

Project Outline: YumLog

1. Motivation and Problem Statement

1.1 The Problem

In the modern culinary landscape, users struggle with:

- Finding restaurants that align with dietary needs, location, and budget.
- Recreating restaurant meals at home due to lack of structured recipes.

Traditional review platforms often include unverified or noisy feedback, and few tools exist that bridge restaurant discovery and home-cooking guidance.

1.2 YumLog's Solution

YumLog is an AI-powered food discovery and recipe platform that:

- Provides personalized restaurant and dish recommendations.
- Matches restaurant dishes with structured at-home recipes.
- Offers direct ingredient procurement links.
- Enables community interaction through food discussions and reviews.

2. Features Overview

2.1 Features We Will Implement

Restaurant Discovery & Recommendations

Users can perform detailed, multi-criteria searches for restaurants based on:

- **Location:** Filter results by city, neighborhood, or proximity to find nearby dining options.
- **Cuisine & Dietary Preferences:** Choose from various cuisine types and dietary categories (e.g., vegan, gluten-free, halal, Italian).
- **Pricing Tiers:** Narrow down options based on price levels (e.g., \$, \$\$, \$\$\$).

User Reviews & Ratings

Enables users to share their experiences by:

- Giving star ratings (1–5) and writing descriptive reviews.

- Viewing reviews on each restaurant's detail page, sorted by relevance or recency.
- Upvoting helpful reviews to surface the most useful content to others.

AI-Powered Meal Suggestions

Our system provides personalized restaurant and dish recommendations using:

- Users' past interactions (clicks, reviews, favorites, search histories).
- Similar users' preferences through collaborative filtering.
- Dishes highly rated by people with similar profiles.

Recipes and Dishes Combination

We match restaurant menu items with home-cooking recipes:

- **Dish to Recipe:** A user selects a restaurant dish, and the system finds similar recipes.
- **Recipe to Dish:** A user selects a recipe and finds restaurants offering similar dishes.

Community Engagement & Social Features

Facilitates social interaction through:

- **User Profiles:** Display user reviews, photos, and follower counts.
- **Comments and Likes:** Engage with others' reviews, food photos, or posts.
- **Polls:** Ask the community for food recommendations (e.g., "Best ramen in town?").

2.2 Optional (Stretch) Features

Restaurant Discovery

- **Special Filters:** Add "Open Now", "Outdoor Seating", or "Family Friendly" as options.

Reviews

- **Photo Uploads:** Let users attach food pictures to their reviews.
- **Visual Sentiment Tags:** Auto-tag reviews with descriptive labels like "Spicy", "Budget-Friendly", or "Great for Groups".

Meal Suggestions

- **Taste Profile Embedding:** Learn user preferences over time using machine learning.
- **Explainable Recommendations:** Add context like "Because you liked X, we suggest Y."
- **Real-Time Feedback Loop:** Let users mark dishes as "Tried" or "Wishlist" to improve personalization.
- **Mood or Occasion-Based Filters:** Users select context (e.g., "Quick Lunch") to get dynamic suggestions.

Recipes

- **Recipe Generation:** If no recipe matches a dish, generate one using AI based on the dish name and context.

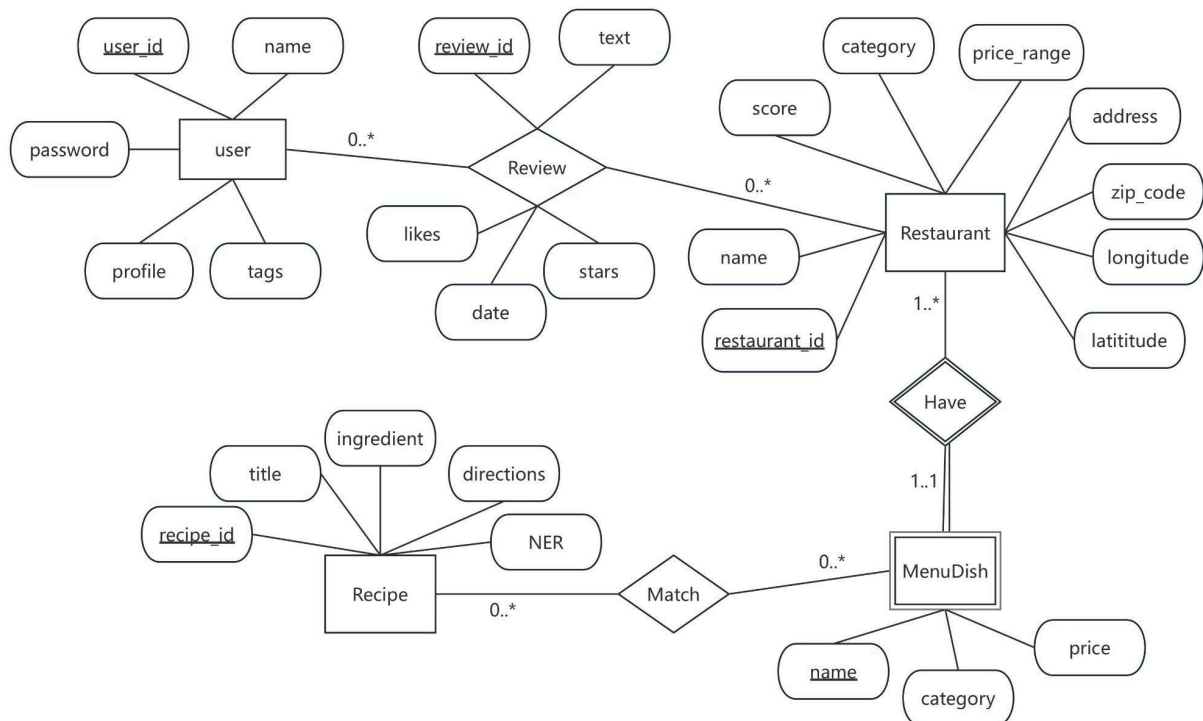
Community

- **Topic-Based Forums:** Users can form groups around cuisines, cities, or dietary goals.
- **Cooking Contests:** Monthly or seasonal contests that users can join and share results.

3. Application Pages

- **Homepage** – Featured restaurants, popular dishes, and search bar.
- **Discovery Page** – Nearby restaurant list with filters and map view.
- **Restaurant Detail Page** – Reviews, info, bookmarks, and photos.
- **Profile Page** – User reviews, photos, and follower stats.
- **Community Page** – Posts, comments, and polls.
- **Recipe & Dish Match Page** – Dish-to-recipe matching interface.
- **Smart Recommendation Page** – Personalized suggestions based on mood or history.

4. ER Diagram



5. SQL DDL

- **Restaurants Table** – Stores restaurant details and menu information.
- **Users Table** – Tracks user preferences, search history, and reviews.
- **Recipes Table** – Stores structured cooking recipes.
- **MenuDishes Table** – Lists available menu dishes linked to restaurants.
- **Reviews Table** – Contains user-generated restaurant reviews.
- **Match Table** – Contains relationship between recipes and menu dishes.

```
CREATE TABLE Restaurants (  
    restaurant_id INT PRIMARY KEY,  
    name VARCHAR(255),  
    address TEXT,  
    zip_code VARCHAR(20),  
    latitude FLOAT,  
    longitude FLOAT,  
    score FLOAT CHECK (score BETWEEN 0 AND 5),  
    category TEXT,  
    price_range VARCHAR(3) CHECK (price_range LIKE '%$%')  
);
```

```
CREATE TABLE Users (  
    user_id VARCHAR(50) PRIMARY KEY,  
    name VARCHAR(100),  
    password VARCHAR(50),  
    tags TEXT,  
    profile TEXT  
);
```

```
CREATE TABLE Reviews (  
    review_id INT PRIMARY KEY,  
    user_id VARCHAR(50) REFERENCES Users(user_id),  
    restaurant_id INT REFERENCES Restaurants(restaurant_id),  
    stars INT CHECK (stars BETWEEN 1 AND 5),
```

```

text TEXT,
likes INT,
date TIMESTAMP
);

CREATE TABLE MenuDishes (
    restaurant_id INT REFERENCES Restaurants(restaurant_id),
    name VARCHAR(100),
    category TEXT,
    price FLOAT,
    PRIMARY KEY (restaurant_id, name)
);

CREATE TABLE Recipes (
    recipe_id VARCHAR(50) PRIMARY KEY,
    title VARCHAR(100),
    ingredient TEXT,
    directions TEXT,
    NER TEXT
);

CREATE TABLE Match (
    recipe_id VARCHAR(50) REFERENCES Recipes(recipe_id),
    restaurant_id INT,
    dish_name VARCHAR(100),
    PRIMARY KEY(recipe_id, restaurant_id, dish_name),
    FOREIGN KEY(restaurant_id, dish_name) REFERENCES MenuDishes(restaurant_id, name)
);

```

6. Data Cleaning & Preprocessing

6.1 Missing Values

- Drop or impute null entries in restaurant and recipe data.

6.2 Text Normalization & Tokenization

- Clean review text and instructions (lowercase, trim, remove noise).
- Tokenize for analysis and entity recognition.

6.3 Entity Resolution

- Use fuzzy NLP matching to align dishes with recipes.

6.4 Outlier Detection

- Remove abnormal reviews or noisy recipes.

6.5 Standardization & Engineering

- Normalize cuisines/ingredients.
- Create location-based features.
- Embed taste profiles and popularity metrics.

7. Technology Stack

- **Backend:** Python (FastAPI)
- **Database:** PostgreSQL
- **Frontend:** React.js + Tailwind CSS
- **APIs:** Google Places API, RecipeNLG Dataset, Uber Eats Dataset
- **AI/NLP:** spaCy, scikit-learn, Hugging Face, ChatGPT
- **Deployment:** GitHub, AWS

8. Team Responsibilities

Member	Role
Jiayang Chen	Database schema, SQL, GitHub setup
Yongyin Yang	Backend development, recommendation logic
Yongqi Zhong	Frontend design, search UI
Shutong Jiang	Data processing, NLP & entity matching