Team 4 (Food-Truck-Finder)

1. **Nam Ton**
   - **Student ID**: 028268680
   - **Email**: Nam.Ton01@student.csulb.edu
2. **Glen Lee**
   - **Student ID**: 027083106
   - **Email**: Glen.lee@student.csulb.edu
3. **Nicholas Andrade**
   - **Student ID**: 028433936
   - **Email**: Nicholas.Andrade01@student.csulb.edu
4. **Adam Hernandez**
   - **Student ID**: 027904069
   - **Email**:Adam.Hernandez02@student.csulb.edu
5. **Kumara Srivatsa Kondapalli**
   - **Student ID**: 028490187
   - **Email**: kumarasrivatsa.kondapalli01@student.csulb.edu
6. **Tom Nguyen**
   - **Student ID**: 028136561
   - **Email**: Tom.Nguyen01@student.csulb.edu

Preface

Version	Date	Changes
1.0	Oct 24, 2023	- Basic project setup - Initial map implementation - Basic UI components - Food truck data model - Search component structure
1.1	Oct 2, 2023	- Map indicator implementation - Locate me feature started - Star rating system research - Time implementation for food trucks - Cuisine data structure
2.0	Oct 31, 2023	- Drag and drop location feature - Review system implementation - Cuisine filter dropdown - Distance Calculation - Food Truck owner features
2.1	Nov 1 – Nov 15	- Bug Fixes and improvements - Owner/user mode implementation - Enhanced review system - Real-time updates

Purpose

This document serves as a comprehensive guide for the development and comprehension of the software project entitled "Food-Truck-Finder."

Audience

This document is intended for the users who want to use this app but need more context on what features we have and why they should use it and potential stakeholders/investors. (Specifically people who typically use food ordering apps or food review apps)

Introduction

(Introduce the software project, its goals, and the problem it aims to solve)

Project Overview

"Food Truck Finder" is a web-based application designed to connect food truck vendors with potential customers. The system enables real-time location tracking of food trucks, provides detailed menu information, and facilitates customer reviews and ratings.

Project Goals

- Enable food truck owners to broadcast their location and menu information
- Help customers easily locate and discover food trucks in their vicinity
- Provide a reliable rating and review system for customer feedback
- Streamline the food truck discovery process through filtering and search capabilities

Glossary

- (Define key terms and acronyms used throughout your document, unless they are commonly known to each possible stakeholder (e.g., "Cell phone") AND used with their common meaning. Do not expect your stakeholders to be experts. If in doubt, define a term.)
- **UI**: User Interface, the visual elements users interact with
- **Next.js:** A React framework for building web applications
- **React Leaflet:** A library for implementing interactive maps

User Requirements and Use Cases

(Outline what the system must do from the user's perspective. User stories need to use the format discussed in class and on our slides. Use cases provide detailed scenarios of system interactions.)

User Stories (10)

(A collection of user stories that apply to the project.)

1. As a person craving Chinese food, I want to quickly find open Chinese food trucks near my current location, so that I can satisfy my late-night hunger without wasting time on closed or far-away options after my long hospital shift.
2. As a food truck owner, I want to be able to broadcast my live location easily so that my customers can quickly find me when I set up in a new spot.
3. As a customer, I want to rate and review food trucks I've visited, so that I can help other customers make informed decisions about where to eat. **(Use Case)**
4. As a busy office worker, I want to see the current wait times and busyness levels of food trucks, so that I can plan my lunch break efficiently.
5. As a food enthusiast, I want to browse food trucks by cuisine type, so that I can explore different food options in my area.
6. As a customer with dietary restrictions, I want to filter food trucks by cuisine type and view their menus, so that I can find options that meet my dietary needs.
7. As a food truck owner, I want to be able to delete my food truck listing from the system, so that customers won't see my business when I'm no longer operating.
8. As a hungry customer, I want to see a clear separation between open and closed food trucks, so that I don't waste time looking at food trucks that aren't currently serving.
9. As a hungry person who forgot to pack lunch, I want to find a nearby food truck I can go to with my coworkers so that I can relieve my hunger without having to travel too far and quickly find an open spot.
10. As a computer science college student who's a picky eater, I want to be able to know the rating of the food trucks so I can avoid the hassle of having to take a risk on choosing a food truck that could serve poor-quality food.

Use Case: Rate and Review Food Trucks

Identifier	Rate and Review Food Trucks
Purpose	Allow users to rate and review a food truck they have visited
Requirements	• Users must be able to submit a rating (1-5 stars) for a food truck • Users must be able to write a text review for a food truck

	• The system must update the food truck's average rating based on new reviews
Development Risks	• Ensuring the integrity of the review system against fake or malicious reviews • Handling potential high volumes of concurrent reviews during peak times
Pre-conditions	• User is logged in / on the application • User has selected a specific food truck to review
Post-conditions	• New review is saved in the system • Food truck's average rating is updated • Review is displayed on the food truck's page

Seq#	Actor's Action	System's Response
1	User selects "Write a Review" option for a food truck	System displays review form with rating options and text field
2	User selects a star rating (1-5) and enters review text	System updates the visual representation of the star rating and tracks the text input
3	User clicks "Submit Review" button	System validates input, saves the review to the database, recalculates the food truck's average rating, and displays a confirmation message with the updated review and rating

*Table **2**: Alternate Course of Action*

Seq#	Actor's Action	System's Response
1	User selects "Write a Review" option for a food truck	System displays review form with rating options and text field
2	User selects a star rating but leaves the review text empty	System updates the visual representation of the star rating
3	User clicks "Submit Review" button	System detects empty review text and prompts user to enter review text or confirm submission of rating only
4	User confirms submission of rating only	System saves the rating to the database, recalculates the food truck's

		average rating, and displays a confirmation message with the updated rating

*Table **3**: Exceptional Course of Action*

Seq#	Actor's Action	System's Response
1	User selects "Write a Review" option for a food truck	System displays review form with rating options and text field
2	User enters review text and selects a rating	System stores the input temporarily
3	User clicks "Submit Review" button	System attempts to save the review but encounters a database error
4	User waits for system response	System displays an error message informing the user that the review couldn't be saved due to a technical issue
5	User acknowledges the error message	System returns the user to the food truck's page without saving the review

System Architecture

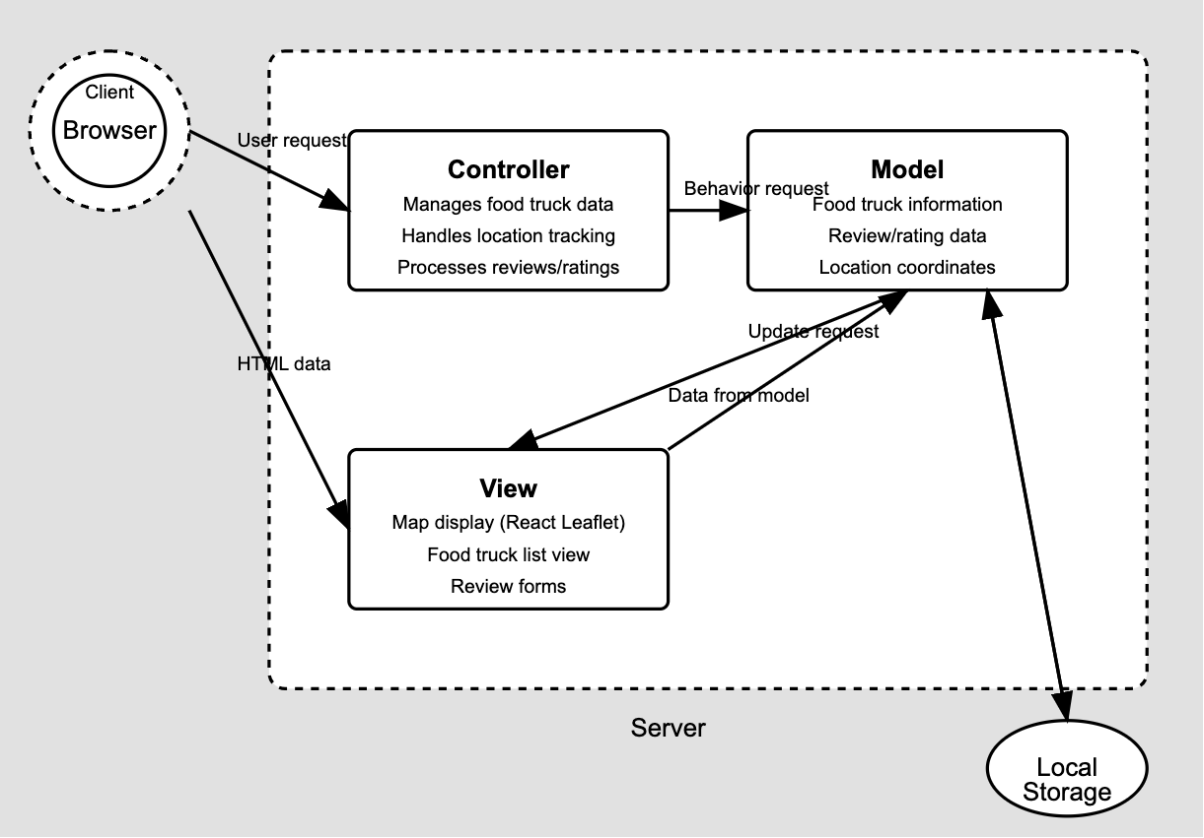
(Describe the high-level design of the software. List the components, name the architectural pattern you've used and provide a diagram.)

Components

1. **Frontend**: Next.js (React Framework)
2. **Backend**: JavaScript
3. **Database**: Local Storage

Architectural Pattern

We implemented the Model-View-Controller (MVC)


Client
Browser
User request
Controller
Manages food truck data
Handles location tracking
Processes reviews/ratings
Behavior request
Model
Food truck information
Review/rating data
Location coordinates
Update request
HTML data
Data from model
View
Map display (React Leaflet)
Food truck list view
Review forms
Server
Local Storage