Milestone 2 - Project Outline | CIS 550

Yuxi Dai, Zhihang Yuan, Yuxuan Li, Yuyan Zhou| 2022-03-27

1. Motivation for the idea/description of the problem the application solves
   We aim at providing a recipe recommendation system for users. Users are able to search for the recipe by using the name(keywords), ingredients, author, and many other interesting ways. As for a typical recipe, a user can view its basic information such as name, author, as well as cooking method in detail. As a recommendation system, we also include reviews for recipes. A user can leave comments and rate the recipe. We support customized ways for users to get recipe information and find those which fit their needs. We will also show a radar plot for the main dishes' nutritions, one can better know his/her daily nutrition.
2. List of features you will definitely implement in the application
   a. Main dishes
      i. Search for the related recipe according to ingredients
         1. Support partial words matching

      ii. Filter for recipes according to cooking time.
         1. Rank time span
         2. Slide bar

      iii. Show the radar plot of the nutritions
         1. Nutritions include calories (#), total fat (PDV), sugar (PDV), sodium (PDV), protein (PDV), saturated fat (PDV), and carbohydrates (PDV))
      iv. Show the number of steps of the recipes
         1. Filter for recipes according to a number of steps
            a. Slide bar to filter the rating
         2. Rank steps functionality

      v. Ratings
         1. Filter rating
            a. Slide bar to filter the rating
         2. Rank rating functionality
      vi. Comment
         1. Search by keywords

   b. Pastry dataset
      i. Search for the related recipe according to the recipe name
      The following will be displayed

            1. Recipe name, authors, ratings, total time, cook time, prepare time
            2. The total number of reviews
        ii. Total time
            1. Slide bar to filter the time
            2. Rank functionality
        iii. Ratings
            1. Slide bar to filter the ratings
            2. Rank functionality
        iv. Pictures
            1. Display the picture of the recipe.

3. List of features you might implement in the application, given enough time
    a. Show several reviews with the highest ratings, lowest ratings for selected recipes.

    b. Generate all the user names randomly to fill in the dataset. (Since we only access the user ID. Adding the user names will make it more meaningful to users who use our system.)

    c. User page, which contains all the users who have commented on some recipes.

4. List of pages the application will have and a 1-2 sentence description of each page. We expect that the functionality of each page will be meaningfully different than the
functionality of the other pages.
    a. Main dish page
        i. Can search and filter recipes using ingredients, number of steps, cook time, and rating.
        ii. Can show radar plot of nutritions for the selected recipe.
        iii. Can sort recipes by a number of steps, cook time, and rating.

    b. Pastry page
        i. Can search and filter recipes using recipe name, total time, cook time, prepare time, the total number of reviews, and rating.
        ii. Can show the picture of the selected recipe.

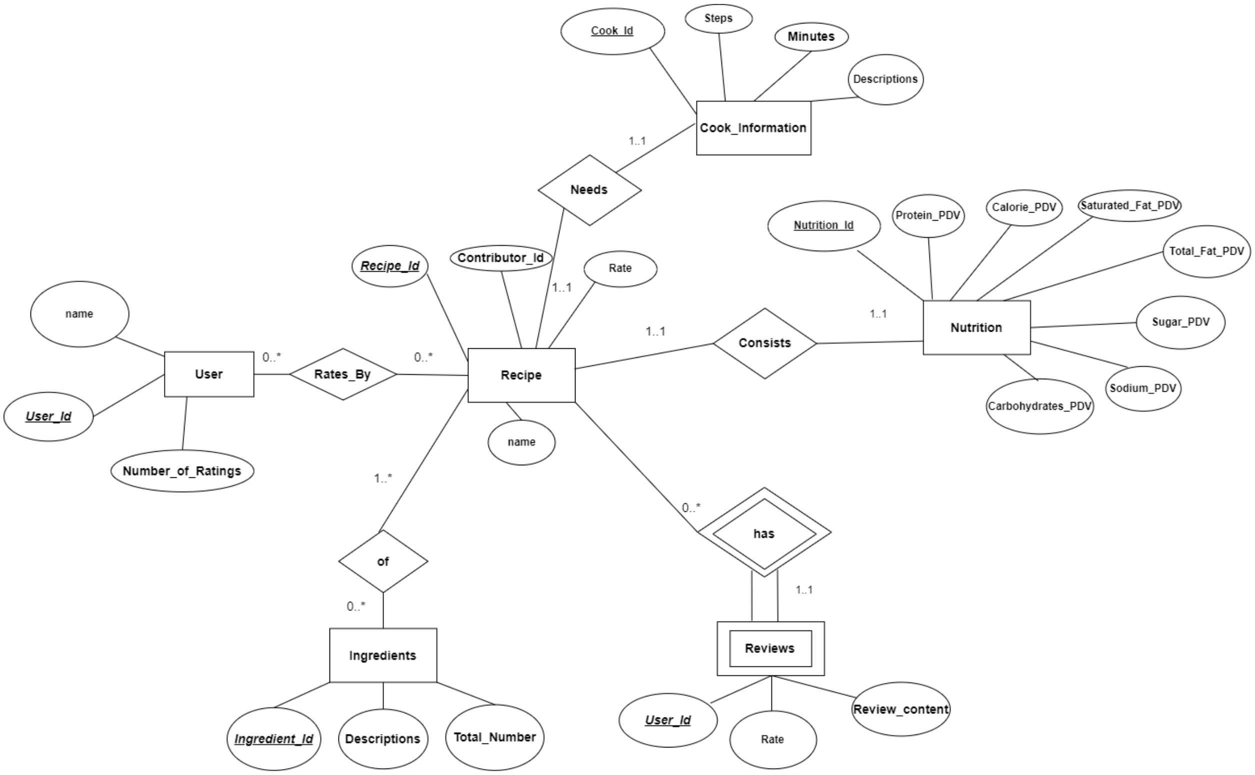
    c. Review page
        i. We can click a recipe and show all the reviews on the page.
        ii. We can also track the reviewer to see what other reviews he/she made.
        iii. We can sort the rating if any, given the reviews.

    d. User page (optional)

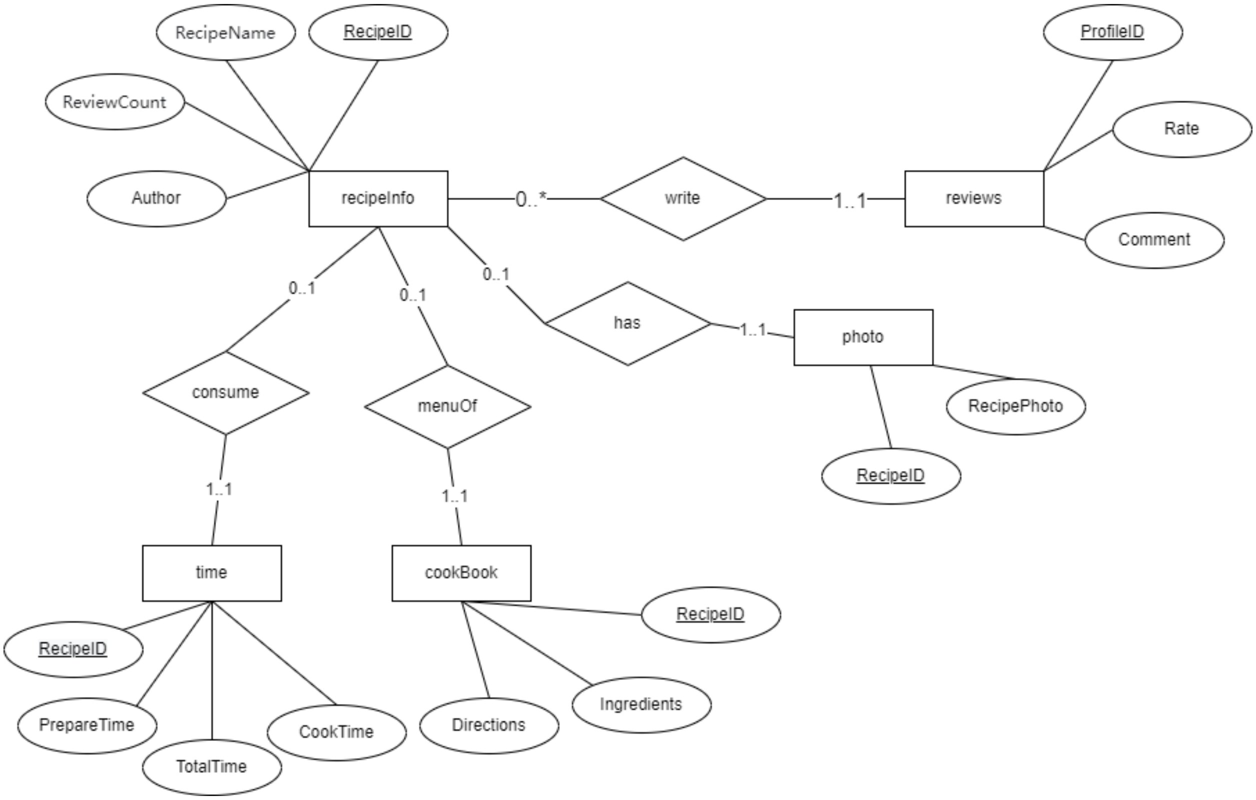
i. This page contains all the user information including the contributor and reviewer.

ii. We may track the user to see what they commented or contributed.

5. Relational schema as an ER diagram



Dataset A - Main dishes



Dataset B - Pastry dataset

6. SQL DDL for creating the database

For dataset A - Main dishes

```sql
CREATE TABLE Recipe (
  Recipe_Id INT NOT NULL,
  Contributor_Id INT,
  Rate FLOAT,
  Name VARCHAR (50),
  PRIMARY KEY (Recipe_Id)
);

CREATE TABLE Nutrition (
  Nutrition_Id INT,
  Calorie_PDV DECIMAL (5,2),
  Total_Fat_PDV DECIMAL (5,2),
  Sugar_PDV DECIMAL (5,2),
  Protein_PDV DECIMAL (5,2),
  Sodium_PDV DECIMAL (5,2),
  Saturated_Fat_PDV DECIMAL (5,2),
  Carbohydrates_PDV DECIMAL (5,2),
  PRIMARY KEY(Nutrition_Id),
  FOREIGN KEY (Nutrition_Id) REFERENCES Recipe (Recipe_Id)
);


CREATE TABLE Cook_information (
  Cook_Id INT,
  Minutes INT,
  Steps VARCHAR (500),
  Descriptions VARCHAR (500),
  PRIMARY KEY(Cook_Id),
  FOREIGN KEY (Cook_Id) REFERENCES Recipe (Recipe_Id)
);

CREATE TABLE Review_Belong_To (
  Recipe_Id INT,
  User_Id INT,
  Rate FLOAT,
  Review_Content VARCHAR (500),
  PRIMARY KEY (Recipe_Id, User_Id),
  FOREIGN KEY(Recipe_Id) REFERENCES Recipe (Recipe_Id)
);

CREATE TABLE USER (
  User_Id INT,
  Name VARCHAR (50),
  Number_of_Ratings INT,
  PRIMARY KEY(User_Id)
);

CREATE TABLE Ingredients (
  Ingredient_Id INT,
  Descriptions VARCHAR (500),
  Value DECIMAL (5,2),
  PRIMARY KEY (Ingredient_Id)
  FOREIGN KEY (Ingredient_Id) REFERENCES Recipe (Recipe_Id)
);
```

Dataset B - Pastry dataset

```
CREATE TABLE RecipeInfo
(
   RecipeID    INT,
   RecipeName  VARCHAR(255),
   ReviewCount INT,
   Author      VARCHAR(20),
   PRIMARY KEY (RecipeID)
);

CREATE TABLE Photo
(
   RecipeID    INT,
   RecipePhoto VARCHAR(255),
   PRIMARY KEY (RecipeID),
   FOREIGN KEY (RecipeID) REFERENCES RecipeInfo (RecipeID)
);

CREATE TABLE CookingTime
(
   RecipeID    INT,
   PrepareTime FLOAT,
   CookTime    FLOAT,
   TotalTime   FLOAT,
   PRIMARY KEY (RecipeID),
   FOREIGN KEY (RecipeID) REFERENCES RecipeInfo (RecipeID)
);

CREATE TABLE CookBook
(
   RecipeID    INT,
   Ingredients VARCHAR(255),
   Directions  VARCHAR(500),
   PRIMARY KEY (RecipeID),
   FOREIGN KEY (RecipeID) REFERENCES RecipeInfo (RecipeID)
);

CREATE TABLE Reviews
(
   RecipeID  INT,
   ProfileID INT,
   Rate      FLOAT,
   Comment   CHAR(255),
   PRIMARY KEY (RecipeID),
   FOREIGN KEY (RecipeID) REFERENCES RecipeInfo (RecipeID)
);
```

7. Explanation of how you will clean and pre-process the data.

   a. For the dataset - Main dishes

      1. Delete some anomalous symbol and duplicate rows in the original data ‘Calories’ like ‘[‘and unify data types for individual nutrients.

      2. Delete useless attributes (columns in the original csv file that were not used for the later database design) for the original data during the pre-processing process like ‘Contributor_id’ and ‘nutrition’ in the ‘Raw_recipes.csv’.

      3. Classify the recipes into some self-defined categories and enable more target recommendations (including cook information, nutrition, etc.)

      4. For different data files (RAW_RECIPES.csv and PP_RECIPES.csv as an example), create some merged versions of the recipes data (where rows refer to a unique recipe) with only the columns we care about. Rename the ID column to specifically state that it is refers to a recipe ID, which will be useful when merging data between data frames.

      5. For the original ‘ingre_map.pkl’. We split the term ‘ingredient_ids’ with respect to the ‘name_tokens’ and the ‘recipe_id’. After the process, Ingredients IDs are in their own column and can be grouped by and check the quantity of the recipes that each belongs to.

   b. For the dataset - Pastry dataset

      We will use python to pre-process the data. We eliminate the rows where the recipe steps/recipe descriptions are null, which means, we will ignore instances which does not contain the description of the recipe. (Since our focus is on the recipe steps/descriptions).

      More specifically, we split the two big tables - Recipe & Reviews into 5 separate smaller tables, the details of which are listed above in the ER diagram or SQL DDL statements.

8. List of technologies you will use. You must use some kind of SQL database. We recommend using MySQL or Oracle specifically because you will use MySQL in HW2, and we will provide guidance for setting up a MySQL database. Some groups in the past have had issues with MySQL, but Oracle is another option.

   a. React
   b. MySQL
   c. Python