
Team M

**OneBite
Software Requirements Specification
For Online Restaurant System
Final Report**

Developers: Chenhao L., Peggie L., Mavis Y., Riaz A., Jessie S.

Introduction

Python 3.x.x

The project was built using Python, one should have any version of python 3 installed on their machine.

Flask

Flask was our main module for this project. It served as the front end render engine for loading HTML and CSS pages and backend communication with the database.

Jinja

This was provided in the installation of Flask, and it is used to dynamically show data in the HTML.

MySQL

MySQL database was used to store all the content from the website. This includes the customer, manager, chef, dishes, blacklists, and so much more.

Flask-mysql

Flask MySQL module helps make working with MySQL in Flask a lot easier. This module was used heavily to read and write data from the database.

Before starting the project, please add "config.json" file to the directory in order for the database to work

```
{
  "SECRET_KEY" : "",
  "MYSQL_HOST" : "localhost",
  "MYSQL_USER" : "root",
  "MYSQL_PASSWORD" : "",
  "MYSQL_DB" : "team_m_restaurant"
}
```

Make sure to fill in the secret key and the password before running the program.

Bootstrap

Although most of the website design was written in our own custom styles. Bootstrap has helped to speed things out at the end when we are running out of time to make the UI look good.

System Specification Requirement Checklist

1. Provide a GUI, not necessarily web-based, with pictures to show the descriptions of each dish and price; each registered customer/VIP has a password to login, when they log in, based on the history of their prior choices, different registered customer/VIP will have different top 3 listing dishes. For new customers or visitors, the top 3 most popular dishes and top 3 highest rated dishes are listed on the first page **[FINISHED]**

2. A customer can choose to 1) pick up the dishes in person, or 2) by restaurant delivery. For case 1) s/he can only complain/compliment the chef **[FINISHED]**
3. A customer can file complaints/compliments to chef of the food s/he purchased and deliver person who delivered the dish or other customers who didn't behave in the discussion forums. Delivery person can complain/compliment customers s/he delivered dishes, all complaints/compliments are handled by the manager. The complained person **[FINISHED]**
4. Registered customers having 3 warnings are de-registered. VIPs having 2 warnings are put back to registered customers (with warnings cleared). The warnings should be displayed in the page when the customer logs in **[FINISHED]**
5. Every customer should deposit some money to the system. If the price of the order is more expensive than the deposited money in the account, the order is rejected and the customer receives one warning automatically for being reckless **[FINISHED]**
6. Customers who are kicked out of the system or choose to quit the system will be handled by the manager: clear the deposit and close the account. And kicked-out customer is on the blacklist of the restaurant: cannot register any more **[FINISHED]**
7. The chef whose dishes received consistently low ratings (<2) or 3 complaints, will be demoted (less salary), a chef demoted twice is fired. Conversely, a chef whose dishes receive high ratings (>4) or 3 compliments, will receive a bonus. One compliment can be used to cancel one complaint. The delivery people are handled the same way **[PARTIALLY DONE]**

8. The delivery people will compete to deliver the order by bidding, the manager assigns the order from bidding results: the one with lowest delivery price is generally chosen; if the one with higher asking price is chosen, the manager should write a memo in the system as justifications. The delivery person who didn't deliver any in the past 5 orders will automatically receive one warning **[FINISHED]**
9. Each team comes up with a creativity feature of the system to make it more exciting, e.g., smart-phone based system, voice-based features, or efficient route planning for delivery, which is worth 10% of overall score of the final project **[FINISHED]**

Contributions

Name	Contributions
Chenhao Li	<ul style="list-style-type: none">- Created about, menu, footer, and other miscellaneous pages- Creating the required class objects such as the User, Chef, Manager, Delivery Personnel, and Dish- Created sessions so the user can stayed logged in- Added safety checks so guests cannot access the dashboard, cart, profile, etc- Added ratings to dishes- Customer can order dishes through the menu page- Added notification for warnings
Mavis Ye	<ul style="list-style-type: none">- Created the home, cart, checkout, order confirmation, profile, and orders page

	<ul style="list-style-type: none"> - Created and managed all aspects of the database - Created the functionality for the profile page. (ex: change password, set address, update wallet, etc) - UI design for most of the pages - Implemented login/signup - VIP functionalities - Orders system - Blacklist functionalities - Delivery bidding system
Peggie Liang	<ul style="list-style-type: none"> - Creation of the regular and the hidden Navbar - Building the customer, manager, and chef's dashboard - Added option for manager to dispute a claim - Close accounts, give promotions/demotions, and blacklist customers - Finalize the forum page and connected it to the database
Jessie Su	<ul style="list-style-type: none"> - Created the login, forgot password, and new user page. - Building the delivery/chef personnel's dashboard - Added comments and compliment/complaints system for the customer and delivery personnel - Connected delivery and chef dashboard to database

	<ul style="list-style-type: none"> - Connected dispute/warnings to database - Chef can modified the menu
Riaz Ahmed	<ul style="list-style-type: none"> - Built the otemplate for the menu page and forum page - Updated about page with relevant information

Git Repo

<https://github.com/Charptr0/CSC322-OneBite>

Remarks

The whole group except for Riaz worked really hard on this project. During the course of the project, he put minimal effort into the project. He did not meet weekly sprints and after we brought up the fact that he was not pulling his weight, and he agreed, he still did not show any progress. As a group, we have come to an agreement to have Riaz penalized.

As a team, we decided that Riaz deserves 40% of the grade. Here is a list of the reasons why:

1. Until we mentioned that he was not putting enough effort into his work, he did not show any initiative towards our project. After confronting him, he said that he would pick it up and told us to assign things to him because he said he didn't know what to work on. After telling us that he was going to pick it up, he only worked on the project for 1 week and did not help after that.
2. We gave him things to work on for weekly sprints but when the week came he did not finish what he had to do. He said he would pick up the slack but he didn't.

3. He only worked on the template for order page and forum page but because of the lack of efficiency of the page, group members had to redo everything.
4. Him being on the team made the progress inefficient due to a lot of backtracking. This resulted in the group being off schedule and other group members had to pick up extra work that was assigned to him.