMARY KEY (uid, mealName, mealPlanName, ingredientName, recipeName,
        groceryStoreName, groceryStoreAddress, foodName),
        FOREIGN KEY(uid) REFERENCES R2(uid)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(mealName) REFERENCES R4(mealName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(mealPlanName) REFERENCES R3(mealPlanName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(ingredientName) REFERENCES R9(ingredientName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(recipeName) REFERENCES R7(recipeName),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(groceryStoreName, groceryStoreAddress) REFERENCES
        R11(groceryStoreName, groceryStoreAddress),
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(foodName) REFERENCES R5(foodName),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
```

INSERT
INTO R15(uid, nutritionalReqID, mealName, mealPlanName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName)
VALUES(131, 156, ‘Steak with asparagus and potatoes’, ‘regular daily’, ‘asparagus’, ‘Steak with Asparagus and potatoes’, ‘Costco’, ‘9151 Bridgeport Rd, Richmond, BC’, ‘Steak with Asparagus and potatoes’);

INSERT
INTO R15(uid, nutritionalReqID, mealName, mealPlanName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName)
VALUES(131, 156, ‘Steak with asparagus and potatoes’, bulking’, ‘asparagus’, ‘Steak with Asparagus and potatoes’, ‘Costco’, ‘9151 Bridgeport Rd, Richmond, BC’, ‘Steak with Asparagus and potatoes’);

INSERT
INTO R15(uid, nutritionalReqID, mealName, mealPlanName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName)
VALUES(131, 156, ‘Steak with asparagus and potatoes’, ‘low fat’, ‘asparagus’, ‘Steak with Asparagus and potatoes’, ‘Costco’, ‘9151 Bridgeport Rd, Richmond, BC’, ‘Steak with Asparagus and potatoes’);

INSERT
INTO R15(uid, nutritionalReqID, mealName, mealPlanName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName)
VALUES(168, 460, ‘Steak with asparagus and potatoes’, ‘regular daily’, ‘asparagus’, ‘Steak with Asparagus and potatoes’, ‘Costco’, ‘9151 Bridgeport Rd, Richmond, BC’, ‘Steak with Asparagus and potatoes’);

INSERT
INTO R15(uid, nutritionalReqID, mealName, mealPlanName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName)
VALUES(189, 416, ‘Steak with asparagus and potatoes’, ‘regular daily’, ‘asparagus’, ‘Steak with Asparagus and potatoes’, ‘Costco’, ‘9151 Bridgeport Rd, Richmond, BC’, ‘Steak with Asparagus and potatoes’);

Appendix A - Finding Minimal Cover

Step I: Standardizing FD's

- nutritionalReqID → nutritionalReqTotalSugars
- nutritionalReqID → nutritionalReqTotalFats
- nutritionalReqID → nutritionalReqTotalProteins
- nutritionalReqID → nutritionalReqTotalCalories
- uid → uname
- uid → uaddress
- uid → budget
- uid → phone
- uid → nutritionalReqID
- uid, mealPlanName → duration
- uid, mealPlanName → date
- mealName → diningTime
- mealName → mealCuisine
- foodName → foodCost
- foodName → nutritionalFactName
- foodName → discount
- foodName → timeRequired

- nutritionalFactsName→ foodCost
- nutritionalFactsName→ foodName
- nutritionalFactsName→ discount
- nutritionalFactsName→ timeRequired

- nutritionalFactsName → nutritionalFactsTotalCalories
- nutritionalFactsName → nutritionalFactsTotalProteins
- nutritionalFactsName → nutritionalFactsTotalSugars
- nutritionalFactsName → nutritionalFactsTotalFats
- nutritionalFactsName → foodName

- foodName→ nutritionalFactsTotalCalories
- foodName→ nutritionalFactsTotalProteins
- foodName→ nutritionalFactsTotalSugars
- foodName→ nutritionalFactsTotalFats
- foodName→ nutritionalFactsName

- restaurantAddress, restaurantName → restaurantRating
- restaurantAddress, restaurantName → restaurantCuisine
- recipeName → foodName, instructions
- ingredientName → category

- ingredientName, recipeName → quantity
- groceryStoreName, groceryStoreAddress → openHours
- groceryStoreName, groceryStoreAddress → groceryStoreRating
- groceryStoreName, groceryStoreAddress, ingredientName → sellsCost
- restaurantCuisine → mealCuisine
- instructions → timeRequired
- instructions → recipeName
- nutritionalReqTotalSugars, nutritionalReqTotalFats, nutritionalReqTotalProteins → nutritionalReqTotalCalories
- nutritionalFactsTotalSugars, nutritionalFactsTotalFats, nutritionalFactsTotalProteins → nutritionalFactsTotalCalories

Step II: Reduce LHS

- No LHS was found to be verbose.

Step III: Remove redundancies

- nutritionalReqID → nutritionalReqTotalSugars
- nutritionalReqID → nutritionalReqTotalFats
- nutritionalReqID → nutritionalReqTotalProteins
- nutritionalReqID → nutritionalReqTotalCalories
- uid → uname
- uid → uaddress
- uid → budget
- uid → phone
- uid → nutritionalReqID
- uid, mealPlanName → duration
- uid, mealPlanName → date
- mealName → diningTime
- mealName → mealCuisine
- ~~nutritionalFactsName → foodCost~~
- nutritionalFactsName → foodName
- ~~nutritionalFactsName → discount~~
- ~~nutritionalFactsName → timeRequired~~
- foodName → foodCost
- foodName → nutritionalFactsName
- foodName → discount
- foodName → timeRequired
- nutritionalFactsName → nutritionalFactsTotalCalories
- nutritionalFactsName → nutritionalFactsTotalProteins
- nutritionalFactsName → nutritionalFactsTotalSugars
- nutritionalFactsName → nutritionalFactsTotalFats
- ~~nutritionalFactsName → foodName~~

- ~~foodName → nutritionalFactsTotalCalories~~
- ~~foodName → nutritionalFactsTotalProteins~~
- ~~foodName → nutritionalFactsTotalSugars~~
- ~~foodName → nutritionalFactsTotalFats~~
- ~~foodName → nutritionalFactsName~~
- restaurantAddress, restaurantName → restaurantRating
- restaurantAddress, restaurantName → restaurantCuisine
- recipeName → foodName, instructions
- ingredientName → category
- ingredientName, recipeName → quantity
- groceryStoreName, groceryStoreAddress → openHours
- groceryStoreName, groceryStoreAddress → groceryStoreRating
- groceryStoreName, groceryStoreAddress, ingredientName → sellsCost
- restaurantCuisine → mealCuisine
- instructions → timeRequired
- instructions → recipeName
- ~~nutritionalReqTotalSugars, nutritionalReqTotalFats, nutritionalReqTotalProteins → nutritionalReqTotalCalories~~
- ~~nutritionalFactsTotalSugars, nutritionalFactsTotalFats, nutritionalFactsTotalProteins → nutritionalFactsTotalCalories~~

Step IV: Find minimal key

LHS - uid, mealPlanName, mealName, restaurantAddress, restaurantName, ingredientName, groceryStoreName, groceryStoreAddress

MID - nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats, nutritionalReqTotalProteins, foodName, nutritionalFactsName, nutritionalFactsTotalProteins, nutritionalFactsTotalSugars, nutritionalFactsTotalFats, restaurantCuisine, recipeName, instructions

RHS - uname, uaddress, budget, phone, duration, date, diningTime, foodCost, discount, timeRequired, restaurantRating, category, quantity, openHours, groceryStoreRating, sellsCost, mealCuisine, nutritionalReqTotalCalories, nutritionalFactsTotalCalories

Because of these two relations:
- nutritionalFactName → foodName
- foodName → nutritionalFactName

We only need to choose one to include in our minimal key, and we chose foodName. Since nutritionalReqID determines nutritionalReqTotalSugars, nutritionalReqTotalFats, and nutritionalReqTotalProteins, we added it to our key.

Our minimal key includes the following attributes:

- uid, mealName, mealPlanName, restaurantAddress, restaurantName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName

However we noticed that it wouldn't make sense to include this minimal key into our relations since no food items would have non-null values for restaurantAddress, restaurantName, ingredientName, recipeName, groceryStoreName, and groceryStoreAddress at the same time due the ISA constraint. Therefore we divided this minimal key into two keys which would map each subclass to their respective and appropriate entities:

- uid, mealName, mealPlanName, restaurantAddress, restaurantName, foodName
- uid, mealName, mealPlanName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName

Step V: Merge relations

- R1 (nutritionalReqID, nutritionalReqTotalSugars, nutritionalReqTotalFats, nutritionalReqTotalProteins, nutritionalReqTotalCalories)
- R2 (uid, uname, uaddress, budget, phone, **nutritionalReqID**)
- R3 (**uid**, mealPlanName, duration, date)
- R4 (mealName, diningTime, mealCuisine)
- R5 (foodName, foodCost, **nutritionalFactName**, discount, timeRequired)
- R6 (nutritionalFactName, nutritionalFactTotalProteins, nutritionalFactTotalSugars, nutritionalFactTotalFats, nutritionalFactsTotalCalories, **foodName**)
- R7 (recipeName, instructions, **foodName**)
- R8 (restaurantName, restaurantAddress, restaurantRating, restaurantCuisine)
- R9 (ingredientName, category)
- R10 (**ingredientName**, **recipeName**, quantity)
- R11 (groceryStoreName, groceryStoreAddress, openHours, groceryStoreRating)
- R12 (**groceryStoreName**, **groceryStoreAddress**, **ingredientName**, sellsCost)
- R13 (**restaurantCuisine**, mealCuisine)
- R14 (**uid, mealName, mealPlanName, restaurantAddress, restaurantName, foodName**)
- R15 (**uid, mealName, mealPlanName, ingredientName, recipeName, groceryStoreName, groceryStoreAddress, foodName**)

APPENDIX B - Tuples

The following will list out the table rows that are planned for insertion. The section after will show the insertion SQL DDL.

Nutrition Requirement:

- ID 156, total protein: 48g, total sugar: 250g, total fats: 40g fat, total calories: 1700

- ID 460, total protein: 70g, total sugar: 300g, total fats: 55g fat, total calories: 2400
- ID 416, total protein: 280g, total sugar: 483g, total fats: 65g fat, total calories: 3400
- ID 233, total protein: 211g, total sugar: 367g, total fats: 46g fat, total calories: 2400
- ID 521, total protein: 120g, total sugar: 280g, total fats: 44g fat, total calories: 2000

User details:

- Name: Jennifer, ID 131, Nutritional requirement ID: 156, Address: 335 Elbing street, budget: $150 Phone: 778-365-7145
- Name: David, ID 168, Nutritional requirement ID: 460, Address: 1743 Grey Avenue, budget: $120 Phone: 778-563-9865
- Name: Chadwick, ID 189, Nutritional requirement ID: 416, Address: 8987 Webber street, budget: $400 Phone: 778-451-2333
- Name: Mathew, ID 457, Nutritional requirement ID: 233, Address: 133 14th street, budget: $200 Phone: 236-446-3535
- Name: Victoria, ID 381, Nutritional requirement ID: 521, Address: 8763 cross drive, budget: $600 Phone: 778-867-9025

Meal plan:

- User ID: 131, meal plan name: regular daily, starting date: 2023-06-07, duration: 49 days
- User ID: 168, meal plan name: regular daily, starting date: 2023-03-15, duration: 42 days
- User ID: 189, meal plan name: bulking, starting date: 2023-11-11, duration: 30 days
- User ID: 457, meal plan name: low fat starting date: 2024-01-12, duration: 27 days
- User ID: 381, meal plan name: regular daily, starting date: 2023-08-15, duration: 30 days

Meal:

- Meal name: Steak with asparagus and potatoes, dining time: 40 min, Cuisine type: Italian
- Meal name: Caesar salad with chicken breast, dining time: 25 min, Cuisine type: American
- Meal name: greek yogurt and oats, dining time: 10 min, Cuisine type: American
- Meal name: Beef stir fry noodles, dining time: 30 min, Cuisine type: Chinese
- Meal name: kimchi fried rice, dining time: 20 min, Cuisine type: Korean
- Meal name: burger and fries, dining time: 25 min, Cuisine type: American
- Meal name: French toast, dining time: 15 min, Cuisine type: French

Food nutrition fact:

- Name: Steak, cost: $15, calories: 300, sugar: 0 g, protein: 30 g, fat: 15 g
- Name: Caesar salad, cost: $8, calories: 200, sugar: 12 g, protein: 5 g, fat: 15 g
- Name: greek yogurt, cost: $6, calories: 120, sugar: 6 g, protein: 17 g, fat: 1 g
- Name: beef noodle stir fry, cost: $10, calories: 400, sugar 30 g, protein: 20 g, fat: 15 g
- Name: kimchi fried rice, cost: $12, calories: 350, sugar: 44 g, protein: 10 g, fat: 12 g
- Name: mashed potatoes, cost: $4, calories: 210, sugar: 30 g, protein: 3 g, fat: 7 g
- Name: asparagus, cost: $3, calories: 30, sugar: 2.1 g, protein: 0.3 g, fat: 0.2 g

Restaurant:

- restaurant address: (3380 Shrum Lane, Vancouver, BC), restaurant name:Togo Sushi, restaurant rating: 3.7, restaurant cuisine: Japanese
- restaurant address: (6488 University Blvd, Vancouver, BC), restaurant name:Mercante, restaurant rating: 4.0, restaurant cuisine: Italian
- restaurant address: (5728 University Blvd B9, Vancouver, BC), restaurant name:My Home Cuisine, restaurant rating: 3.7, restaurant cuisine: Chinese
- restaurant address: (115-5743 Dalhousie Rd, Vancouver, BC), restaurant name:The Corner Kitchen, restaurant rating: 4.1, restaurant cuisine: Korean
- restaurant address: (5728 University Blvd #101, Vancouver, BC), restaurant name:McDonald's, restaurant rating: 3.4, restaurant cuisine: American

Grocery store:
- Grocery store name: Costco, Grocery store address: (9151 Bridgeport Rd, Richmond, BC), open hours: 9:00-20:30, store rating: 4.1
- Grocery store name: Costco, Grocery store address: (605 Expo Blvd, Vancouver, BC), open hours: 9:00-20:30, store rating: 4.3
- Grocery store name: Save On Foods, Grocery store address: (5945 Berton Ave, Vancouver, BC), open hours: 7:00-22:00, store rating: 3.9
- Grocery store name: Real Canadian Superstore, Grocery store address: (3185 Grandview Hwy, Vancouver, BC), open hours: 7:00-23:00, store rating: 4.1
- Grocery store name: Safeway, Grocery store address: (2733 W Broadway, Vancouver, BC), open hours: 7:00-23:00, store rating: 4.1

Ingredient:
- name: asparagus, type: vegetable
- name: greek yogurt, type: diary
- name: steak, type: meat
- name: potato, type: vegetable
- name: lettuce, type: vegetable
- name: ranch, type: dressing

Recipe:

Steak with Asparagus and potatoes:
Grill steaks in a hot pan for 8 minutes then cover steak with tin foil to rest, cook the asparagus in the steak pan and bake potatoes in the oven at 425 F for 20 mins then cover with cheese to serve.

Caesar Salad Recipe:
Wash and cut lettuce into bite size; add croutons and ranch dressing and mix well. Optional topping of olives, bacon bits, cheese.

Greek yogurt and oats Recipe:
Slightly roast oats in the oven then in a bowl put in jam, greek yogurt, diced fruits of choice and oats when cooked to golden brown and rested to room temperature.

Burger & fries Recipe:
Form a ball with ground beef before flattening, then put patty on pan at medium heat until brown. Turn off the stove and place a piece of american cheese and some onions on top of the patty. Put desired condiments & sliced vegetables on a bun and serve with fries.
Wash and peel potatoes then cut them into strips. Dry with a towel then sprinkle some oil on top before putting in the air fryer,fry for 20 min then sprinkle salt to serve.

French toast Recipe:
Create an egg wash by adding two eggs, vanilla extract and cinnamon into a bowl. Heat up the pan and place a small slice of butter. Dip toast in egg wash and fry until golden brown. Add condensed milk as topping before serving.