Product Design and Requirement Document

ECE651- UW Foodie

- Group Member
- Objective
  - Background
  - Requirement
- Release
- System design
  - Module
  - Database
    - ER Diagram
    - Fetch Data from API to Database
    - Technique Requirement
    - Front End
    - Back End
- User flow and Design
  - List page
    - Prototype
    - Filter
  - Restaurant Detail Page

Group Member

Shuhang Yan- 20868596
Xin Liu- 20597920
Yixiang Liu- 20861946
Ye Fan- 20868356

Dev Team Front End – Shuhang Yan
Dev Team Back End- Xin Liu
Product Owner- Yixiang Liu
ScrumMaster- Ye Fan

Objective

Background

Studying and working at the University of Waterloo consumes a great deal of energy. It's necessary and sometimes urgent for students and faculty to find the nearest tasty food that's available in a convenient way. A web app that collects the local food store info and provides the search function can make eating at a campus more enjoyable.

A potential customer said, No food, little hope. During the last term, after studying in my office for a whole day, I could hardly find a place on campus to feed myself. Those stores were closed or did not provide the food that I wanted. Since most of the food stores on campus are currently not accessible on google map, a web app that collects and shares the info of the local food store would be helpful.

Requirement

1. Provide information about the current food stores on campus for the students and faculty.
2. Filter the stores that users need, such as opening status, store types, etc.

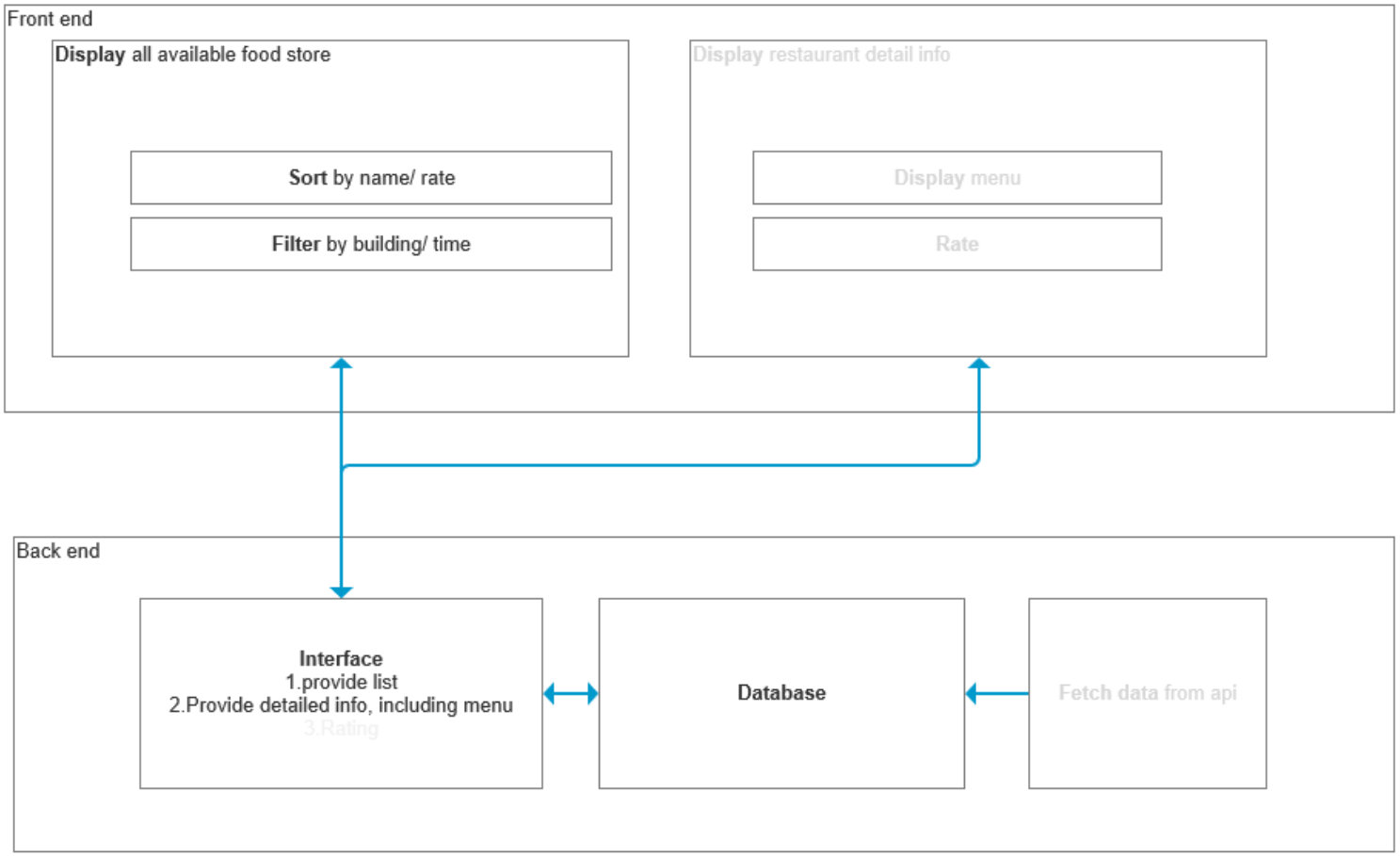
Release

Release	*1.00*
Date	*Feb-13-2020*
Content	Restaurant List Display

Release	*2.00*
Date	*Apr-1-2020*
Content	1.Restaurant detail info 2.Design optimization 3.Data fetching

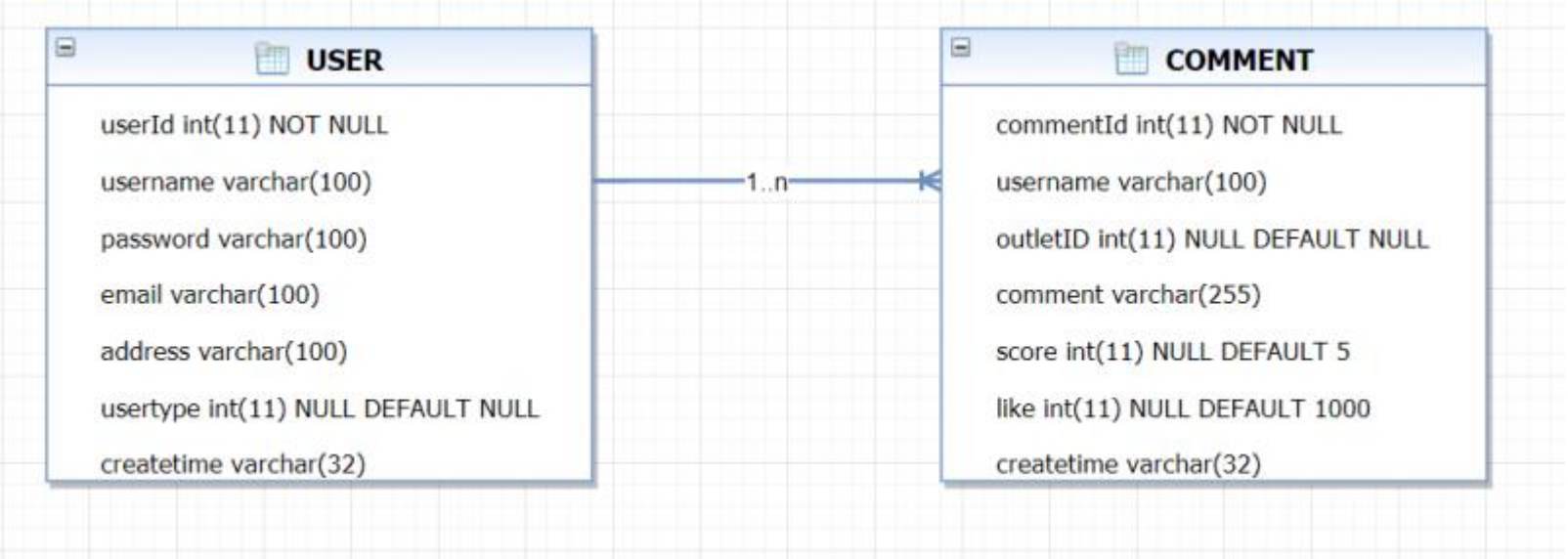
System design

Module



Database

ER Diagram



Fetch Data from API to Database

Now front end should use Node.js to fetch most of the data from API.
For comment and user info, they should store in the database and communicate with an interface.

In that way, we can ensure that all the shop info is up-to-date, while we still have the comment and user info stored safely.

Technique Requirement

Front End

React
JS

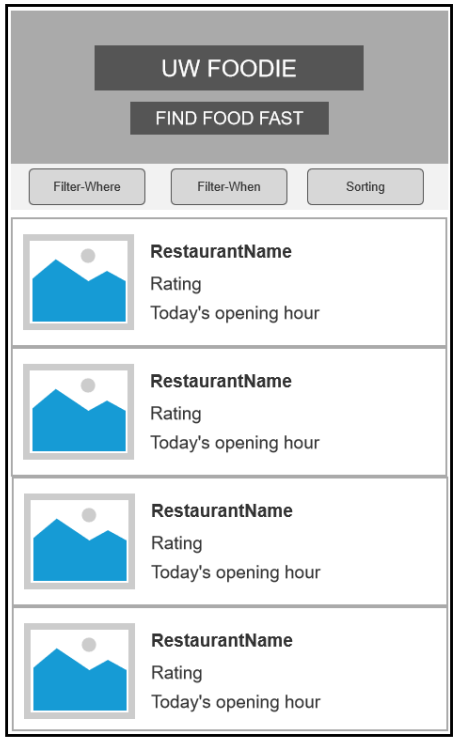
Back End

Java (JDK 1.8+)
MySQL 5.5
maven 3.6
Redis

User flow and Design

List page

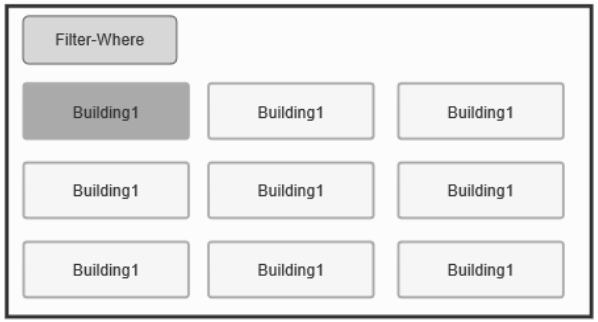
Prototype

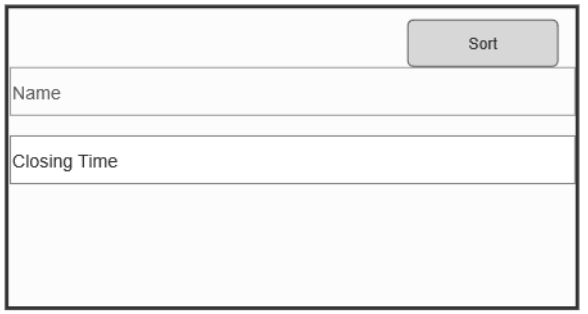


Default sort: by name, alphabetical. When the character number is beyond (20?), using … after the 30 characters.

Name: show restaurant detail
Rating: Show the rating by stars plus 4.5/5.0.

Filter



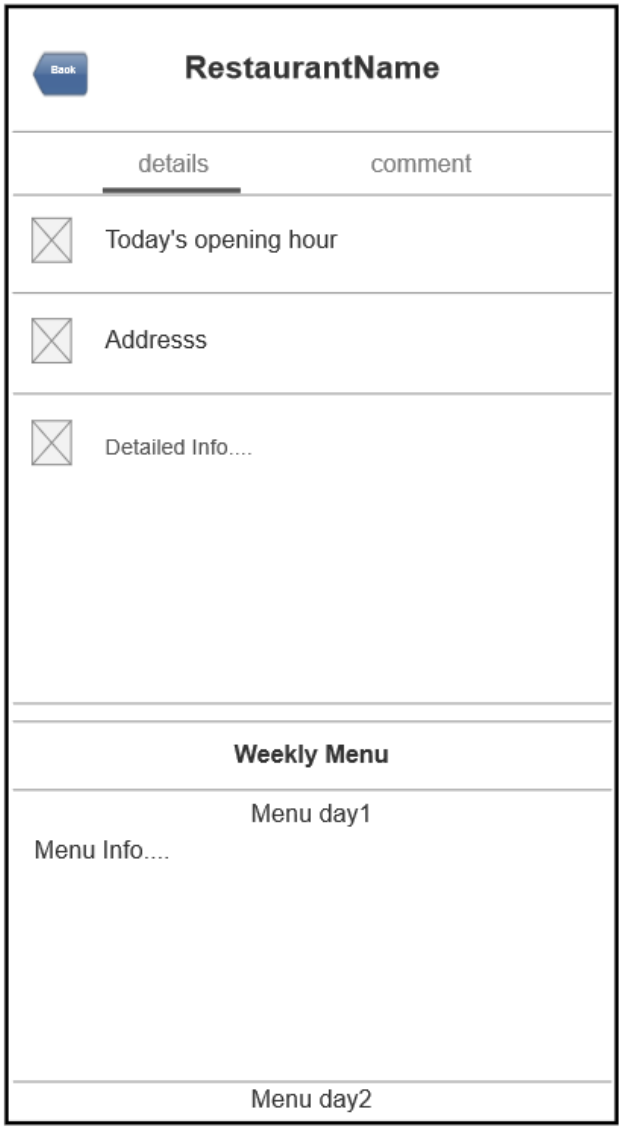


The filters can be combined
Where to eat:
Can only choose one option. After applying the filter, the website can show all the opening stores at that location, and the filter itself changed to the location.
When to eat:
Can only choose one option. After applying the filter, the website can show all the opening stores at that time, and the filter itself changed to the time.
Sort by:

1. Name alphabetical, from A-Z
2. Closing time, from early to late

Can only choose one option. After applying the sorting option, the filter can show which sort the user has chosen.

Restaurant Detail Page



Show Detailed Info as listed below

Detail Tab	Comment Tab
Shop name	Shop name
Opening hour	1.User name 2.Shop score by user 3.Shop comment by user As a form with submit button
Shop address	
Detailed shop info	
Weekly menu show by day	List view of shops comments include 1.User name 2.Shop score by user 3.Shop comment by user