RUEating: A Food Truck Discovery and Recommendation Platform 🌮

Group members: Shang-Yi Lin, Lin Guo, Ananya Chepuri, Wenliang Ding
Date: September 20, 2025

1. Introduction

1.1 Background

Food trucks are popular in many cities because they bring different types of food to busy areas. Unlike restaurants, trucks move around and change their menus often. This makes it hard for people to know where the trucks are or what food they are serving. Many apps and websites do not update fast enough, so the information is often wrong. This also results in users missing out on local food options and food truck owners struggling to reach a consistent audience. A simple system that uses good data management can solve this problem by giving real-time updates to both truck owners and customers.

1.2 Project Questions

Our project will try to answer these main questions:

- How can we keep track of truck locations that change many times in one day?
- What is the best way to store and update menus so they stay correct?
- How can reviews be collected, checked, and shown clearly?
- What type of database can handle location data and still be fast and easy to use?

2. Problem Statement

Food trucks move from place to place and often change their menus, so keeping information correct is a challenge. Many existing websites or apps are designed for fixed restaurants, not for mobile vendors. As a result, the data about food trucks is often outdated, incomplete, or missing key details like operating hours and current location. This causes problems for both sides of the market.

From the owners' side, they need a simple way to share their current position, hours, and menu with customers. Without such a tool, they may lose potential sales because people cannot find them or do not know what food is available. Truck owners also benefit from knowing how customers react through ratings and reviews, but they need these reviews to be clear and fair rather than filled with spam or irrelevant comments.

From the customers' side, people want to locate trucks near them quickly, check if the truck is open, and look at the most recent menu before deciding what to eat. They also want reviews they can trust, written by real users who have tried the food. If the information is wrong or hard to find, customers waste time walking to trucks that are closed or no longer at that spot.

Without a good data system to connect these needs, the entire process stays confusing. Owners may lose income, customers lose trust in the platform, and the overall food truck community becomes harder to grow. The core goal is to build a robust database-driven platform that handles dynamic data, such as locations and complex relational/embedded data, such as menus and reviews, to provide meaningful real-time recommendations.

3. Methodology

Our project uses a **data management approach**:

- **Database Choice**: We use MongoDB because it supports flexible data, menus with many levels, and location searches.
- **Front-end:** HTML/CSS/JavaScript and Streamlit to build a dynamic, map-based user interface for easy viewing and interaction.
- **Back-End:** Node.js for a high-performance runtime environment.
- **Data Model**: Collections include trucks, locations, menus, reviews, and users. Menus are saved with versions, so old ones can still be seen. Reviews include status and simple voting to rate their quality.
- **Analytics**: ETL pipelines make reports and charts, like heatmaps showing busy times and ratings by cuisine type.

4. Target Users

- **Food Truck Owners**: Can update menus and share current location. They also get customer reviews.
- **Customers**: Can search for trucks nearby, view menus, and leave reviews.
- **City Planners/Analysts**: Can study trends with anonymous data about food trucks.

5. Application Goals and Features

- **Real-Time Search**: Find trucks within a chosen distance, filter by cuisine, and check if they are open now.
- **Deliver Actionable Recommendations:** Move beyond simple search by providing intelligent recommendations based on proximity, price, and quality.
- **Enable Real-Time Data Management:** Allow food truck owners to instantly update critical data, especially current location and operating status.
- **Menu Updates**: Owners can upload and change menus with version history.

- **Reviews**: Customers leave ratings and comments. Simple filters stop spam.
- **Reports**: Create charts showing popular cuisines, busiest times, and truck activity.
- **Scalability**: System supports many updates and searches without slowing down.

6. Justification for Chosen Database Type

MongoDB is best for this project because:

- **Flexible Data**: Menus can look very different from one truck to another. MongoDB can store this without needing fixed tables.
- **Location Queries**: MongoDB has built-in tools to search by distance (“near me”).
- **Fast Updates**: Location changes happen many times daily. TTL removes old data automatically.
- **Simple Versioning**: Menus and reviews can be saved as documents with past versions.

Other Database Option

- **PostgreSQL + PostGIS**: A relational option with strong SQL and geospatial features. Better for strict schemas, but less flexible for nested menu data compared to MongoDB.

7. Development plan

Phase 1: Foundation

- **Goal:** Establish the database, core schema, and demonstrate basic data input
- **Key Tasks:**
    - Set up MongoDB (database instance and connection).
    - Design and implement the core schema
    - Develop the Owner Portal to create new trucks and update their real-time location.

Phase 2: Consumer Discovery & Basic Output

- **Goal:** Build the primary user interface and enable initial location-based searching.
- **Key Tasks:**
    - Develop front-end code to detect and use the user's current location.
    - Implement geospatial queries (e.g., MongoDB's `$near`) to find nearby food trucks.
    - Integrate a map service to display an Interactive Map View of open trucks.
    - Build the Detailed Truck Profile page to display a truck's menu and status.

Phase 3: Recommendation Engine & Advanced Features

- **Goal:** Implement the complex logic for ratings, reviews, and the full recommendation system.
- **Key Tasks:**

    - Develop the Review System (user input and saving to the database).
    - Implement the Real-Time Aggregate Updater logic to automatically calculate and store the `average_rating` on the `FoodTruck` document.
    - Integrate the Weighted Recommendation Logic: filter by price, then sort by the calculated `average_rating`.

Phase 4: Final Polish & Submission

- **Goal:** Test thoroughly, complete documentation, and finalize the application for the final demo.
- **Key Tasks:**
    - Perform comprehensive end-to-end Testing and bug fixing.
    - Finalize and Comment the Codebase for clarity.
    - Prepare the Final Database Schema documentation.