

CPSC 304 Project Cover Page

Milestone #: 2

Date: 11 June 2023

Group Number: 25

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---------------|----------------|-------------------|--------------------------|
| Eric Wong-Liu | 43745637 | a4z1l | ericwongliu@hotmail.com |
| Xuan Tung Luu | 30236798 | q6s5q | xuantung.brian@gmail.com |
| Daniel Lee | 18576249 | y9w1b | 2hyungyu@naver.com |

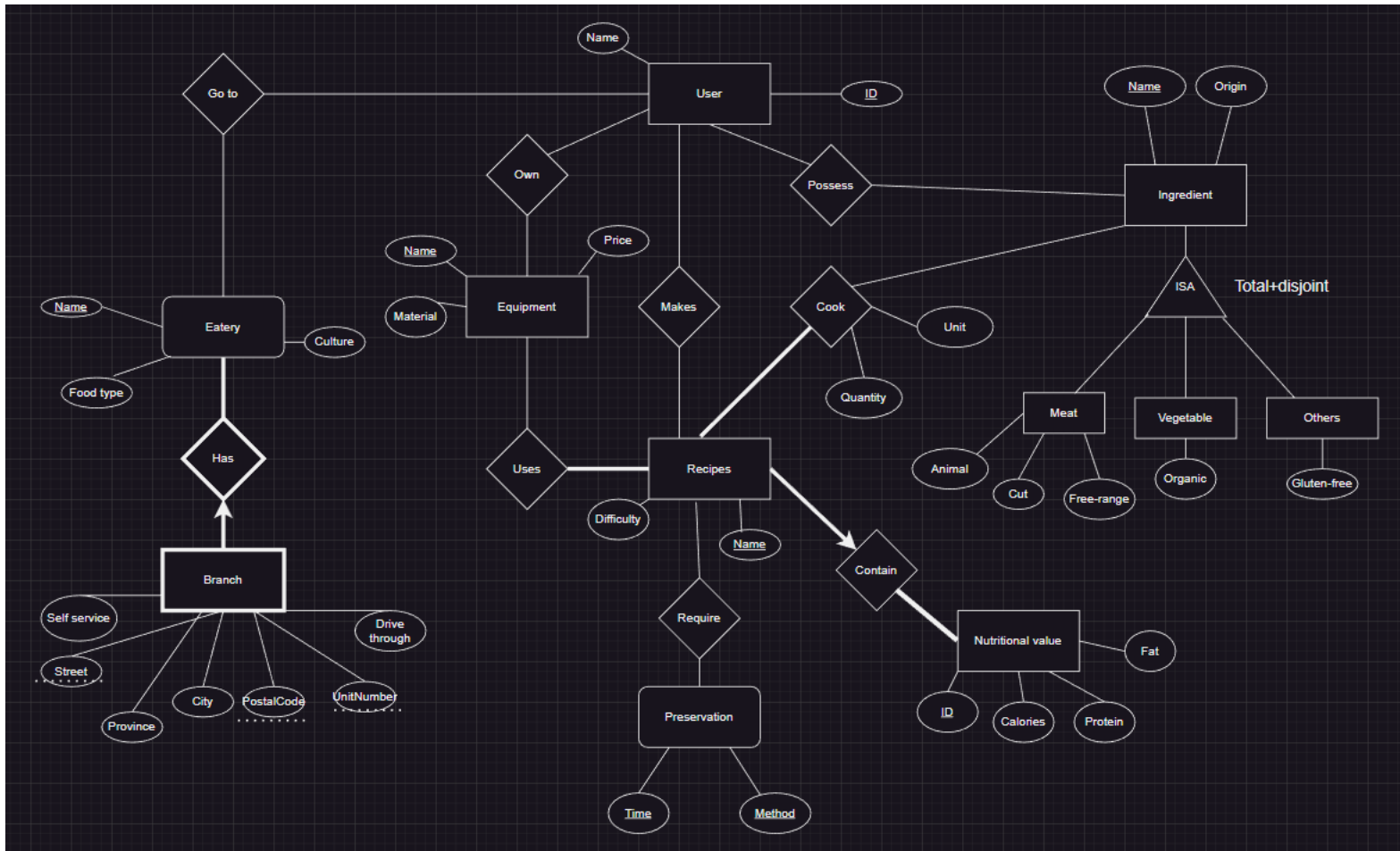
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

Project Summary:

Our project will allow users to query for recipes based on the ingredients and equipment that they have. They will have access to information about the recipe such as nutritional value and preservation method. Users can also query for restaurants instead based on criterias such as the type of food.

ER diagram:



Changes from milestone 1:

- Added the attributes **unit** and **quantity** to the **uses** relationship between **recipes** and **ingredients** to indicate how much of each ingredient does the recipe require.
- To avoid confusion, we renamed duplicate relation names from “Has” to “Own” and “Possess”, and from “Uses” to “Cook”
- Change the first letter in each word to be uppercase
- Added key constraint from Recipe to Nutritional Value since it makes more sense for each recipe to have only one associated nutritional value.

Schemas:

Underline = Primary Key (It is implied that all underlined attributes are NOT NULL)

Bold = Foreign key

- User(Name: char[50], ID: Integer)
- Makes(ID: char[50], **RecName**: char[50])
- RecipesContain(Difficulty: char[50], Name: char[50], **NutID**: Integer)
 - NOT NULL: NutID
- GoTo(ID: Integer, **Name**: char[50])
- Eatery(Name: char[50], Culture: char[50], FoodType: char[50])
- BranchHas(DriveThrough: Integer, Street: char[50], PostalCode: char[50], UnitNumber: Integer, Self-service: Integer, **EatName**: char[50], Province: char[2], City: char[50])
- Uses(**EquipName**: char[50], **RecName**: char[50])
- Equipment(Name: char[50], Material: char[50], Price: Integer)
- Own(ID: char[50], **EquipName**: char[50])
- Possess(ID: char[50], **IngName**: char[50])
- Ingredient(Name: char[50], Origin : char[50]):
 - Meat(Animal: char[50], Cut: char[50], Free-range: Integer, **Name** : char[50])
 - Vegetable(Organic : char[50], **Name** : char[50])
 - Others(Gluten-free: Integer, **Name** : char[50])
- Cook(**RecName** : char[50], **IngName** : char[50], Unit: char[50], Quantity: Integer)
- Require(**RecName**: char[50], **PreserveMethod**: char[50])
- Preservation(Time: char[50], Method: char[50])
- Nutritional value (ID: Integer, Calories: Integer, Protein: Integer, Fat: Integer)

The following participation constraints cannot be modeled without assertions:

- Each Recipe uses at least one Equipment
- Each Recipe uses at least one Ingredient

Functional dependencies:

- User()
 - ID \rightarrow Name
- Eatery()
 - Name \rightarrow Culture, FoodType
- Equipment()
 - Name \rightarrow Price, Material
- RecipesContain()
 - Name \rightarrow Difficulty, NutID
- Ingredient()
 - Name \rightarrow Origin
- Meat()
 - Name \rightarrow Animal, Cut, Free-range
- Vegetable()
 - Name \rightarrow Organic
- Others()
 - Name \rightarrow Gluten-free
- NutritionalValue()
 - ID \rightarrow Fat, Calories, Protein
- Cook()
 - RecName, IngName \rightarrow Unit, Quantity
- BranchHas()
 - Postal Code \rightarrow City, Province
 - EatName, Street, PostalCode, UnitNumber \rightarrow SelfService, DriveThrough, City, Province

University of British Columbia, Vancouver

Department of Computer Science

Normalization:

- User(Name: char[50], ID: Integer)
- Makes(ID: char[50], RecName: char[50])
- RecipesContain(Difficulty: char[50], Name: char[50], NutID: Integer)
 - Not Null: NutID
- GoTo(ID: Integer, Name: char[50])
- Eatery(Name: char[50], Culture: char[50], FoodType: char[50])
- Uses(EquipName: char[50], RecName: char[50])
- Equipment(Name: char[50], Material: char[50], Price: Integer)
- Own(ID: char[50], EquipName: char[50])
- Possess(ID: char[50], IngName: char[50])
- Ingredient(Name: char[50], Origin : char[50]):
 - Meat(Animal: char[50], Cut: char[50], Free-range: Integer, Name: char[50])
 - Vegetable(Organic : char[50], Name: char[50])
 - Others(Gluten-free: Integer, Name: char[50])
- Cook(RecName: char[50], IngName: char[50], Unit: char[50], Quantity: Integer)
- Require(RecName: char[50], PreserveMethod: char[50])
- Preservation(Time: char[50], Method: char[50])
- Nutritional value (ID: Integer, Calories: Integer, Protein: Integer, Fat: Integer)
- BranchHas1(DriveThrough: Integer, Street: char[50], PostalCode: char[50], UnitNumber: Integer, Self-service: Integer, EatName: char[50])
- BranchHas2(PostalCode: char[50], Province: char[2], City: char[50])

In BranchHas relation:

{PostalCode}+ = {City, Province, PostalCode}

{EatName, Street, PostalCode, UnitNumber}+ = {SelfService, DriveThrough, City, Province, EatName, Street, PostalCode, UnitNumber}

Decompose BranchHas into BranchHas1 and BranchHas2 because FD PostalCode -> City, Province did not satisfy BCNF. So we have BranchHas2(PostalCode, City, Province) and BranchHas1(DriveThrough, Street, PostalCode, UnitNumber, Self-service, Eatname) (based on FD PostalCode -> City, Province).

University of British Columbia, Vancouver

Department of Computer Science

SQL DDL statements:

```
CREATE TABLE User (  
    Name char[50],  
    ID INT PRIMARY KEY  
);
```

```
CREATE TABLE Makes (  
    ID CHAR(50),  
    RecName CHAR(50),  
    PRIMARY KEY (ID, RecName),  
    Foreign Key (RecName) References RecipesContain(Name),  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    Foreign Key (ID) References User(ID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE RecipesContain (  
    Difficulty CHAR(50),  
    Name CHAR(50),  
    NutID INT NOT NULL,  
    PRIMARY KEY (Name),  
    FOREIGN KEY (NutID) REFERENCES NutritionValue(ID)  
        ON DELETE SET DEFAULT  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE GoTo (  
    ID INT,  
    Name CHAR(50),  
    PRIMARY KEY (ID, Name),  
    FOREIGN KEY (ID) REFERENCES User(ID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    FOREIGN KEY (Name) REFERENCES Eatery(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Eatery (  
    Name CHAR(50),  
    Culture CHAR(50),
```

University of British Columbia, Vancouver

Department of Computer Science

```
FoodType CHAR(50),  
PRIMARY KEY (Name)  
);
```

```
CREATE TABLE BranchHas1 (  
    DriveThrough INT,  
    Street CHAR(50),  
    PostalCode CHAR(50),  
    UnitNumber INT,  
    SelfService INT,  
    EatName CHAR(50) NOT NULL,  
    PRIMARY KEY (Street, PostalCode, UnitNumber, EatName),  
    FOREIGN KEY (PostalCode) REFERENCES BranchHas2(PostalCode)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    FOREIGN KEY (EatName) REFERENCES Eatery(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE BranchHas2 (  
    PostalCode CHAR(50),  
    Province CHAR(2),  
    City CHAR(50),  
    PRIMARY KEY (PostalCode)  
);
```

```
CREATE TABLE Uses (  
    EquipName CHAR(50),  
    RecName CHAR(50),  
    PRIMARY KEY (EquipName , RecName),  
    FOREIGN KEY (EquipName) REFERENCES Equipment (Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    FOREIGN KEY (RecName) REFERENCES RecipesContain (Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Equipment (  
    Name CHAR(50),  
    Material CHAR(50),  
    Price INT,  
    PRIMARY KEY (Name)
```

University of British Columbia, Vancouver

Department of Computer Science

);

```
CREATE TABLE Own (  
    ID CHAR(50),  
    EquipName CHAR(50)  
    PRIMARY KEY(ID, EquipName),  
    FOREIGN KEY (ID) REFERENCES User(ID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    FOREIGN KEY (EquipName) REFERENCES Equipment(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Possess (  
    ID CHAR(50),  
    IngName CHAR(50),  
    PRIMARY KEY(ID, IngName),  
    FOREIGN KEY (ID) REFERENCES User(ID)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    FOREIGN KEY (IngName) REFERENCES Ingredient(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Ingredient (  
    Name CHAR(50),  
    Origin CHAR(50)  
    PRIMARY KEY(Name)  
);
```

```
CREATE TABLE Meat (  
    Animal CHAR(50),  
    Cut CHAR(50),  
    FreeRange INT,  
    Name CHAR(50),  
    PRIMARY KEY(Name),  
    FOREIGN KEY (Name) REFERENCES Ingredient(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```


University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE Vegetable (  
    Organic CHAR(50),  
    Name CHAR(50),  
    PRIMARY KEY(Name),  
    FOREIGN KEY (Name) REFERENCES Ingredient(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Others (  
    GlutenFree INT,  
    Name CHAR(50),  
    PRIMARY KEY(Name),  
    FOREIGN KEY (Name) REFERENCES Ingredient(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Cook (  
    RecName CHAR(50),  
    IngName CHAR(50),  
    Unit CHAR(50),  
    Quantity INT,  
    PRIMARY KEY(RecName, IngName),  
    FOREIGN KEY (RecName) REFERENCES RecipesContain(Name),  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    FOREIGN KEY (IngName) REFERENCES Ingredient(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

```
CREATE TABLE Require (  
    RecName CHAR(50),  
    PreserveMethod CHAR(50),  
    PRIMARY KEY(RecName , PreserveMethod ),  
    FOREIGN KEY (RecName) REFERENCES RecipesContain(Name)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
    FOREIGN KEY (PreserveMethod ) REFERENCES Preservation(Method)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE  
);
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE Preservation (  
    Time CHAR(50),  
    Method CHAR(50),  
    PRIMARY KEY(Time , Method)  
);
```

```
CREATE TABLE "Nutritional value" (  
    ID INT,  
    Calories INT,  
    Protein INT,  
    Fat INT,  
    PRIMARY KEY(ID)  
);
```

The following participation constraints cannot be modeled without assertions:

- Each Recipe uses at least one Equipment
- Each Recipe uses at least one Ingredient

University of British Columbia, Vancouver

Department of Computer Science

INSERT statements:

INSERT INTO User(Name, ID) VALUES

('Daniel', '1'),
('Eric', '2'),
('Tung', '3'),
('Albert', '4'),
('Sam', '5');

INSERT INTO Makes(ID, RecName) VALUES

('1', 'Zucchini Smoothie'),
('2', 'Zucchini Smoothie'),
('3', 'Zucchini Smoothie'),
('4', 'Grilled Tomahawk Steak'),
('5', 'Maple Cupcakes');

INSERT INTO RecipesContain(Difficulty, Name, NutID) VALUES

('Easy', 'Zucchini Smoothie', 1),
('Hard', 'Grilled Tomahawk Steak', 2),
('Medium', 'Maple Cupcakes', 3),
('Easy', 'Scrambled Eggs', 4),
('Easy', 'Scotch Eggs', 5);

INSERT INTO Eatery(Name, Culture, FoodType) VALUES

('Sula Indian Restaurant', 'Indian', 'Traditional Restaurant'),
('Tim Hortons', 'Western', 'Fast food'),
('McDonalds', 'Western', 'Fast food'),
('Starbucks', 'Western', 'Coffee'),
('French Table', 'French', 'Traditional Restaurant');

INSERT INTO Equipment(Name, Material, Price) VALUES

('Oven', 'Stainless steel', '2000'),
('Mixers', 'Stainless steel', '300'),
('Cutting board', 'Wood', '20'),
('Refrigerator', 'Stainless steel', '1000'),
('Spatula', 'Stainless steel', '40');

INSERT INTO Ingredient(Name, Origin) VALUES

('Canola Oil', 'AB'),
('Salt', 'AB'),
('Rice', 'NS'),
('Tomatoes', 'ON'),
('Potatoes', 'BC'),
('Beef Rib', 'AB');

University of British Columbia, Vancouver

Department of Computer Science

('Beef Loin', 'AB'),
('Fish Tail', 'NS'),
('Chicken Wing', 'ON'),
('Chicken Breast', 'ON'),
('Carrots', 'AB'),
('Corn', 'AB'),
('Peas', 'AB'),
('Onions', 'AB');

INSERT INTO Preservation(Time, Method) VALUES

('3m', 'Blanching'),
('15s', 'Pasteurization'),
('15m', 'Sterilization'),
('1d', 'Cool storage'),
('2d', 'Fermentation');

INSERT INTO Nutritional value(ID, Calories, Protein, Fat) VALUES

('1', '70', '2', '1'),
('2', '300', '30', '20'),
('3', '350', '4', '20'),
('4', '140', '12', '10'),
('5', '240', '12', '15');

INSERT INTO Meat(Animal, Cut, FreeRange, Name) VALUES

('Cow', 'Rib', '1', 'Beef Rib'),
('Cow', 'Loin', '1', 'Beef Loin'),
('Fish', 'Tail', '0', 'Fish Tail'),
('Chicken', 'Breast', '0', 'Chicken Breast'),
('Chicken', 'Wing', '1', 'Chicken Wing');

INSERT INTO Vegetable(Organic, Name) VALUES

('1', 'Potatoes'),
('1', 'Carrots'),
('1', 'Corn'),
('1', 'Peas'),
('1', 'Onions');

INSERT INTO Others(Gluten-free, Name) VALUES

('1', 'Canola oil'),
('1', 'Tomatoes'),
('1', 'Rice'),
('1', 'Salt'),
('1', 'Zucchini');

University of British Columbia, Vancouver

Department of Computer Science

INSERT INTO BranchHas2(PostalCode, Province, City) VALUES

('V5L 3X2', 'BC', 'Vancouver'),
('V5P 3W2', 'BC', 'Vancouver'),
('V5V 4E7', 'BC', 'Vancouver'),
('V5R 3L9', 'BC', 'Vancouver'),
('M5V 3M2', 'ON', 'Toronto');

INSERT INTO BranchHas1(DriveThrough, Street, PostalCode, UnitNumber, Self-service, EatName)
VALUES

('0', 'Commercial Drive', 'V5L 3X2', '1128', '0', 'Sula Indian Restaurant'),
('1', 'Fraser St', 'V5V 4E7', '4064', '1', 'Tim Hortons'),
('1', 'Victoria Dr', 'V5P 3W2', '5661', '1', 'McDonalds'),
('0', 'Catherines St', 'V5P 3W2', '1128', '1', 'Starbucks'),
('0', 'Calgary St', 'V5L 3X2', '2068', '0', 'French Table');

INSERT INTO GoTo(ID, Name) VALUES

('1', 'Sula Indian Restaurant'),
('2', 'Starbucks'),
('3', 'Tim Hortons'),
('4', 'French Table'),
('5', 'McDonalds');

INSERT INTO Uses(EquipName, RecName) VALUES

('Mixers', 'Zucchini Smoothie'),
('Cutting Board', 'Grilled Tomahawk Steak'),
('Oven', 'Maple Cupcakes'),
('Spatula', 'Scrambled Eggs'),
('Spatula', 'Scotch Eggs');

INSERT INTO Own(ID, EquipName) VALUES

('1', 'Mixers'),
('2', 'Mixers'),
('3', 'Mixers'),
('4', 'Cutting Board'),
('5', 'Oven');

INSERT INTO Possess(ID, IngName) VALUES

('1', 'Zucchini'),
('2', 'Zucchini'),
('3', 'Zucchini'),
('4', 'Beef Loin'),
('5', 'Canola Oil');

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO Cook(RecName, IngName, Unit, Quantity) VALUES
('Zucchini Smoothie', 'Zucchini', 'Gram', '100'),
('Grilled Tomahawk Steak', 'Beef Loin', 'Gram', '200'),
('Maple Cupcakes', 'Canola Oil', 'TableSpoon', '3'),
('Scrambled Eggs', 'Canola Oil', 'TableSpoon', '1'),
('Scotch Eggs', 'Canola Oil', 'TableSpoon', '2');
```

```
INSERT INTO Requiere(RecName, PreserveMethod) VALUES
('Zucchini Smoothie', 'Cool storage'),
('Grilled Tomahawk Steak', 'Cool storage'),
('Maple Cupcakes', 'Cool storage'),
('Scrambled Eggs', 'Cool storage'),
('Scotch Eggs', 'Cool storage');
```