**University of British Columbia, Vancouver**
Department of Computer Science

# CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20, 2023

Group Number: 4

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---|---|---|---|
| Eric Pien | 18875864 | p9k6b | eric.hs.pien@gmail.com |
| Justin Tang | 36376796 | i0q5c | justin.tangg@gmail.com |
| Kiara Melocoton | 94810421 | u3s3n | kiaramelocoton@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

**University of British Columbia, Vancouver**
Department of Computer Science

---

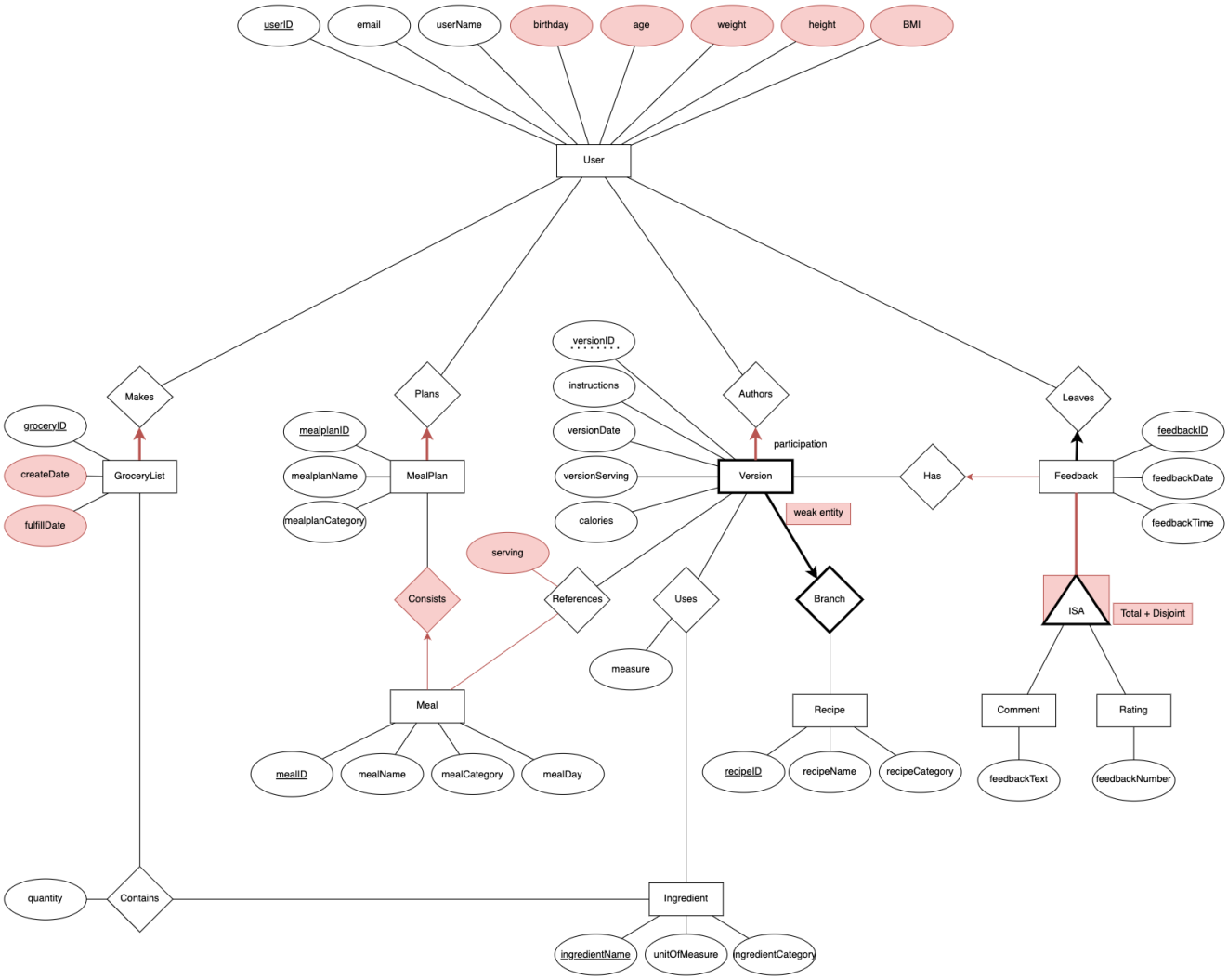
**Summary of Project**

"EatHub" is an application that will allow users to easily track their recipes and manage all aspects of the recipe creation process. The application will model recipe information, collaboration and version control, and ingredient management. The database will allow users to create, edit, clone, fork, and share, and access past versions of recipes.

**ER Diagram**

Changes (highlighted in red below)

- **Accepting TA comments**
    1. **Added attributes to GroceryList**: createDate and fulfillDate
    2. **Fixed ISA**: Comment and Rating now go through one Total and Disjoint ISA relationship.
    3. **Renamed duplicate relationship names**: MealPlan ~~Has~~ Consists (of) Meal
- **New based on Milestone 2 activities**
    1. **Add attributes to Many to Many relationships**: References (serving)
    2. **Add attributes to User**: birthday, age, weight, height, BMI
    3. **Cosmetic Changes**: adopted camelCase naming convention



**University of British Columbia, Vancouver**
Department of Computer Science

---

**Schema**

Table: User

- **Definition**: User(userID: integer, email: char[30], name: char[30], birthday: date, age: integer, weight: integer, height: integer, BMI: integer)
- **PK**: userID
- **CK**: userID, email
- **FK**: n/a
- **Constraints**:
  - email NOT NULL, UNIQUE
  - name NOT NULL

Table: Recipe

- **Definition:** Recipe(recipeID: integer, recipeName: char[30], recipeCategory: char[20])
- **PK**: recipeID
- **CK**: recipeID
- **FK**: n/a
- **Constraints**:
  - recipeName NOT NULL

Table: Version

- **Definition**: Version(**recipeID: integer,** versionID: integer, instructions: char[3000], versionDate: date, versionServing: integer, calories: integer)
- **PK**: {versionID, recipeID}
- **CK**: {versionID, recipeID}
- **FK**: recipeID
- **Constraints**:
  - versionDate NOT NULL

Table: MealPlan

- **Definition**: MealPlan(mealplanID: integer, **userID: integer,** mealplanName: char[30], mealplanCategory: char[30])
- **PK**: mealplanID
- **CK**: mealplanID
- **FK**: userID
- **Constraints**:
  - mealplanName NOT NULL
  - userID NOT NULL

Table: Meal

- **Definition**: Meal(mealID: integer, **mealplanID: integer,** mealName: char[30], mealCategory: char[30], mealDay: date)
- **PK**: mealID

---

- **CK**: mealID
- **FK**: mealplanID
- **Constraints**:
    - mealplanID NOT NULL, UNIQUE

Table: GroceryList

- **Definition**: GroceryList(grocerylistID: integer, **userID: integer**, createDate: date, fulfillDate: date)
- **PK**: grocerylistID
- **CK**: grocerylistID
- **FK**: userID
- **Constraints**:
    - createDate NOT NULL
    - userID NOT NULL

Table: Ingredient

- **Definition**: Ingredient(ingredientName: char[30], unitOfMeasure: char[10], ingredientCategory: char[20])
- **PK**: ingredientName
- **CK**: ingredientName
- **FK**: n/a
- **Constraints**:
    - unitOfMeasure NOT NULL

Table: Comment

- **Definition**: Comment(feedbackID: integer, **userID: integer**, feedbackDate: date, feedbackTime: time, feedbackText: char[400])
- **PK**: feedbackID
- **CK**: feedbackID
- **FK**: userID
- **Constraints**:
    - userID NOT NULL

Table: Rating

- **Definition**: Rating(feedbackID: integer, **userID: integer**, feedbackDate: date, feedbackTime: time, feedbackNumber: real)
- **PK**: feedbackID
- **CK**: feedbackID
- **FK**: userID
- **Constraints**:
    - feedbackNumber NOT NULL

Table: Contains

---

- **Definition**: Contains(**grocerylistID: integer**, **ingredientName: char[30],** quantity: integer)
- **PK**: {grocerylistID, ingredientName}
- **CK**: {grocerylistID, ingredientName}
- **FK**: {grocerylistID, ingredientName}
- **Constraints**:
    - quantity NOT NULL

Table: Uses

- **Definition**: Uses(**versionID: integer**, **recipeID: integer**, **ingredientName: char[30],** measure: integer)
- **PK**: {versionID, recipeID, ingredientName}
- **CK**: {versionID, recipeID, ingredientName}
- **FK**: {versionID, recipeID, ingredientName}
- **Constraints**:
    - measure NOT NULL

Table: References

- **Definition**: References(**versionID: integer**, **mealID: integer,** serving: integer)
- **PK**: {versionID, mealID}
- **CK**: {versionID, mealID}
- **FK**: {versionID, mealID}
- **Constraints**:
    - serving NOT NULL

**Functional Dependencies**

User(userID: integer, email: char[30], name: char[30], birthday: date, age: integer, weight: integer, height: integer, BMI: integer)

- userID -> email, name
- email -> userID, name
- birthday -> age
- weight, height -> BMI

Recipe(recipeID: integer, recipeName: char[30], recipeCategory: char[20])

- recipeID -> recipeName, recipeCategory

Version(**recipeID: integer,** versionID: integer, instructions: char[3000], versionDate: date, versionServing: integer, calories: integer)

- **recipeID**, versionID -> instructions, versionDate, versionServing, calories

MealPlan(mealplanID: integer, **userID: integer,** mealplanName: char[30], mealplanCategory: char[30])

- mealplanID -> **userID**, mealplanName, mealplanCategory

---

Meal(mealID: integer, **mealplanID: integer,** mealName: char[30], mealCategory: char[30], mealDay: date)
- mealID -> **mealplanID**, mealName, mealCategory, mealDay

GroceryList(grocerylistID: integer, **userID: integer**, createDate: date, fulfillDate: date)
- grocerylistID -> **userID**, createDate, fulfillDate

Ingredient(ingredientName: char[30], unitOfMeasure: char[10], ingredientCategory: char[20])
- ingredientName -> unitOfMeasure, ingredientCategory

Comment(feedbackID: integer, **userID: integer**, feedbackDate: date, feedbackTime: time, feedbackText: char[400])
- feedbackID -> **userID**, feedbackDate, feedbackTime, feedbackText

Rating(feedbackID: integer, **userID: integer**, feedbackDate: date, feedbackTime: time, feedbackNumber: real)
- feedbackID -> **userID**, feedbackDate, feedbackTime, feedbackNumber

Contains(**grocerylistID: integer**, **ingredientName: char[30],** quantity: integer)
- **grocerylistID**, **ingredientName**-> quantity

Uses(**versionID: integer**, **recipeID: integer**, **ingredientName: char[30],** measure: integer)
- **versionID**, **recipeID**, **ingredientName** -> measure

References(**versionID: integer**, **mealID: integer,** serving: integer)
- **versionID**, **mealID** -> serving

**Normalization**
For each table, we checked if the table is in BCNF and if not, we normalized below.

User(userID: integer, email: char[30], name: char[30], birthday: date, age: integer, weight: integer, height: integer, BMI: integer)
- Closures
  - userID+ = {userID, email, name, birthday, age, weight height, BMI}
  - email+ = {email, userID, name, birthday, age, weight height, BMI}
  - birthday+ = {birthday, age}
  - weight, height+ = {weight, height, BMI}
- birthday; and weight, height violates BCNF. So we decompose below.

  Decompose on birthday->age
  User1(birthday, age), User2(birthday, userID, email, name, weight, height, BMI)

---

Decompose on weight, height->BMI
User3(weight, height, BMI), User4(userID, email, name, birthday, weight, height)

- User1, User3, User4 conform to BCNF.

Recipe(recipeID: integer, recipeName: char[30], recipeCategory: char[20])
- Closures
  - recipeID+ = {recipeID, recipeName, recipeCategory}
- Recipe conforms to BCNF.

Version(**recipeID: integer,** versionID: integer, instructions: char[3000], versionDate: date, versionServing: integer, calories: integer)
- Closures
  - recipeID, versionID+ = {recipeID, versionID, instructions, versionDate, versionServing, calories}
- Version conforms to BCNF

MealPlan(mealplanID: integer, **userID: integer,** mealplanName: char[30], mealplanCategory: char[30])
- Closures
  - mealplanID+ = {mealplanID, userID, mealplanName, mealplanCategory}
- MealPlan conforms to BCNF

Meal(mealID: integer, **mealplanID: integer,** mealName: char[30], mealCategory: char[30], mealDay: date)
- Closures
  - mealID+ = {mealID, mealplanID, mealName, mealCategory, mealDay}
- Meal conforms to BCNF

GroceryList(grocerylistID: integer, **userID: integer**, createDate: date, fulfillDate: date)
- Closures
  - grocerylistID+ = {grocerylistID, userID, createDate, fulfillDate}
- GroceryList conforms to BCNF

Ingredient(ingredientName: char[30], unitOfMeasure: char[10], ingredientCategory: char[20])
- Closures
  - ingredientName+ = {ingredientName, unitOfMeasure, ingredientCategory}
- Ingredient conforms to BCNF

Comment(feedbackID: integer, **userID: integer**, feedbackDate: date, feedbackTime: time, feedbackText: char[400])
- Closures
  - feedbackID+ = {feedbackID, userID, feedbackDate, feedbackTime, feedbackText}
- Comment conforms to BCNF

---

Rating(feedbackID: integer, **userID: integer**, feedbackDate: date, feedbackTime: time, feedbackNumber: real)

- Closures
  - feedbackID+ = {feedbackID, userID, feedbackDate, feedbackTime, feedbackNumber}
- Rating conforms to BCNF

Contains(**grocerylistID: integer**, **ingredientName: char[30],** quantity: integer)

- Closures
  - grocerylistID, ingredientName+ = {grocerylistID, ingredientName, quantity}
- Contains conforms to BCNF

Uses(**versionID: integer**, **recipeID: integer**, **ingredientName: char[30],** measure: integer)

- Closures
  - versionID, recipeID, ingredientName+ = {versionID, recipeID, ingredientName, measure}
- Uses conforms to BCNF

References(**versionID: integer**, **mealID: integer,** serving: integer)

- Closures
  - versionID, mealID+ = {versionID, mealID, serving}
- References conforms to BCNF

**SQL DDL Statements**

For the tables decomposed, we modified the table to name for clarity

```
CREATE TABLE UserInfo (
     userID    INTEGER,
     email     CHAR(30)  NOT NULL UNIQUE,
     name      CHAR(30)  NOT NULL,
     birthday  DATE,
     weight    INTEGER,
`    height    INTEGER,
     PRIMARY KEY (userID)
)

CREATE TABLE UserAge (
     birthday  DATE,
     age       INTEGER,
     PRIMARY KEY (birthday)
)

CREATE TABLE UserBMI (
     weight    INTEGER,
```

```
     height    INTEGER,
     BMI       INTEGER,
     PRIMARY KEY (weight, height)
)

CREATE TABLE Recipe (
     recipeID       INTEGER,
     recipeName     CHAR(30)  NOT NULL,
     recipeCategory CHAR(20),
     PRIMARY KEY (recipeID)
)

CREATE TABLE Version (
     recipeID       INTEGER,
     versionID      INTEGER,
     instructions   CHAR(3000),
     versionDate    DATE NOT NULL,
     versionServing INTEGER,
     calories       INTEGER,
     PRIMARY KEY (versionID, recipeID),
     FOREIGN KEY(recipeID)
          REFERENCES Recipe(recipeID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE
)

CREATE TABLE MealPlan (
     mealplanID          INTEGER   NOT NULL,
     userID              INTEGER   NOT NULL,
     mealplanName        CHAR(30),
     mealplanCategory    CHAR(30),
     PRIMARY KEY (mealplanID),
     FOREIGN KEY(userID)
          REFERENCES UserInfo(userID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE
)

CREATE TABLE Meal (
     mealID         INTEGER,
     mealplanID     INTEGER   NOT NULL UNIQUE,
     mealName       CHAR(30)
     mealCategory   CHAR(30)
     mealDay        DATE,
     PRIMARY KEY (mealID),
```

```
     FOREIGN KEY(mealplaneID),
          REFERENCES MealPlanID(mealplanID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE
)

CREATE TABLE GroceryList (
     grocerylistID  INTEGER,
     userID         INTEGER   NOT NULL,
     createDate     DATE      NOT NULL,
     fulfillDate    DATE,
     PRIMARY KEY (grocerylistID),
     FOREIGN KEY(userID)
          REFERENCES UserInfo(userID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE
)

CREATE TABLE Ingredient (
     ingredientName      CHAR(30),
     unitOfMeasure       CHAR(10)  NOT NULL,
     ingredientCategory  CHAR(20),
     PRIMARY KEY (ingredientName)
)

CREATE TABLE Comment (
     feedbackID     INTEGER,
     userID         INTEGER   NOT NULL,
     feedbackDate   DATE,
     feedbackTime   TIME,
     feedbackText   CHAR(400),
     PRIMARY KEY (feedbackID)
)

CREATE TABLE Rating (
     feedbackID     INTEGER,
     userID         INTEGER,
     feedbackDate   DATE,
     feedbackTime   TIME,
     feedbackNumber INTEGER   NOT NULL,
     PRIMARY KEY (feedbackID)
)

CREATE TABLE Contains (
     grocerylistID  INTEGER,
```

```
     ingredientName CHAR(30)
     quantity       INTEGER   NOT NULL,
     PRIMARY KEY (grocerylistID, ingredientName),
     FOREIGN KEY (grocerylistID)
          REFERENCES GroceryList(grocerylistID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE,
     FOREIGN KEY (ingredientName)
          REFERENCES Ingredient(ingredientName)
               ON DELETE CASCADE
                    ON UPDATE CASCADE
)

CREATE TABLE Uses (
     versionID      INTEGER,
     recipeID       INTEGER,
     ingredientName CHAR(30),
     measure        INTEGER   NOT NULL,
     PRIMARY KEY (versionID, recipeID, ingredientName),
     FOREIGN KEY (versionID)
          REFERENCES Version(versionID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE,
     FOREIGN KEY (recipeID)
          REFERENCES Ingredient(recipeID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE,
     FOREIGN KEY (ingredientName)
          REFERENCES Ingredient(ingredientName)
               ON DELETE CASCADE
                    ON UPDATE CASCADE
)

CREATE TABLE References (
     versionID      INTEGER,
     mealID         INTEGER,
     serving        INTEGER   NOT NULL,
     PRIMARY KEY (versionID, mealID),
     FOREIGN KEY (versionID)
          REFERENCES Version(versionID)
               ON DELETE CASCADE
                    ON UPDATE CASCADE,
     FOREIGN KEY (mealID)
          REFERENCES Meal(mealID)
               ON DELETE CASCADE
```

```
                    ON UPDATE CASCADE
)
```

**INSERT**

```
INSERT INTO UserInfo (userID, email, name, birthday, weight,
height)
VALUES
(1, 'tony.stark@gmail.com', 'Tony Stark', '1980-02-29', 220,
190),
(2, 'diana.prince@gmail.com', 'Diana Prince', '1985-10-25', 150,
180),
(3, 'bruce.wayne@gmail.com', 'Bruce Wayne, '1975-03-30', 180,
188),
(4, 'peter.parker@gmail.com', 'Peter Parker, '1995-08-15', 167,
178),
(5, 'miles.morales@gmail.com', 'Miles Morales, '2001-11-12',
160, 175);


INSERT INTO UserAge (birthday, age)
VALUES
('1980-02-29', 53),
('1985-10-25', 38),
('1975-03-30', 48),
('1995-08-15', 28),
('2001-11-12', 22);

INSERT INTO UserBMI (weight, height, BMI)
VALUES
(220, 190, 24),
(150, 180, 21),
(180, 188, 27),
(167, 178, 24),
(160, 175, 25);

INSERT INTO Recipe (recipeID, recipeName, recipeCategory)
VALUES
(1, 'Spaghetti Bolognese', 'Pasta'),
(2, 'Chicken Stir-Fry', 'Asian'),
(3, 'Caprese Salad', 'Salad'),
(4, 'Beef Tacos', 'Mexican'),
(5, 'Mushroom Risotto', 'Italian');
```

```
INSERT INTO Version (recipeID, versionID, instructions,
versionDate, versionServing, calories)
VALUES
(2, 1, 'Heat oil in a wok. Add chicken and stir-fry for 5
minutes. Add vegetables and sauce. Stir-fry for an additional 3
minutes.', '2023-10-20', 4, 350),
(2, 2, 'Marinate chicken in soy sauce and garlic for 30 minutes
before stir-frying. Add broccoli and peppers for added flavor.',
'2023-10-20', 4, 370),
(2, 3, 'For a spicier version, add chili flakes and ginger while
stir-frying the chicken. Serve with steamed rice.',
'2023-10-20', 4, 400),
(2, 4, 'Use low-sodium soy sauce and olive oil for a healthier
option. Include sliced carrots and snap peas for added crunch.',
'2023-10-20', 4, 330),
(2, 5, 'Make it gluten-free by using tamari sauce. Add water
chestnuts and baby corn for a unique twist.', '2023-10-20', 4,
360);

INSERT INTO MealPlan (mealplanID, userID, mealplanName,
mealplanCategory)
VALUES
(3, 1, 'Family Dinners', 'Homestyle'),
(4, 2, 'Gluten-Free Week', 'Gluten-Free'),
(5, 1, 'Mediterranean Feast', 'Mediterranean'),
(6, 2, 'Vegan Challenge', 'Vegan'),
(7, 1, 'Asian Fusion', 'Asian');

INSERT INTO Meal (mealID, mealplanID, mealName, mealCategory,
mealDay)
VALUES
(1, 6, 'Breakfast Burrito', 'Breakfast', '2023-10-21'),
(2, 6, 'Quinoa Salad', 'Salad', '2023-10-22'),
(3, 6, 'Veggie Stir-Fry', 'Asian', '2023-10-23'),
(4, 6, 'Tofu Scramble', 'Breakfast', '2023-10-24'),
(5, 1, 'Sushi Rolls', 'Asian', '2023-10-25');


INSERT INTO GroceryList (grocerylistID, userID, createDate,
fulfillDate)
VALUES
(1, 1, '2023-10-20', '2023-10-27'),
(2, 1, '2023-10-20', '2023-10-31'),
(3, 1, '2023-10-20', '2023-11-03'),
```

```
(4, 1, '2023-10-20', '2023-11-07'),
(5, 1, '2023-10-20', '2023-11-10');


INSERT INTO Ingredient (ingredientName, unitOfMeasure,
ingredientCategory)
VALUES
('Chicken', 'Pound', 'Meat'),
('Broccoli', 'Ounce', 'Vegetable'),
('Soy Sauce', 'Tablespoon', 'Sauce'),
('Rice', 'Cup', 'Grain'),
('Garlic', 'Cloves', 'Vegetable');

INSERT INTO Comment (feedbackID, userID, feedbackDate,
feedbackTime, feedbackText)
VALUES
(1, 1, '2023-10-20', '14:30:00', 'Great recipe! I loved it.'),
(2, 1, '2023-10-21', '09:45:00', 'This meal plan is
fantastic.'),
(3, 1, '2023-10-22', '18:15:00', 'It tasted kind of bland.'),
(4, 1, '2023-10-23', '12:20:00', 'The grocery list was very
helpful.'),
(5, 1, '2023-10-24', '15:55:00', 'The Chicken Stir-Fry was
amazing!');

INSERT INTO Rating (feedbackID, userID, feedbackDate,
feedbackTime, feedbackNumber)
VALUES
(6, 1, '2023-10-20', '14:30:00', 5),
(7, 2, '2023-10-21', '09:45:00', 4),
(8, 1, '2023-10-22', '18:15:00', 3),
(9, 2, '2023-10-23', '12:20:00', 4),
(10, 1, '2023-10-24', '15:55:00', 5);

INSERT INTO Contains (grocerylistID, ingredientName, quantity)
VALUES
(1, 'Chicken', 1),
(1, 'Broccoli', 2),
(1, 'Soy Sauce', 1),
(1, 'Rice', 2),
(1, 'Garlic', 1);


INSERT INTO Uses (versionID, recipeID, ingredientName, measure)
VALUES
```

```
(1, 4, 'Ground Beef', 1),
(1, 4, 'Taco Shells', 8),
(1, 4, 'Lettuce', 1),
(1, 4, 'Tomato', 1),
(1, 4, 'Cheddar Cheese', 1),
(1, 4, 'Sour Cream', 1),
(1, 4, 'Taco Seasoning', 1);

INSERT INTO References (versionID, mealID, serving)
VALUES
(1, 2, 1),
(1, 2, 4),
(1, 2, 3),
(1, 2, 2),
(1, 2, 1);
```