

复旦大学

数据库引论

Project 1: Glutton



作者：

胡天晓

张昊晗

刘婧源

2017 年 5 月 6 日

1 项目简介

- 项目背景

Glutton 数据库项目 是为复旦大学（张江校区）的吃货们设计的外卖网页应用平台，以 python 链接 SQL 语言， web 为前端，界面友好。使用者分为商家和用户，支持商家查询、创建、更改、删除菜品，用户查询商家、菜品，创建、更改、删除、评价订单等操作。

- 依赖库 | 组织架构 flask/bootstrap/sqlite
- 项目文件树
- 运行方法
- flask 简介
- bootstrap 简介
- sqlite 简介
- 用户密码 md5 加密
- 其他

2 项目细节

2.1 针对用户的功能

我们的 web app 面向张江地区的人民，可以随意选择喜欢的商家和菜品。主界面的最上方是导航栏——左侧有四个按钮：可以选择以用户或商家身份使用这个应用；在登录后，还可以查看自己的主页和订单。右侧和中部同为登录和注册按钮。最下方包含了作者、版权信息等。

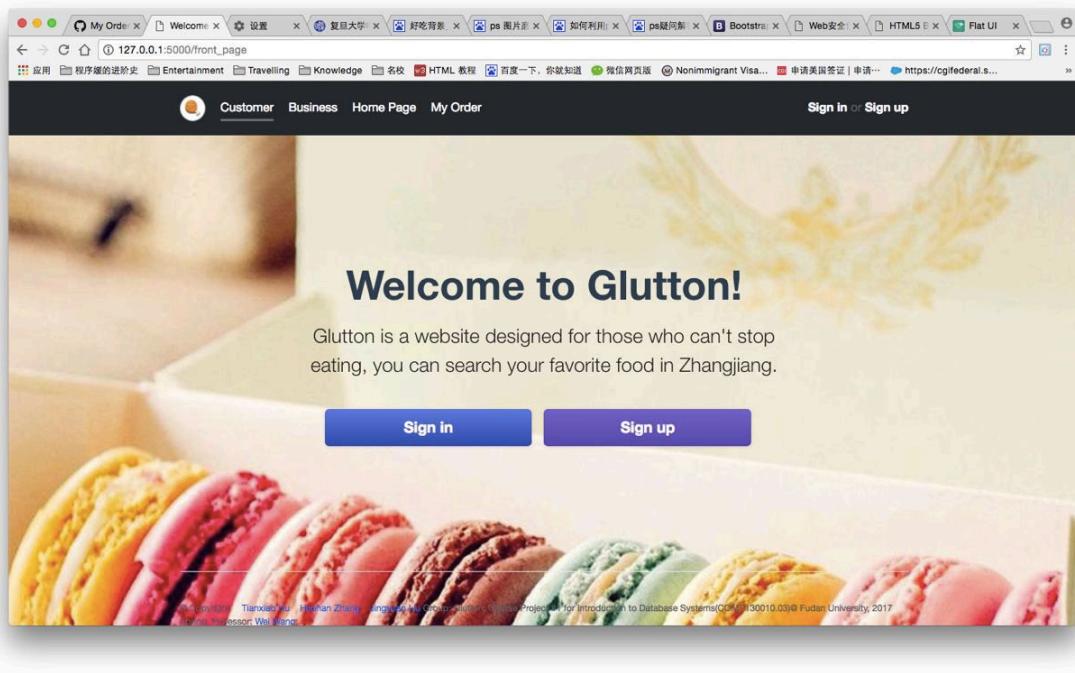


Figure 2-1: 主页面

- 注册/登录

点击 sign up 按钮我们进入注册页面，输入用户名，手机号，密码来注册。手机号必须是唯一的，如果有重复，系统会提示这个手机号已经被注册，请直接登录。

点击 sign in 按钮我们进入登录页面，用手机号和密码登录。如果密码输入错误，系统会提示密码或用户名错误。

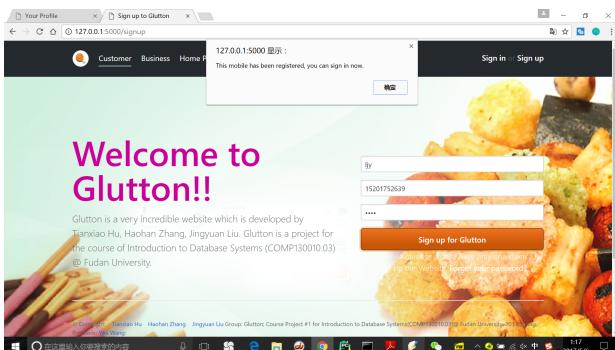


Figure 2-2: 用户注册

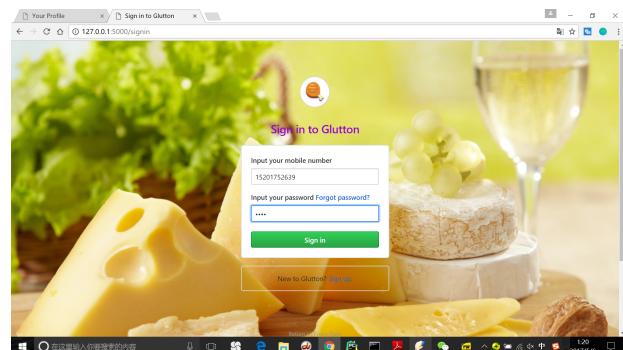


Figure 2-3: 用户登录

- 更改信息/头像

登录成功后，我们进入到用户的主页。主页的最上方是导航栏，左侧三个按钮可以选择主页，个人信息，订单。右侧有搜索栏和退出登录按钮。左侧图片点击可以更改个人口味偏好，下方显示用户名和手机号，以及编辑个人信息按钮。中间同样有搜索栏，

下方按食物分类配备了多种类型，如果汁奶茶、炸鸡汉堡等的食物分区，代表性商家可以直接鼠标点击进入。

The screenshot shows a user profile page with a blue header bar containing 'Home', 'Profile', 'Orders', 'Search Glutton', and 'Sign Out'. On the left, there's a profile picture of a cartoon character named 'bing ji lin' and a mobile number 'Mobile:15201752639'. Below the profile picture is an 'Edit profile' button. The main content area features several sections: 'Find Your Favorite' with a search bar, '果汁奶茶 : 青春中的花样年华' (Juice Milk Tea: The Flower Years of Youth), '美味面馆 : 汤饼一杯银线乱, 萎蒿如箸玉簪横' (Delicious Noodle Shop: A cup of tangbing with silver threads, withered chives like玉簪), '炸鸡汉堡 : 脂肪是治愈一切的良方' (Fried Chicken Hamburger: Fat is a cure for all), and '中华小吃 : 足不出户, 吃遍天下' (Chinese Snacks: Eat at home,遍天下). Each section contains descriptive text and small images.

© Copyright: Tianxiao Hu Haohan Zhang Jingyuan Liu Group: Glutton; Course Project #1 for Introduction to Database Systems(COMP130010.03)@ Fudan University, 2017 Spring.
Professor: Wei Wang;

Figure 2-4: 用户主页

点击 Edit profile 按钮进入修改个人信息页面，左侧可以选择更改的类型：个人口味偏好，基本信息，更改密码，退出登录。

基本信息可以修改用户名、地址、称呼和个人描述；密码修改，如果修改成功，系统会提示“Succeed”

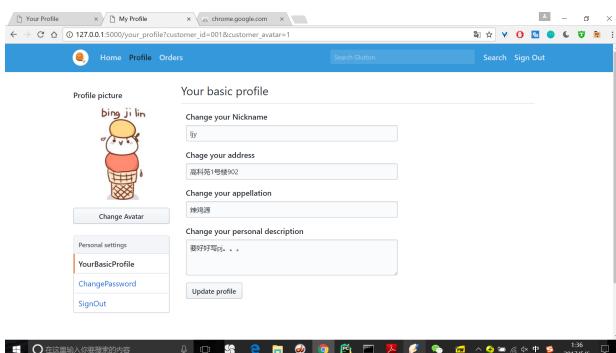


Figure 2-5: 用户更改信息

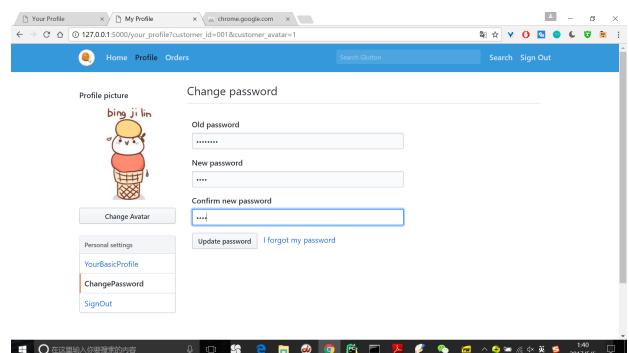


Figure 2-6: 用户更改密码

用户可以点击图片，修改个人口味偏好。共有 9 中偏好类型供选择，任意点击一个，系统会提示转换成功，页面跳转至个人主页，可以看到口味头像已有变化。

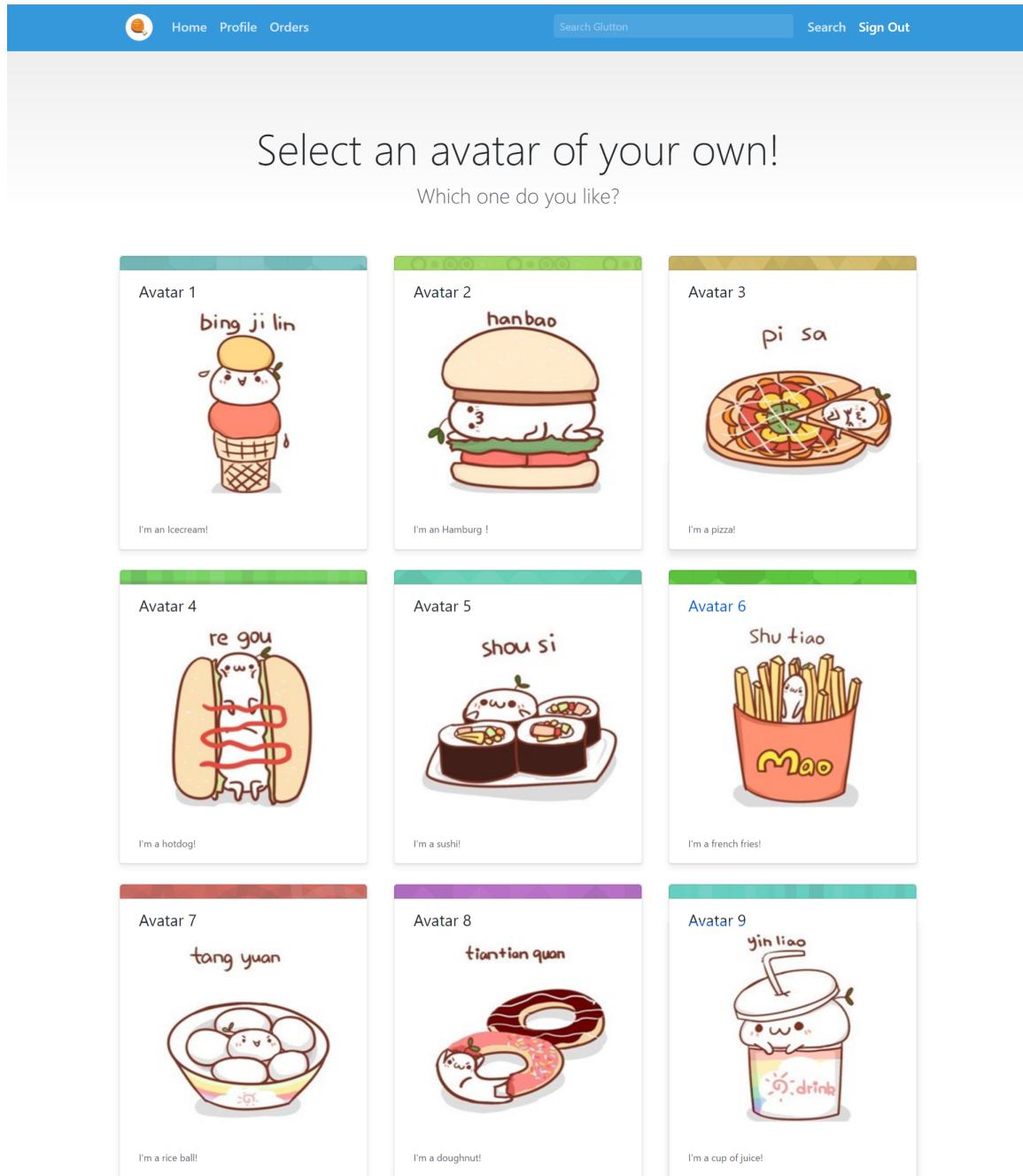


Figure 2-7: 用户口味选择

- 查找商家

在搜索框中输入商家的名称，例如“一点点”，可以查询到对应商家信息。点击可以进入商家。

还可以按照随机、月销量、起送费顺序查看商家列表，右侧有查询历史记录。

同样在商家内，可以按照随机、月销量、商品价格顺序查看菜品列表。

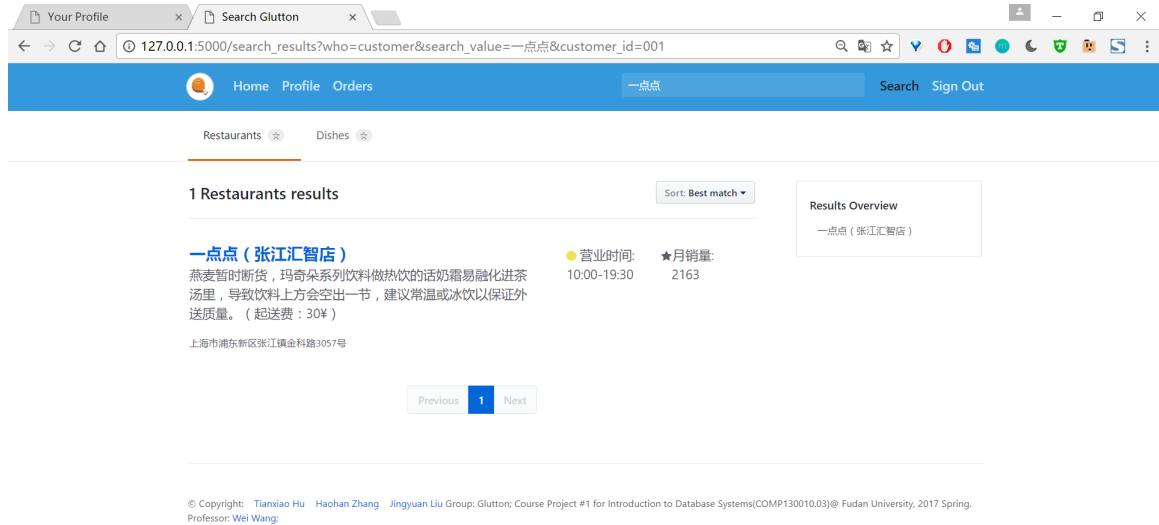


Figure 2-8: 用户查找特定商家

Results Overview

- 学城粥铺
- 肯德基 (张江高科店)
- 正新鸡排 (蔡伦路店)
- 格林果汁吧
- 享甜 (科苑路店)

Restaurant	Address	Operating Hours	Monthly Sales
学城粥铺	上海市浦东新区张江镇华佗路572号	09:00-23:00	743
肯德基 (张江高科店)	上海市浦东新区曹路路635号	10:00-21:55	1384
正新鸡排 (蔡伦路店)	上海市浦东新区张江镇蔡伦路574号	10:00-23:50	1487
格林果汁吧	浦东新区紫荷路210号1-3层	10:00-22:30	281
享甜 (科苑路店)	上海市浦东新区张江镇科苑路1107号 - 6室	09:45-00:55	1840

Figure 2-9: 用户查看商家列表

- 下单

跳转到商家页面，显示所有菜品，对每个菜品可以选择添加或删除，最后点击 Submit Order 提交订单。系统提示订单提交成功。

The screenshot shows a user's profile page on the Glutton app. At the top, there are navigation links for Home, Profile, Orders, a search bar labeled 'Search Glutton', and options to Sign Out. Below this, the user's name '一点点 (张江汇智店)' is displayed, along with their address and a brief description about their products.

The main content area lists various drink items with their details and purchase buttons:

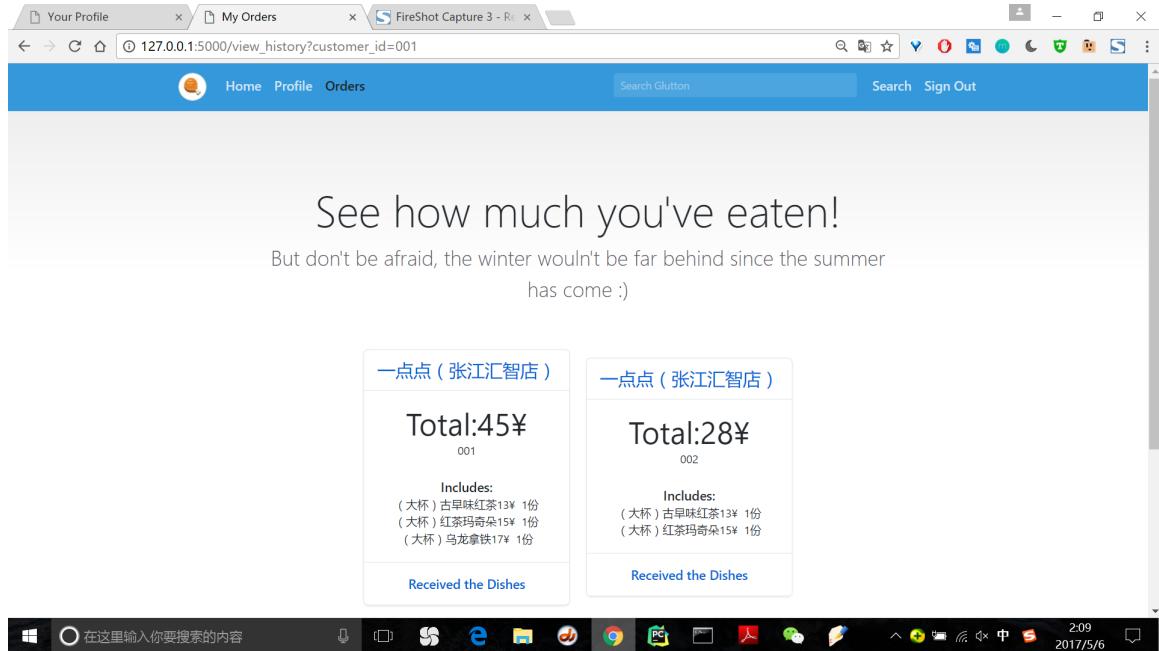
- (大杯) 红茶玛奇朵**: Price 15¥, Monthly Sales 717. Buttons: Cut (0), Add (green).
- (大杯) 波霸奶茶**: Price 13¥, Monthly Sales 1080. Buttons: Cut (0), Add (green).
- (大杯) 乌龙玛奇朵**: Price 15¥, Monthly Sales 388. Buttons: Cut (0), Add (green).
- (大杯) 红茶拿铁**: Price 17¥, Monthly Sales 156. Buttons: Cut (0), Add (green).
- (大杯) 乌龙拿铁**: Price 17¥, Monthly Sales 130. Buttons: Cut (0), Add (green).
- (大杯) 四季春玛奇朵**: Price 15¥, Monthly Sales 250. Buttons: Cut (0), Add (green).
- (大杯) 波霸奶绿**: Price 13¥, Monthly Sales 477. Buttons: Cut (0), Add (green).
- (大杯) 冰淇淋红茶**: Price 17¥, Monthly Sales 92. Buttons: Cut (0), Add (green).
- (大杯) 古早味红茶**: Price 13¥, Monthly Sales 13. Buttons: Cut (1), Add (green).
- (大杯) 养乐多绿茶**: Price 17¥, Monthly Sales 83. Buttons: Cut (0), Add (green).
- (大杯) 金桔柠檬**: Price 15¥, Monthly Sales 78. Buttons: Cut (0), Add (green).

At the bottom, there is a large green 'Submit Order' button.

Figure 2-10: 用户下单

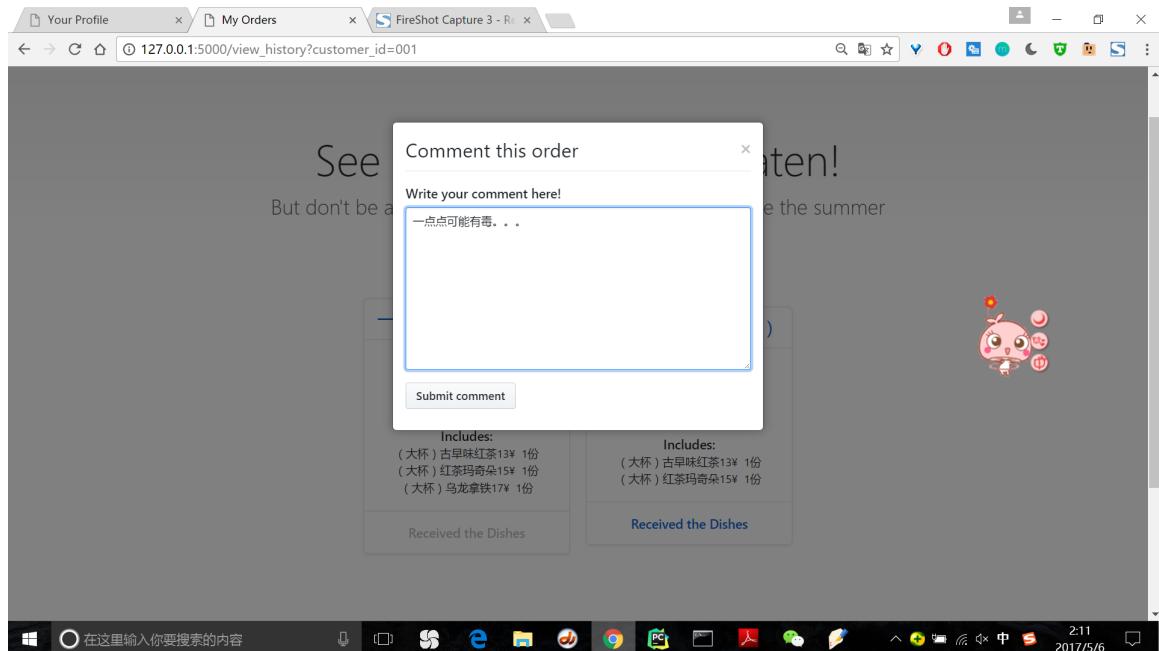
- 收货

点击 order 按钮，跳转到订单页面，可以查看所有订单信息。点击 Received the Dishes 确认收货。

**Figure 2-11:** 用户查看订单确认收货

- 评论

点击确认收货按钮后，会弹出评价窗口，用户可以写评价，系统提示提交评价成功，跳转到订单界面。

**Figure 2-12:** 用户填写评价

2.2 针对商家的功能

商家主界面与用户类似，右侧和中部为登录商家和注册商家按钮。

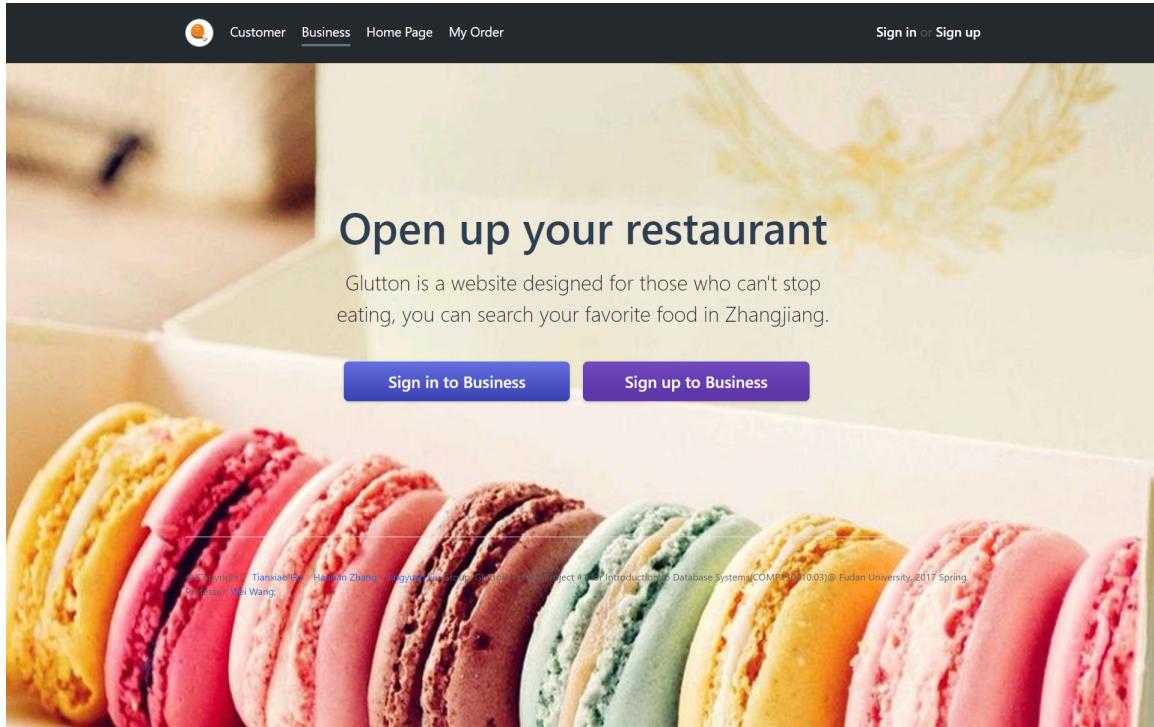


Figure 2-13: 主页面

- **注册/登录**

点击 sign up to business 按钮我们进入商家注册页面，输入用户名，店铺名，密码来注册。

点击 sign in to business 按钮我们进入商家登录页面，用用户名和密码登录。如果密码输入错误，系统会提示密码或用户名错误。

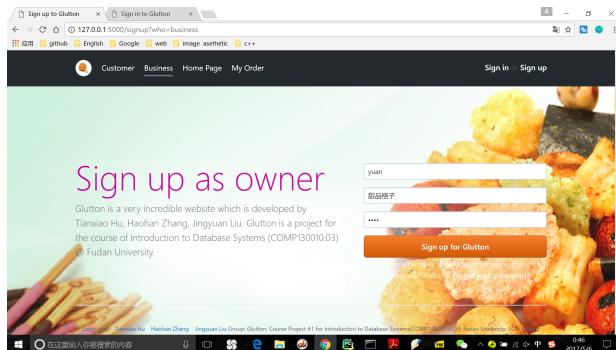


Figure 2-14: 商家注册

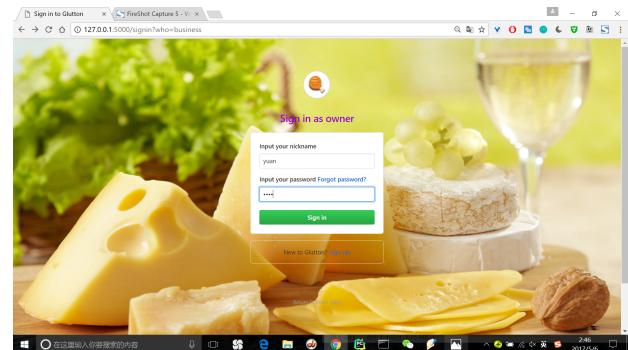


Figure 2-15: 商家登录

- **更改信息**

登录成功后，我们进入到商家的主页。主页的最上方是导航栏，左侧四个按钮可以选择主页，商家信息，菜品，订单。右侧有搜索栏和退出登录按钮。
左侧显示用户名和商家号，及修改商家信息按钮。

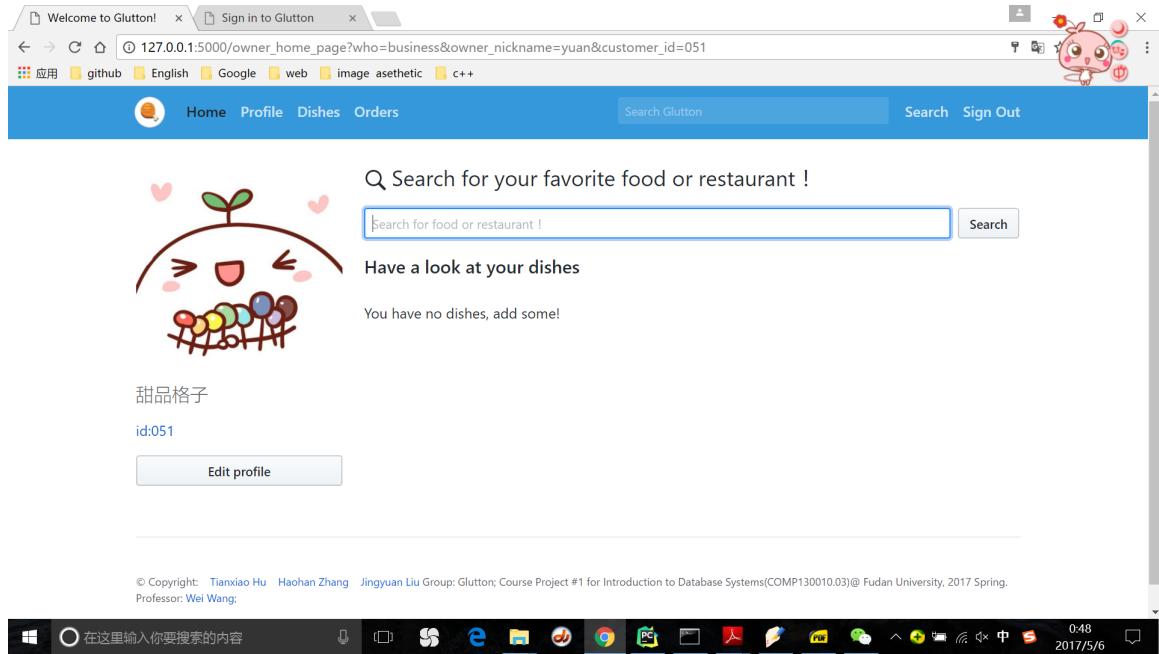


Figure 2-16: 商家主页

点击 Edit profile 按钮进入修改商家信息页面，左侧可以选择类型：基本信息，更改密码，查看订单，管理菜品，添加菜品，退出登录。

基本信息可以修改店铺名、地址、配送费、起送价、平均配送时间和店铺公告；密码修改，如果修改成功，系统会提示“Succeed”

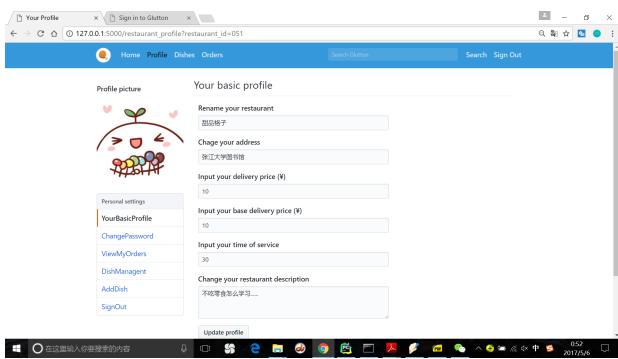


Figure 2-17: 商家更改信息

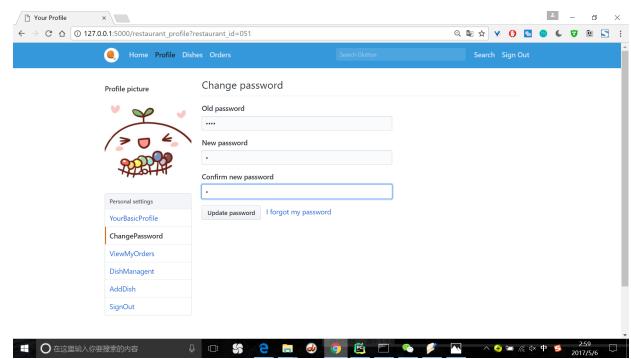


Figure 2-18: 商家更改密码

- 增加菜品

点击 AddDish 按钮可以添加菜品，需要填写菜品名和价格，菜品添加成功，系统提示载入成功。

我们也可以点击左侧 DishManagement, 或导航栏的 Dishes 进入菜品管理页面, 会显示已有的菜品, 并可以执行编辑、增加、删除操作。

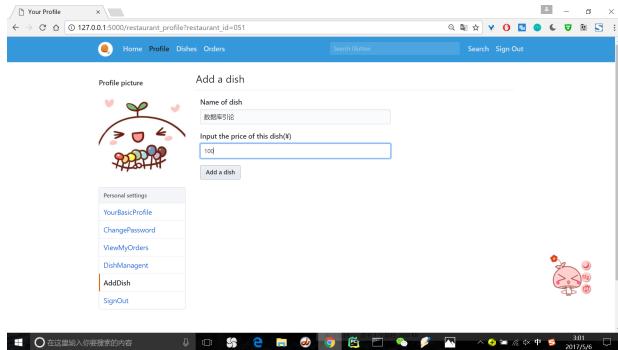


Figure 2-19: 商家增加菜品



Figure 2-20: 商家查看菜品

- 修改菜品

在菜品管理页面中, 点击 Edit 按钮, 可以对菜品名, 单价进行修改。

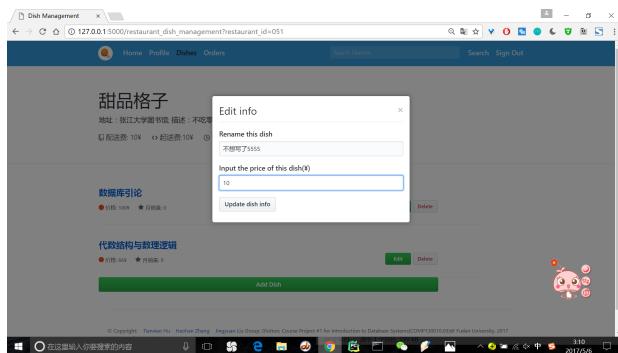


Figure 2-21: 商家修改菜品信息

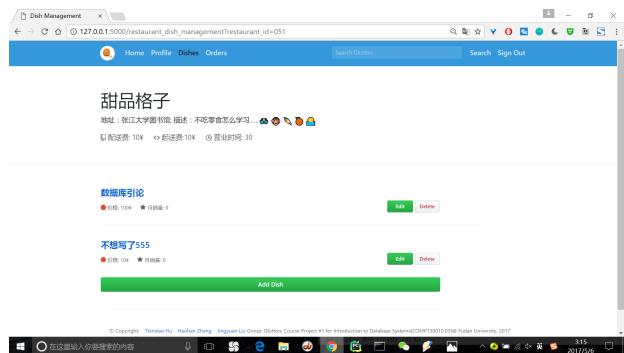


Figure 2-22: 商家修改菜品后

- 删除菜品

在菜品管理页面中, 点击 Delete 按钮, 可以对菜品删除。系统会提示是否确定要删除该菜品, 确定后, 提示菜品删除成功, 并跳转到菜品管理页面。

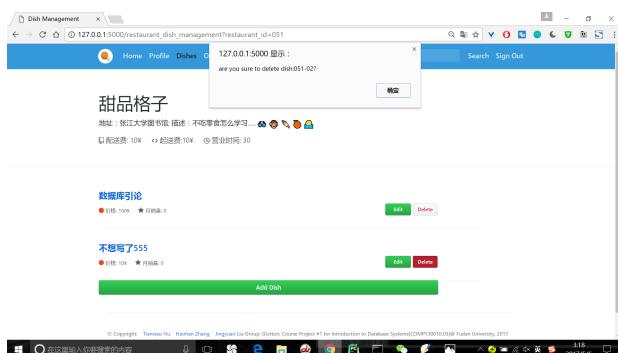


Figure 2-23: 商家删除菜品

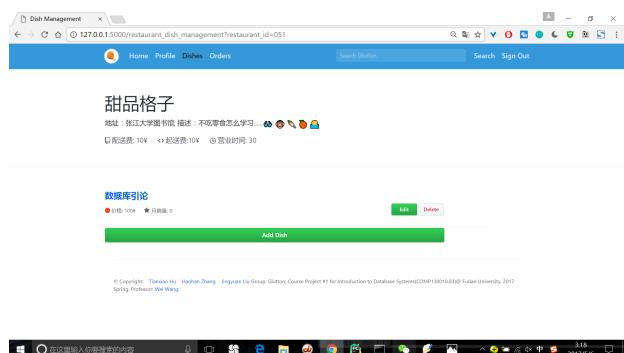


Figure 2-24: 商家删除菜品后

- 查看商家订单历史

新开的店铺订单信息为空，我们以用户身份登录，下单这家店铺的菜品，可以看到所有订单信息。

我们还可以查看每个订单的评论。

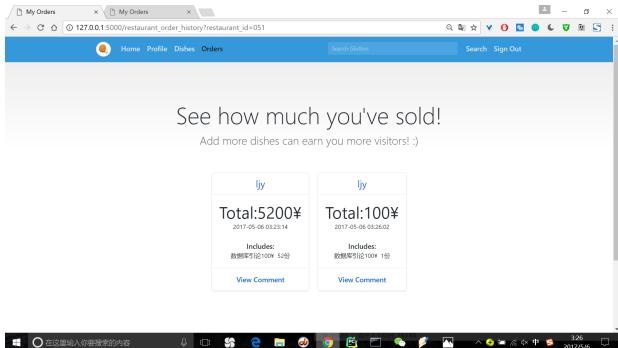


Figure 2-25: 商家查看订单历史

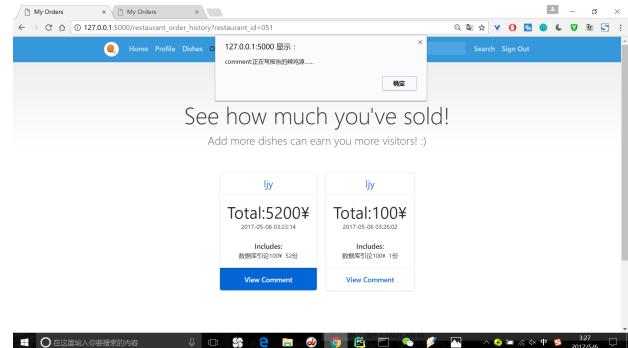


Figure 2-26: 商家查看订单评价

3 数据库简介

- 数据源

数据库 Glutton 所用数据来源于外卖平台“饿了么”网站，以复旦大学（张江校区）为中心，选取真实的商家信息作为数据源，保证了数据的真实性和有效性。

- 数据量

数据选取以蔡伦路、科苑路、华佗路的商铺为主，涵盖 50 个商家。每个商家菜品若干，最多为 17 道菜，最少为 4 道菜，共 393 道菜，平均每个商家 8 (7.86) 道菜。整个数据库包含 443 条商家及菜品记录。

- ER 图

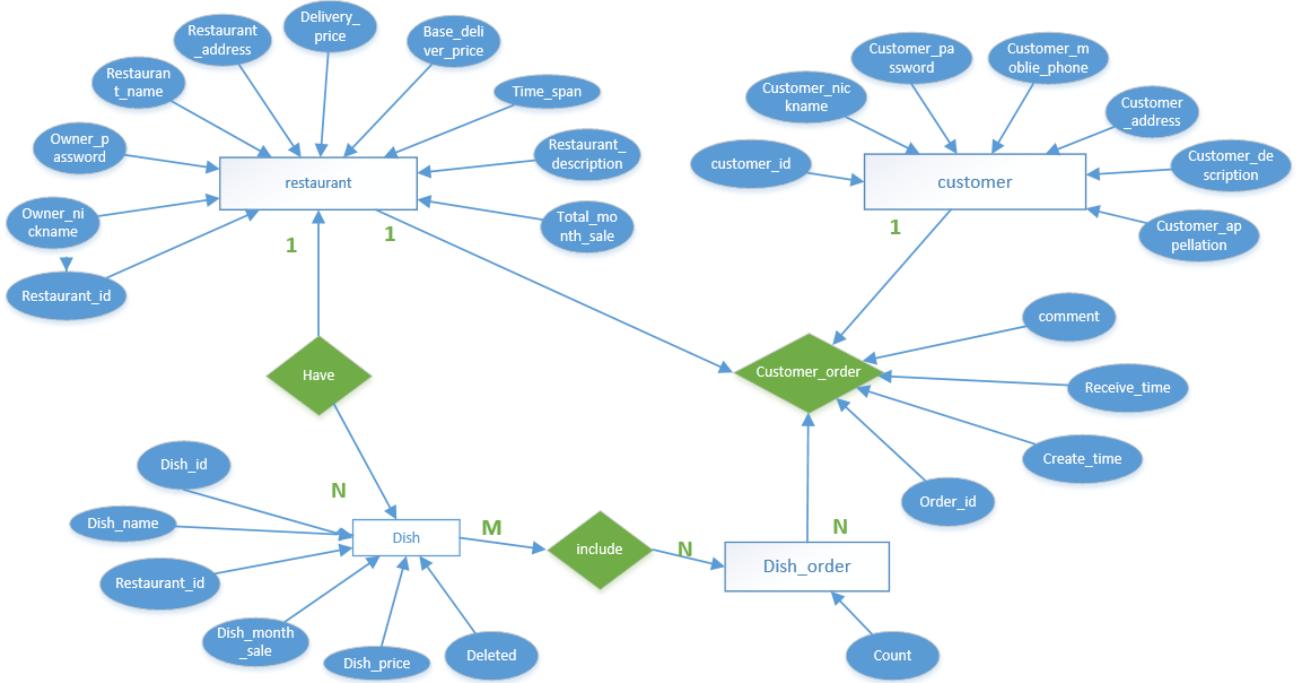


Figure 3-27: 数据库 ER 图

- 关系图

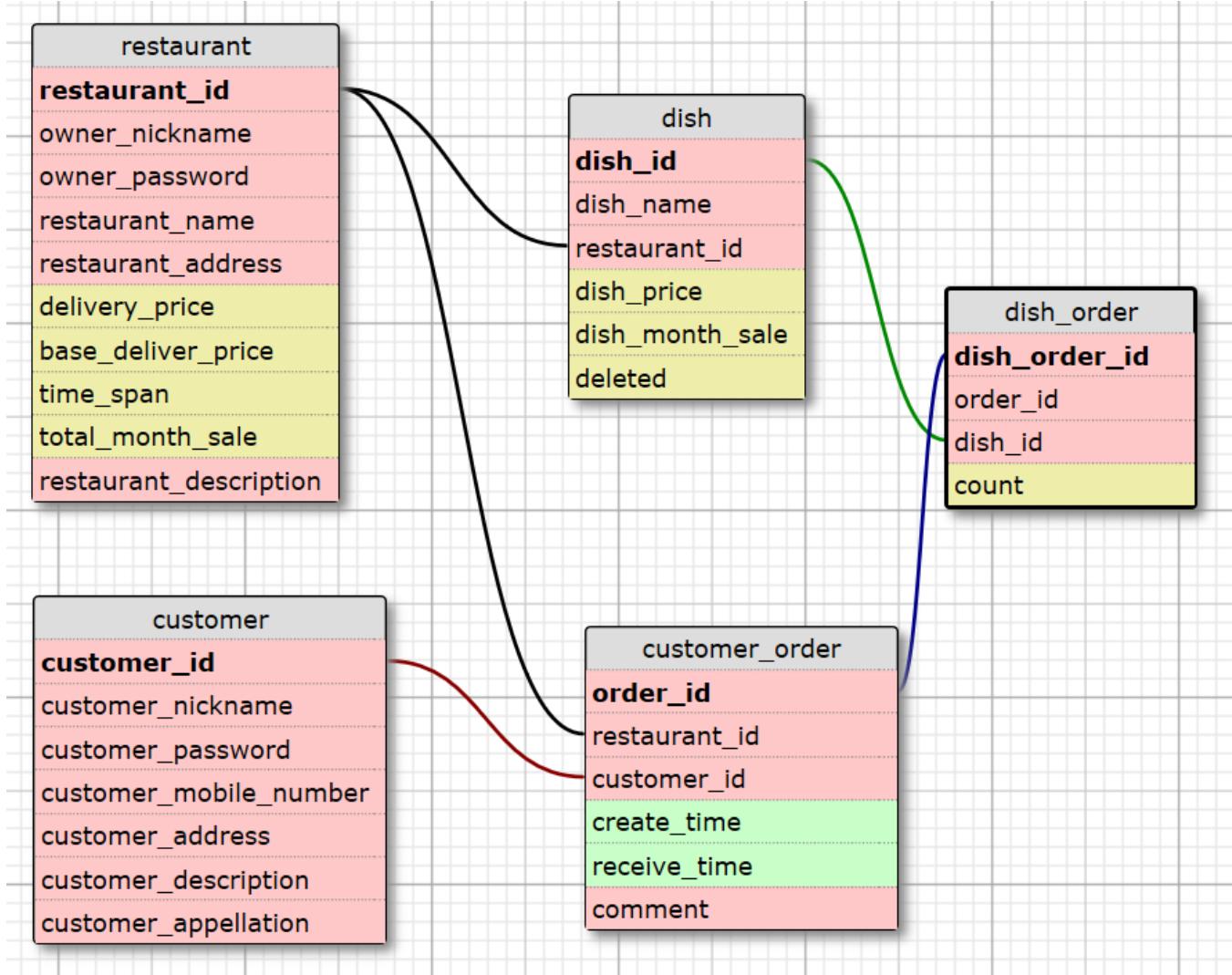


Figure 3-28: 数据库表格关系图

- 建表语句

数据库 Glutton 共包含 5 张表：商家表、用户表、菜品表、用户订单表、菜品订单表。

- 商家表信息包括：商家号、用户名、密码、商家名、商家地址、配送费、起送价、平均配送时间、营业时间、月销量、商家公告，主键为商家号。

```

CREATE TABLE restaurant(
    restaurant_id CHAR(3) NOT NULL,
    owner.nickname CHAR(20) NOT NULL UNIQUE,
    owner.password CHAR(40) NOT NULL,
    restaurant.name CHAR(50) NOT NULL,
    restaurant.address CHAR(100),
    delivery.price DECIMAL(10, 2),
    base.deliver.price DECIMAL(10, 2),
    time.span TIME,
    total.month.sale DECIMAL(10, 2),
    restaurant.description TEXT
);
    
```

```

    delivery_price DECIMAL(5,2),
    base_deliver_price DECIMAL(5,2),
    time_span SMALLINT,
    open_time CHAR(20),
    total_month_sale INTEGER,
    restaurant_description CHAR(200),
    PRIMARY KEY(restaurant_id)
);

```

- **用户表**信息包括：用户号、用户名、密码、手机号、收获地址、用户个人描述、用户称呼、用户口味偏好，主键为用户号，其中用户手机号应该是唯一的，加 UNIQUE 区分。

```

CREATE TABLE customer(
    customer_id CHAR(3) NOT NULL,
    customerNickname CHAR(20) NOT NULL,
    customer_password CHAR(40) NOT NULL,
    customer_mobile_number CHAR(20) UNIQUE,
    customer_address CHAR(100),
    customer_description CHAR(100),
    customer_appellation CHAR(20),
    customer_avatar CHAR(20) NOT NULL,
    PRIMARY KEY(customer_id)
);

```

- **菜品表**信息包括：菜品号、菜品名、所属商家号、菜品价格、菜品月销量，主键为菜品号，外键为所属商家号，即可将菜品与商家联系起来。其中，商家删除菜品后，不能影响用户历史订单对这个菜品显示，所以加 deleted 属性。当商家删除菜品后，deleted 为 true，该菜品不在商家中显示，而用户历史订单中仍然可以查询到。

```

CREATE TABLE dish(
    dish_id CHAR(6) NOT NULL,
    dish_name CHAR(30) NOT NULL,
    restaurant_id CHAR(3) NOT NULL,
    dish_price DECIMAL(5,2) NOT NULL,
    dish_month_sale SMALLINT,
    deleted BOOL,
    PRIMARY KEY(dish_id),

```

```
FOREIGN KEY(restaurant_id)REFERENCES restaurant(restaurant_id)
);
```

- 用户订单表信息包括：商家号、用户号、用户订单号、创建时间、收货时间、评论，主键为用户订单号，外键商家号、用户号分别关联商家表和用户表。

```
CREATE TABLE customer_order(
    restaurant_id CHAR(3) NOT NULL,
    customer_id CHAR(3) NOT NULL,
    order_id CHAR(3) NOT NULL,
    create_time DATETIME NOT NULL,
    receive_time DATETIME,
    comment CHAR(100),
    PRIMARY KEY(order_id),
    FOREIGN KEY(restaurant_id)REFERENCES restaurant(restaurant_id),
    FOREIGN KEY(customer_id)REFERENCES customer(customer_id)
);
```

- 菜品订单表信息包括：菜品订单号、用户订单号、菜品号、菜品数量，主键为菜品订单号，外键用户订单号、菜品号分别关联用户订单表和菜品表。这种设计使一个订单可以表示多个菜品，并减少了数据冗余。

```
CREATE TABLE dish_order(
    dish_order_id CHAR(4) NOT NULL,
    order_id CHAR(4) NOT NULL,
    dish_id CHAR(6) NOT NULL,
    count SMALLINT NOT NULL ,
    PRIMARY KEY(dish_order_id),
    FOREIGN KEY(order_id)REFERENCES customer_order(order_id) ON DELETE
        CASCADE,
    FOREIGN KEY(dish_id)REFERENCES dish(dish_id)
);
```

- 插入语句

利用 INSERT 语句向数据库中插入商家、菜品、用户、订单信息。

- 建立数据库时的数据插入

```
//插入商家信息
```

```

INSERT INTO restaurant VALUES
('001','restaurant1','password','学城粥铺','上海市浦东新区张江镇华佗
路572号',2,0,45,'09:00-23:00',743,
'需要米饭的亲们,可以在商品里面点哦,我们的炒菜是不配米饭的,除了商务套餐和
盖浇饭以外,敬请谅解');
//插入菜品信息
INSERT INTO dish VALUES
('002-10','伴鸡伴虾堡','002',19,7);

```

- 应用期间的数据插入

```

//用户在创建账号时插入: 用户, 用户名, 用户密码, 用户手机号, 用户口味偏好ID
INSERT INTO customer VALUES
('%s', '%s', '%s', '%s', NULL, NULL, NULL, '1');

//用户在创建订单时, 插入用户订单信息: 商家号、用户号、用户订单号、创建时间
INSERT INTO customer_order VALUES
('%s', '%s', '%s', '%s', NULL, NULL);

//用户在创建订单时, 插入菜品订单信息: 菜品订单号、用户订单号、菜品号、菜品
数量
INSERT INTO dish_order VALUES
('%s', '%s', '%s', '%d');

//商家在新开一家店铺时, 插入店铺信息: 商家号、用户名、密码、商家名
INSERT INTO restaurant VALUES
('%s', '%s', '%s', '%s', NULL, NULL, NULL, NULL, NULL, NULL, NULL);
//.....

```

- 查询语句

利用 SELECT 语句查询商家、菜品、订单信息。

- 商家

可以查看自己店铺的菜品和订单，并可以选择以一定的顺序（如菜品月销量，订单创建时间等）列表显示。

- 用户

用户可以查询某一店铺、菜品，按照某一顺序查看所有商家、菜品，查询自己的订单历史记录等，部分代码如下：

```

//用户模糊查询某商家
SELECT * FROM restaurant WHERE restaurant_name LIKE '%%%s%%';

```

```

//用户模糊查询某道菜品
SELECT dish_id, dish_name, dish.restaurant_id, dish_price,
       dish_month_sale, restaurant_name FROM dish, restaurant
WHERE dish_name LIKE '%%%s%%'
AND dish.restaurant_id = restaurant.restaurant_id AND NOT deleted;
//用户按照最低起送价顺序查询商家
SELECT * FROM restaurant
WHERE restaurant_name LIKE '%%%s%%'
ORDER BY base_deliver_price
//用户按照最高月销量顺序查询商家
SELECT * FROM restaurant
WHERE restaurant_name LIKE '%%%s%%'
ORDER BY total_month_sale DESC
//用户按照最低价格顺序查询菜品
SELECT dish_id, dish_name, dish.restaurant_id, dish_price,
       dish_month_sale, restaurant_name
FROM dish, restaurant WHERE dish_name LIKE '%%%s%%' AND
       dish.restaurant_id = restaurant.restaurant_id AND NOT deleted
ORDER BY dish_price
//用户按照最高月销量顺序查询菜品
SELECT dish_id, dish_name, dish.restaurant_id, dish_price,
       dish_month_sale, restaurant_name FROM dish, restaurant
WHERE dish_name LIKE '%%%s%%' AND dish.restaurant_id =
       restaurant.restaurant_id AND NOT deleted
ORDER BY dish_month_sale DESC
//.....

```

- 更改语句

利用 update 语句修改商家、菜品、用户、订单等信息。

- 商家

```

//商家修改店铺信息：商家名、商家地址、配送费、起送价，营业时间，店铺公告
UPDATE restaurant SET restaurant_name = '%s', restaurant_address =
       '%s',
       delivery_price = '%s', base_deliver_price = '%s', open_time = '%s',
       restaurant_description = '%s' WHERE restaurant_id = '%s';
//商家修改菜品信息：

```

```
UPDATE dish SET dish_price = '%f', dish_name = '%s' WHERE dish_id = '%s';
```

- 用户

```
//用户修改个人信息: 用户名, 用户地址, 用户描述  
UPDATE customer SET customerNickname = '%s', customerAddress = '%s',  
    customerDescription = '%s', customerAppellation = '%s'  
WHERE customer_id = '%s'  
//用户修改密码  
UPDATE customer SET customer_password = '%s' WHERE customer_id = '%s';  
//用户修改个人口味偏好  
UPDATE customer SET customer_avatar = '%s' WHERE customer_id = '%s';
```

- 删除语句

使用 DELETE 语句实现显式删除, UPDATE 语句结合菜品 deleted 属性实现隐式删除。

- 商家删除菜品

采用隐式删除的方式, 对于被删除的菜品, deleted 属性被置为 true, 在商家列表中不再显示, 且用户不能再查询到该菜品。但在用户历史订单中仍可以看到该菜品, 这种设计更符合实际情况。

```
UPDATE dish SET deleted = 1 WHERE dish_id = '%s'
```

- 用户删除订单

采用级联删除的方法, 对于被删除的用户订单 customer_order, 删除这一条元组的同时, 在菜品订单 dish_order 表中同时删除以该订单为外键的菜品订单。
