Proposal Outline

1. Introduction

1.1 Background

1.2 Group members

1. Application purpose and function

2.1 Application Purpose

2.2 Application range and potential stakeholders

2.3 Application main function

2.3.1 Application functions

2.3.2 UML Diagram

2.3.3 Data structure

2.3.4 Data Parse Example

2.3.5 Data Algorithm

1. Project analyze

3.1 GUI feature

3.1.1 Designed GUI

3.1.2 Example Output UI

3.2 Website Performance

3.3 Future Development and Shortcoming

1. Reference
2. **Introduction**

**1.1 Background and Project Name**

Urbanization development at a recent time highly increases choices the people can make when they would like to find a place to have dinner, play games or buy things. With more and more stores opened, people can hardly make a choice among them, especially for those who have difficulties to make a choice. Also, facing several new stores, people can not make a decision based on their quality. To solve this problem, we are going to develop an application called Public Review.

Github link: <https://github.com/Steven-ZhangJM/Hackathon>

**1.2 Group members**

The application are developed by four people from University of Wisconsin-Madison: Chengpo Yan: [cyan46@wisc.edu](mailto:cyan46@wisc.edu); Bo Li: [bli379@wisc.edu](mailto:bli379@wisc.edu); Zhewei Song: [zsong96@wisc.edu](mailto:zsong96@wisc.edu); Jinming Zhang: jzhang2279@wisc.edu.

1. **Application purpose and function**

**2.1 Application purpose**

To help users know what they want about the surrounding buildings, including restaurants, book stores, hospitals, or the skyscraper. The user can search for one category of the building and the application will recommend a list of places which conform the user needs. Also, for each place the user had met, they can write down evaluation and let others know what this place is like.

**2.2 Application range and potential stakeholders**

For best estimation, the application can cover the whole U.S. Due to the time limit, we will only show one sample as taking the gps location and give recommendation. The user of the application may be people under 40 years old who have interest in discovering new places.

**2.3 Application main function**

In this section, we are going to present the main functionalities of our application, and the algorithm supporting this application to fulfill the users’ needs.

**2.3.1 Application function**

**2.3.1.1 Location function**

The application utilizes GPS data to locate current users’ positions. It also stores the location of various buildings in GPS coordinates. Users can put the searching results in the order of distances. They can filter the range of distance to narrow down the result.

**2.3.1.2 Label function**

For each location, the system allocate several labels based on its

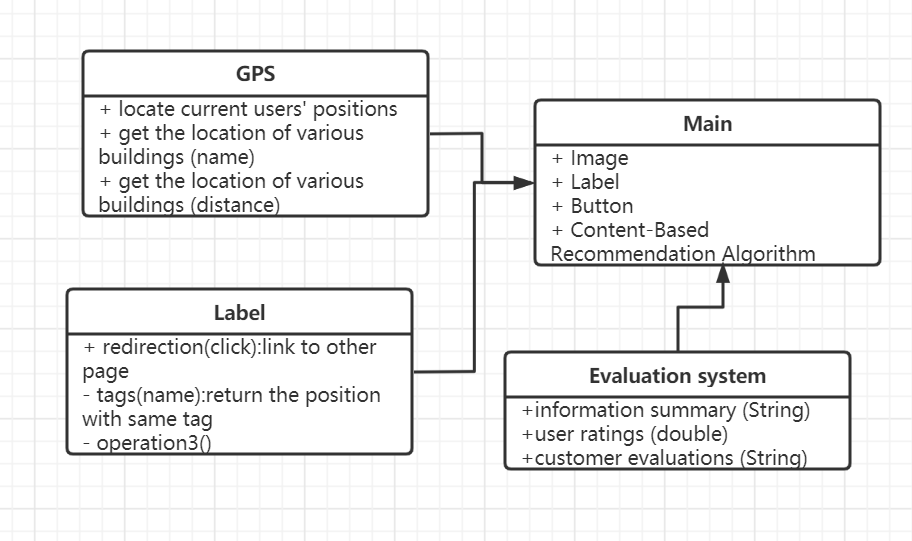
functionality, such as restaurants, bookstores, etc. Similar labels or a group of labels where users always visit together are viewed as extensions and they are provided to users under the suggestion tab. Users can search the place they want to go to through typing labels in the search bar. For precise search, direct store name can lead to a more desirable output.

**2.3.1.3 Evaluation System**

Each of the buildings or stores has its own homepage in the

application. It contains store information summary, user ratings, and customer evaluations. Customers can only write evaluation while they are physically inside the store in order to prevent malicious actions.

**2.3.2 UML Diagram**

****

**2.3.3 Data structure**

**2.3.3.1 Input data**

Input data refers to the data related with different stores or

buildings. It contains names, addresses, imagine of the building, and users evaluation data. The program uses SQL to fetch data from the database into the application.

**2.3.3.2 User input data**

User input data contains data from search bar and customer

evaluations. Customer evaluations are stored in a hash table based on the rating and elements in the same hashtable are categorized based on the length of the comments.

**2.3.3.3 User GPS data**

User location data will be parsed into a 4-Byte data and then separate to be distinguished.

**2.3.4 Data Parse example**

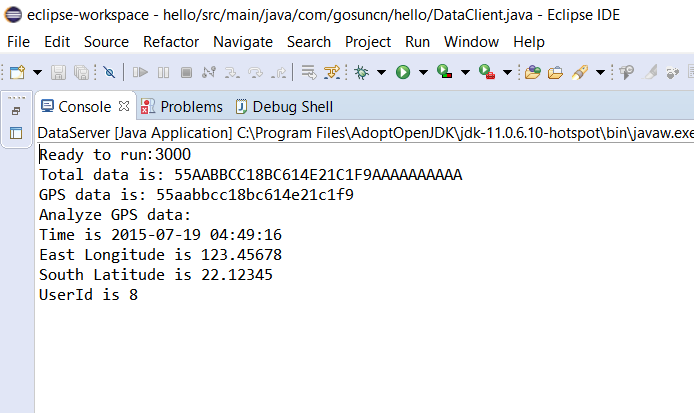
Using netty to build both client and server. The client sends 16Bytes data to the server and then the server parses the data into different categories, which may be used for future recommendation. The data 16Byte data includes 11 bytes for gps location and 5 bytes for personal choice, in the test, we will only use 11 bytes to find the gps location of the user. Detailed code is in Github.

**2.3.4.1.Definition of GPS data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Byte Endianness | 00-03 | 04 | | | 05-07 | 08-10 |
| Bit segment | 31-0 | 7 | 6 | 5-0 | 23-0 | 23-0 |
| Meaning | Time | E/W Longitude  0x0 - East  0x1 - West | S/N  Latitude  0x0 South  0x1 North | User ID | Longitude  Value | Latitude  Value |

**2.3.4.2 Data Example of GPS location**

We will use Data “55AABBCC18BC614E21C1F9AAAAAAAAAA” to do the test, the gps data is “55AABBCC18BC614E21C1F9”. Inside, the latitude degree is from 0 to 90 with accuracy: 0.00001, and longitude degree from 0 to 180 with accuracy: 0.00001.



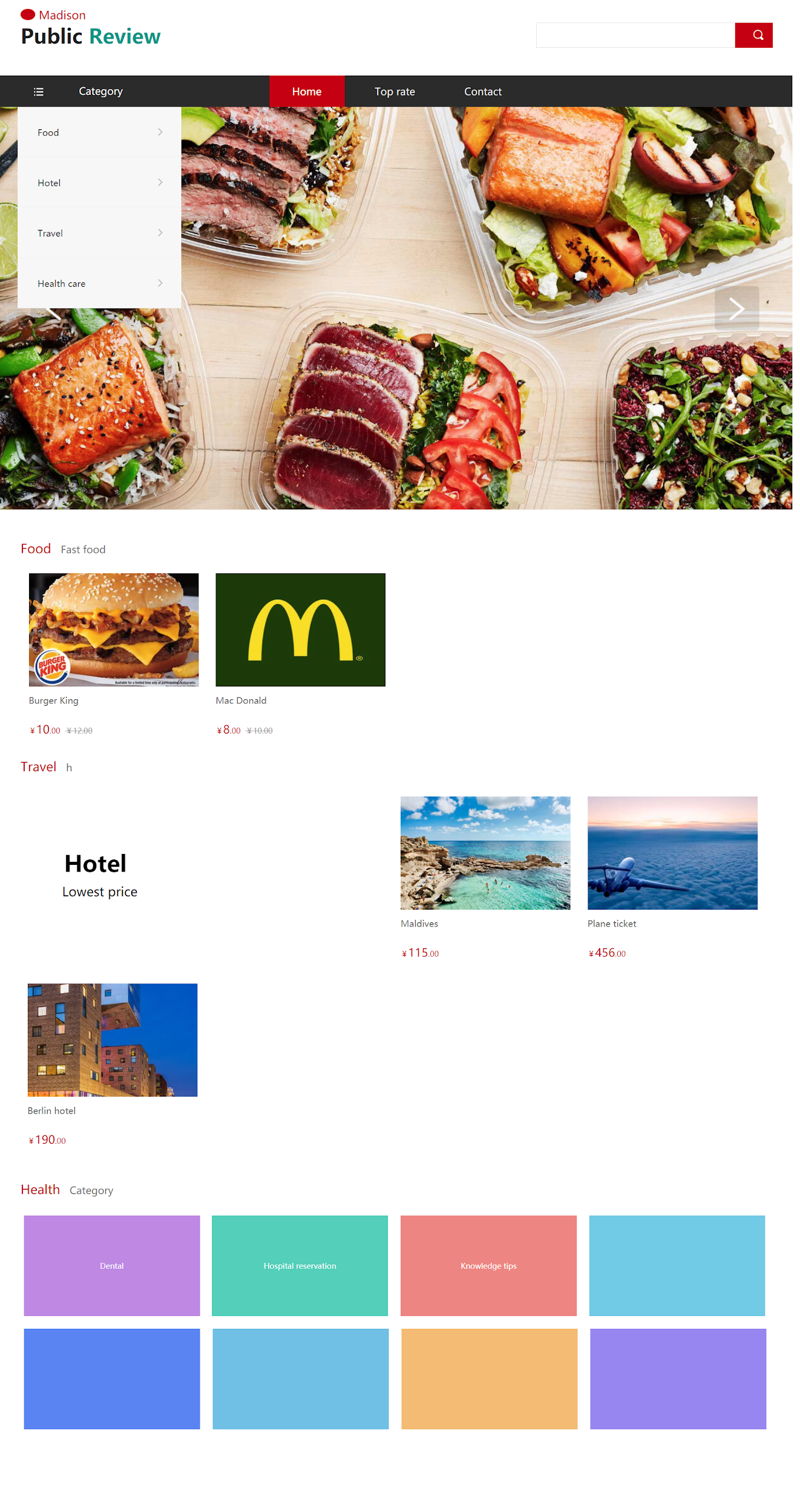
**2.3.5 Data Algorithm**

Using Content-Based Recommendation Algorithm, we simply use KNN algorithm to recommend places for users. We assume the users have already searched for the same kind of place several times so our application may work better for them.

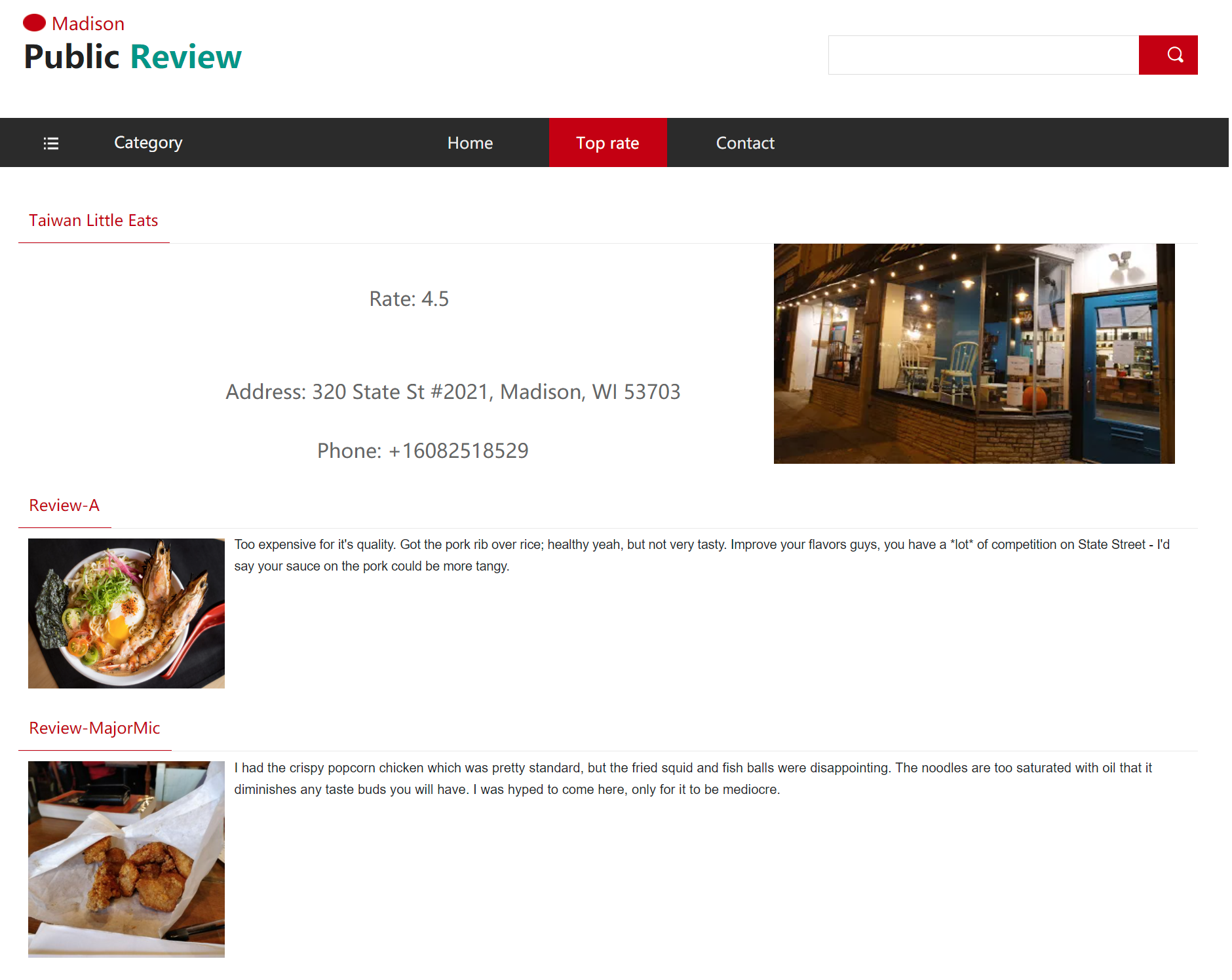
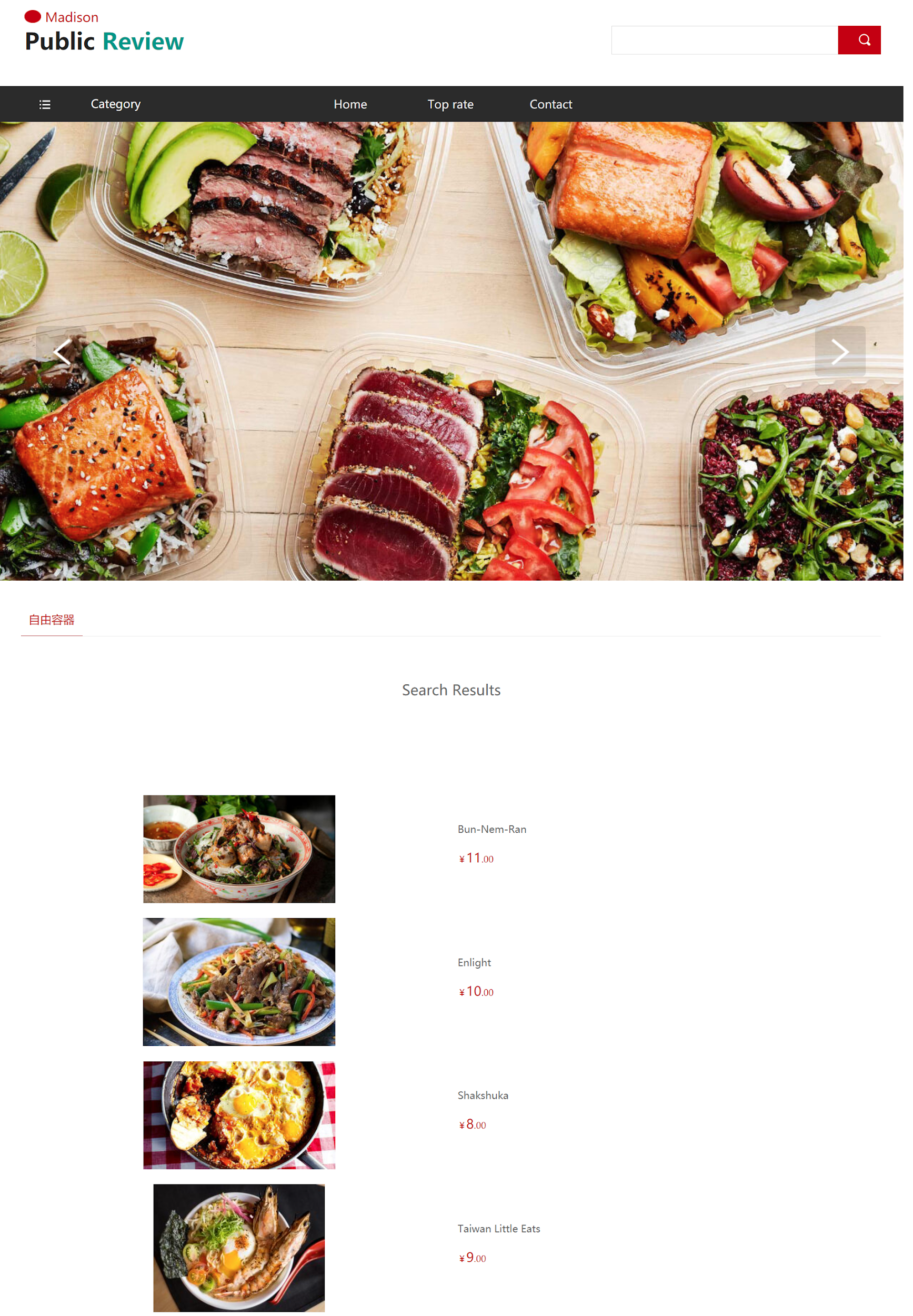
1. **Project Analyze**

**3.1 GUI feature**

**3.1.1 Designed GUI**

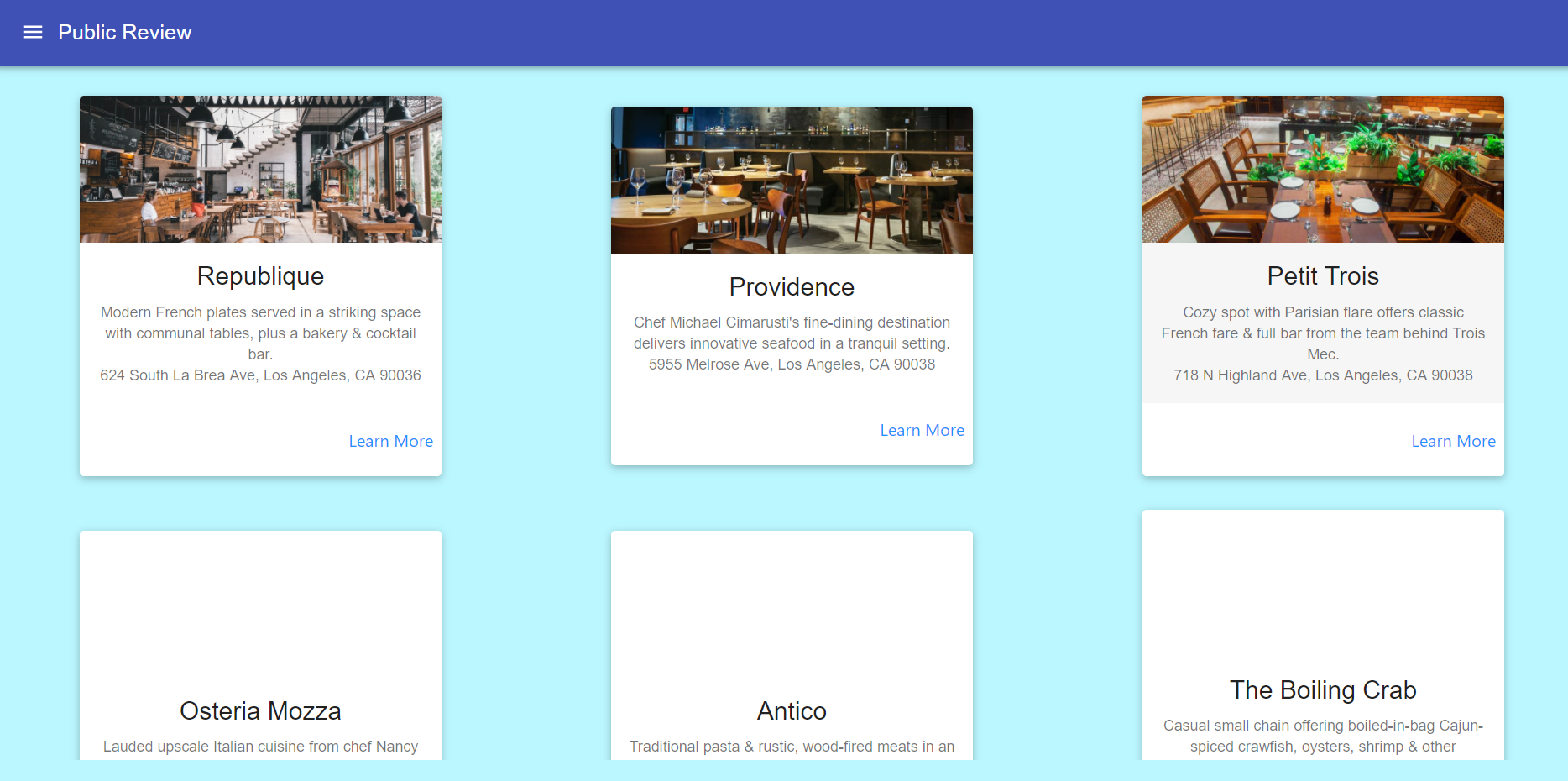
****

**3.1.2 Example Output UI**

****

**3.2 Website Performance**

This website shows a thumbnail of the application, using react to create and the code is in the Github.

****

**3.3 Future Development and Shortcoming**

Our project is just a beginning of our application, we have established the main structure of our application and we have begun to build some modules for it. However, we still lack time to develop it more completely. And it still needs plenty of knowledge to realize this project, but we might continuously work on that idea.

1. **Reference** 1.<https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm_device=c&utm_keyword=&utm_matchtype=b&utm_network=g&utm_adpostion=&utm_creative=278443377095&utm_targetid=dsa-429603003980&utm_loc_interest_ms=&utm_loc_physical_ms=9061377&gclid=EAIaIQobChMIztWJrfz26wIVj8_tCh0WvwcCEAAYASAAEgJKK_D_BwE>

2. <https://netty.io/wiki/user-guide-for-4.x.html>