

目录

需求分析.....	2
整体说明.....	2
模块说明.....	3
难点分析与改进.....	12
测试帐号.....	13
用例图与流程图.....	13
未来工作.....	16
使用说明.....	16
文档说明.....	16
分工.....	16
总结与感言	17

JavaScript 课程实践 报告

袁仲贤 1900013105 信息科学技术学院

吴显宏 1900094804 信息科学技术学院

需求分析

随着北京疫情的趋势越发严重，不少的饭堂都禁止了餐内就餐，本质上的原因是希望大家减少聚集，避免疫情在社区内爆发，因此不少的人们选择以外卖的方式来维持一日三餐。在北大中，学校也配合政策的要求，取消饭堂的餐内就餐服务，并更改为打包回宿舍就餐的形式，但在等待点餐的时候，由于等待时间较长，不少同学仍需聚集在一起等候点餐，以闲聊的方式消磨时间，甚至有些同学因此摘下口罩，因此为了进一步配合禁止堂食的本质原因，所以我希望搭建一个网络点餐平台，每个同学都能进行预先点餐，并且在取餐的时候按批放好，那么便能实现快速取餐的目标。与此同时，我们也设立会员制度，以纪念品或者折扣奖励的形式鼓励同学选用此平台，再者，我们更希望为同学们设置一个反馈平台，让同学可以向平台或者饭堂的管理人员直接反映最真实的意见。

整体说明

架构：

1. 前端:bootstrap, CSS, HTML, JS
2. 后端:SQLite, Python Flask

主要模块：

1. 登入注册模块(Auth)
2. 用户模块(User)
3. 负责人模块(Chef)
4. 管理员模块(Admin)

整体情况：

1. 在本次大作业当中采用了 Python Flask 作为我们的前后端整合工具，并且采取了不同的模块进行开发，目的在于方便进行版本管理和协作开发。此外，我们还加入了对路由

的访问权限，即未登入只能访问登入注册模块的页面，已登入只能访问该用户的相关页面(如：user 只能访问用户模块中的页面)。

2. 管理员模块当中引用了 github 的 coreUi-free 模版进行修改。
<https://github.com/coreui/coreui-free-bootstrap-admin-template/>
3. 其余模块引用了 bootstrap 和 bootstrap-table
<https://bootstrap5.hexschool.com/>
<https://bootstrap-table.com/>
4. 本次大作业的所有图片都利用了 base64 进行转码，因为保存在数据库当中的都为 base64 串

模块说明

登入注册模块(route: /auth/....)

相关文件:

HTML:

template 中的 login, [AuthLayout](#), register

CSS:

bootstrap, static/css 中的 myCss

JS:

Jquery, static/js 中的 authPageJs

后端:

auth.py, dbfunc.py

实现细节:

1. AuthLayout 为其余 HTML 的母模版 (JinjaHTML 实现)
2. 登入功能使用了两个 input 接收帐号和密码，并且利用 post 请求进行验证
3. 注册功能使用了三个 input 接收帐号，密码和重覆密码，并且利用了 ajax 请求实时对三者进行验证（验证条件为帐号不能存在，密码和重覆密码相等），最后利用 post 请求提交
4. 由于密码是用户较为重要的资讯，我们使用了 AES 算法对其进行加密后，才会保存到数据库当中（AESKEY 确保了每一次加密的方法相同）

用户模块(route: /user/....)

相关文件:

HTML:

template 中的 [userLayout](#), [userHistory](#), [userOrder](#), [userProfile](#), [userFeedBack](#)

CSS:

[bootstrap](#), [bootstrap-table](#), [myCss](#), [font-awesome](#)

JS:

[Jquery](#), [static/js](#) 中的 [userFunc](#), [userHistoryJs](#), [userOrderJs](#), [userProfileJs](#), [userFeedBackJs](#)

后端:

[user.py](#), [dbfunc.py](#) 【分隔线以上】

实现细节:

1. [userLayout](#) 为其余 HTML 的母模版 (JinjaHTML 实现)
2. [userFunc.js](#) 为其余 js 文档都需要使用的函数
 - (1) 由于用户在每一个页面都能进行山登出, 购物车操作, 因此此全局 js 实现了登出请求和针对于右下方购物车按钮的功能
 - (2) 购物车功能可以对已选择菜品进行查看和删减, 以及结帐, 为了实现切换页面仍可以记录用户已选择的菜品以及数量, 我们使用了 Session 进行记录(格式为 a_b;c_d, a 类商品有 b 个, c 类商品有 d 个)。



3. [userOrder](#) 页面分为上下结构实现
 - (1) 上方为一个轮播框, 随机在数据库当中取出三个食品作为展示

- (2) 下方为一个左右结构，当我们选择了需要下单的餐厅后，便会向发出 post 请求该餐厅的所有菜品，并且可以指定某个或某几个菜品类别(不选择默认为全选)
- (3) 右方的表格使用了 bootstrapTable 进行实现(具体参数体现在 userOrderJs 中的 initTable)，考虑到当菜品过多时，从后端传到前端的数据过多导致负荷过大，所以采取了后端分步传送局部数据，每次请求都会向后端发送 offset，以得到当前 table 分页的第一项应该是那一个。并且加入了可针对价格排序的功能。
- (4) 每个显示的菜品项能够点击视图按钮打开该菜品图片，点击加入购物车进行选购



4. userHistory 页面

- (1) 同样地，使用了 bootstrap-table，实现原理与 3 无异，可以查看用户的历史订单，并且订单是按照 id 逆序排序的，原因在于用户希望最先能够看到刚下单的订单

历史订单记录

	订单号	食品	下单时间	总价	状态
-	3	鸡排饭 x1	2022-07-01 01:34:17	13	已完成
取货地点: 衣园 订单备注: terence购买					
+	2	豆腐鱼 x1 猪脑 x1	2022-07-01 01:34:17	123	没有完成

显示第 1 到第 2 条记录，总共 2 条记录

5. userProfile 页面

(1) 我们使用了一个 input 群组，接收用户希望更改的电邮，电话，以及头像

你的个人档案



帐号

terence

电邮

110@163.com

电话

110

个人相片

選擇檔案

未選擇任何檔案

更改 (请输入进行更改, 不输入默认不更改)

6. userFeedBack 页面

(1) 该页面提供了一个 textarea 和 dropDown 接收需要反馈的餐厅和反馈内容

(2) 发送后会得到一个提示

未名餐饮平台 用户订餐 查看历史订单 更改个人资料 提交反馈 积分: 0.0 等级: 0 (还需1000.0升级) 退出

提交反馈

选择餐厅

输入反馈留言

提交

通知
你的反馈已发送!!!

7. 积分和等级功能

(1) 每个用户都拥有自己的积分和等级(每 1000 提升 1 等级，最高 5 级)

(2) 当订单状态被更改为已完成时，用户才能够得到积分

积分: 0.0 等级: 0 (还需1000.0升级)

负责人模块(route: /chef/....)

相关文件:

HTML:

template 中的 chefMain, chefFeedBack

CSS:

bootstrap, bootstrap-table, myCss, font-awesome

JS:

Jquery, static/js 中的 chefJs

后端:

chef.py, dbfunc.py【分隔线以上】

实现细节:

1. chefMain 使用了 bootstrap-table 实现, 与原理与 user 的 3 无异, 差别在于可以对表项的按钮进行操作, 当点击时, 订单为变成已处理。

处理订单

	订单号	食品	下单时间	总价	状态
+	1	蕃茄鸡蛋面 x1	2022-06-30 16:39:35	8	按此完成

显示第 1 到第 1 条记录, 总共 1 条记录

2. chefFeedBack 同样也使用了 bootstrap-table 实现, 可以查看到用户发出的反馈内容, 以及透过用户的电话或电邮进行联系

查看反馈

	用户	反馈内容
-	terence	测试投诉!!!!!!
用户联络资讯: 电话: 110 电邮: 110@163.com		
+	terence	aaaa

显示第 1 到第 2 条记录, 总共 2 条记录

管理员模块(route: /admin/....)

相关文件:

HTML:

template 中的 *originalAdmin*, *addproduct*, *FoodManage*, *OrderManage*, *UserManage*

CSS:

addproduct, *simplebar*, *simplebar1*, *foodmanage*, *style*, *examples*

JS:

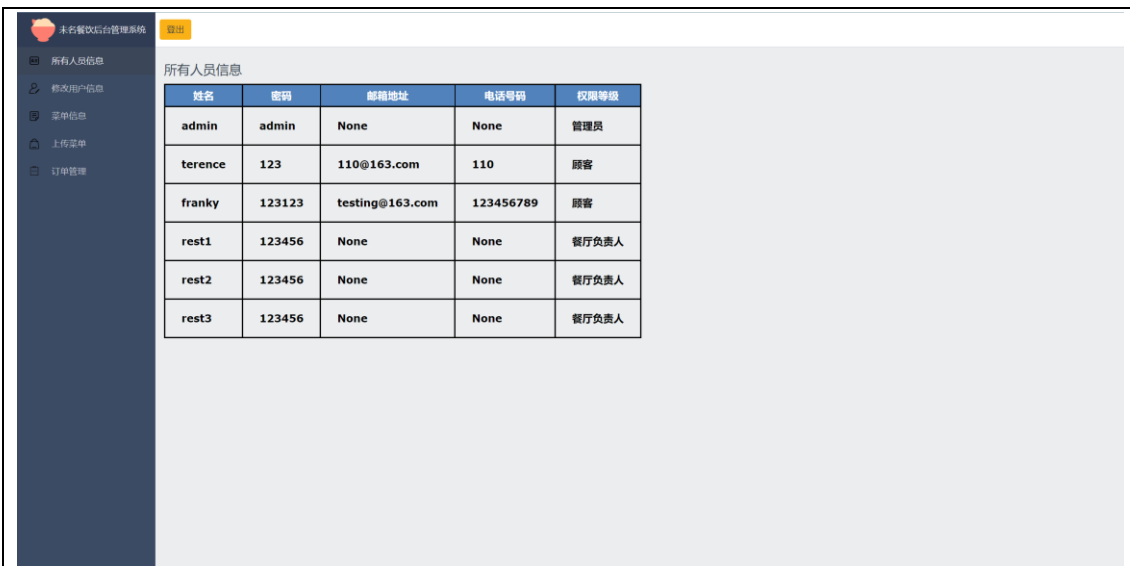
admin, *deletemenu*, *static/js* 中的

后端:

admin.py, *dbfunc.py* 【分隔线以下】

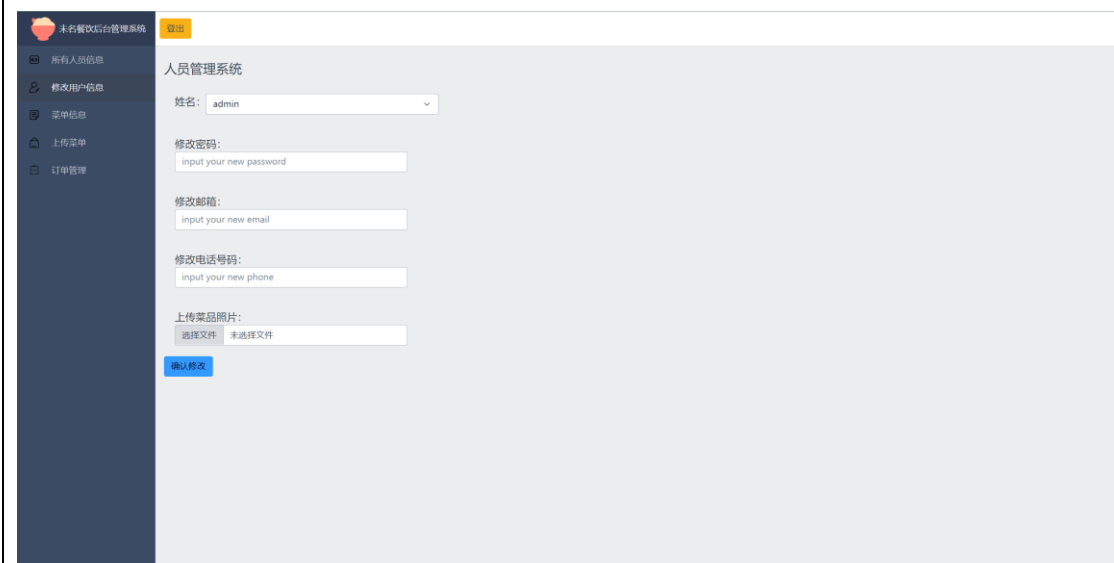
实现细节:

1. 管理员账号为 admin, 密码为 admin。
2. *originalAdmin* 作为管理员界面的母模板, 总共实现了五个不同的模块, 分别是所有人员信息界面、修改用户信息界面、菜单信息界面、上传菜单界面、订单管理界面。
3. 首先登录 admin 账号之后, 直接进入查看所有人信息界面, 如图所示





在这个界面中，顶部左上角有登出按钮，然后可以查看后台所有人员信息详细数据，**密码从数据库中通过 AES.decrypt，转回来正常的密码**，权限等级是 0-管理员，1-餐厅负责人，2-顾客，其他都是从 userInfo 表中正常读取。

4. 跳转到修改用户信息页面中，如图所示




在这个界面中，同理有登出按钮，可以在选择框中选择已经**注册的用户姓名**，首先用 onclick 函数获取当前页面中对应 userInfo 中 id 的值，然后利用 ajax 传输该 id 值到路由中，再从路由中对 userInfo 表执行更新操作，也就是对 userInfo 表中的密码、邮箱、电话号码和个人照片进行修改，其中个人照片也利用 base64 存储，便于压缩空间。

5. 跳转到菜单信息页面中，如图所示

<div>  本名餐饮后台管理系统 退出 </div> <div> <div> <div>所有人员信息</div> <div>修改用户信息</div> <div>菜单信息</div> <div>上传菜单</div> <div>订单管理</div> </div> <div>菜单信息</div> </div>					
菜品编号	菜品图片	菜品名称	菜品单价	餐厅位置	操作
1		豆腐鱼	45.0	中美园	下架
2		香辣牛肉面	10.7	中美园	下架
3		麻辣香锅	17.12	农园	下架

在这个页面中，可以随时**查看不同食堂所提供的菜品**，如果遇到了菜品不足的情况，可以选择及时**下架**的操作，也就是从 food 表中删除该食品信息。

6. 跳转到上传菜单页面，如图所示

<div>  本名餐饮后台管理系统 退出 </div> <div> <div> <div>所有人员信息</div> <div>修改用户信息</div> <div>菜单信息</div> <div>上传菜单</div> <div>订单管理</div> </div> <div>菜品信息管理</div> </div>	
菜品名称:	<input type="text" value="input the name"/>
菜品金额 (单点):	<input type="text" value="input the price"/>
类别:	<input checked="" type="radio"/> 饭 <input type="radio"/> 面 <input type="radio"/> 饮料 <input type="radio"/> 其他
饭量:	<div>勺圆</div>
上传菜品照片:	<div>选择文件 未选择文件</div>
<div>提交</div>	

在这个页面中，可以**上传食品的信息**，其中包括菜品名称、菜品单点金额、选择类别、制作的饭堂、还有食物照片，点击**提交**即可以更新到 food 表，上传后的菜品信息可以跳转到菜单信息页面中进行确认。

7. 跳转到订单管理页面，如图所示

丰名餐饮后台管理系统

退出

所有人员信息

修改用户信息

菜单信息

上传菜单

订单管理

订单信息

订单号	日期	顾客	清单	总金额(元)	留言	订单状态	操作
0	2022-06-30 16:39:35	franky	水*1,农家小炒肉饭*1	19.5	测试购物	未接受	<div>取消订单</div> <div>接受</div>
1	2022-06-30 16:39:35	franky	番茄鸡蛋面*1	8.0	测试购物	未接受	<div>取消订单</div> <div>接受</div>
2	2022-07-01 01:34:17	terence	豆腐鱼*1,猪脑*1	123.0	terence购买	订单接受	<div>取消订单</div> <div>接受</div>
3	2022-07-01 01:34:17	terence	鸡排饭*1	13.0	terence购买	未接受	<div>取消订单</div> <div>接受</div>
4	2022-07-01 23:24:18	terence	农家小炒肉饭*1	16.0	最后测试!	未接受	<div>取消订单</div> <div>接受</div>
5	2022-07-01 23:24:18	terence	鸡排饭*1,炸猪排饭*1	26.0	最后测试!	未接受	<div>取消订单</div> <div>接受</div>
6	2022-07-01 23:26:10	terence	猪脑*15	1170.0	升级测试	订单接受	<div>取消订单</div> <div>接受</div>

在这个页面中，可以查看从客户端中**发送订单**的消息，订单信息从 orderTable 表中获取，订单消息包括下单日期、顾客名字、清单、总金额、客户留言、订单状态，其中订单状态可以执行**取消订单**和**接受**操作，假如遇到恶意订单【一次性添加过多食物、留言辱骂餐厅等】，那么在后台管理系统就可以执行取消订单的操作，否则就可以代替负责人模块接受订单。

数据库设计 (web.db)

category (食物分类表)

id 主码

name 食物分类名字

food (食物表)

id 主码

name 食物名字

price 食物价格

foodPrice 食物图片

sellFrom 用户主码 (外码)

category 分类主码 (外码)

orderTable (订单表)

id 主码

orderFoodId 该订单食物的字符串 (a_b;c_d 格式, a 食物有 b 个, c 食物有 d 个)

fromUser 用户主码 (外码)

time 下单时间

comment 下单备注

sumOfPrice 总价

<i>status</i> 订单状态 <i>sellFrom</i> 餐厅主码 (外码)
restaurant (餐厅表) <i>id</i> 主码 <i>name</i> 餐厅名字 <i>location</i> 餐厅位置 <i>operator</i> 用户主码 (外码)
userFeedBack (用户反馈表) <i>id</i> 主码 <i>content</i> 反馈内容 <i>userId</i> 用户主码 (外码) <i>chefId</i> 用户主码 (外码)
userInfo (用户表) <i>id</i> 主码 <i>username</i> 用户名 <i>password</i> 密码 <i>profilePhoto</i> 个人图片 <i>email</i> 个人电邮 <i>phoneNum</i> 个人电话 <i>permission</i> 用户权限 (0 = 管理员, 1 = 负责人, 2 = 用户) <i>exp</i> 等级 <i>point</i> 积分

难点分析与改进

1. bootstrap 使用

- (1) 难点:bootstrap 是一个集成的 css,Js 工具, 通过使用 `className` 来对 Html element 的样式进行修改
- (2) 改进: 在详细阅读文档后, 发现众多 `className` 只是简化了 Style 的写法, 因此通过我们手动添加不同的 `className` 来达到我们所需要的样式

2. 购物车功能实现

- (1) 难点: 需要保存一个购物 Json (以下称为 J) 。最初, 曾考虑使用 Js 中的全局变量对其进行保存, 但由于每次加载文件都会使其全局变量刷新, 因此不可行。我们也曾考虑每次转换都伴随传送 J, 但此为一“无用功”行为。

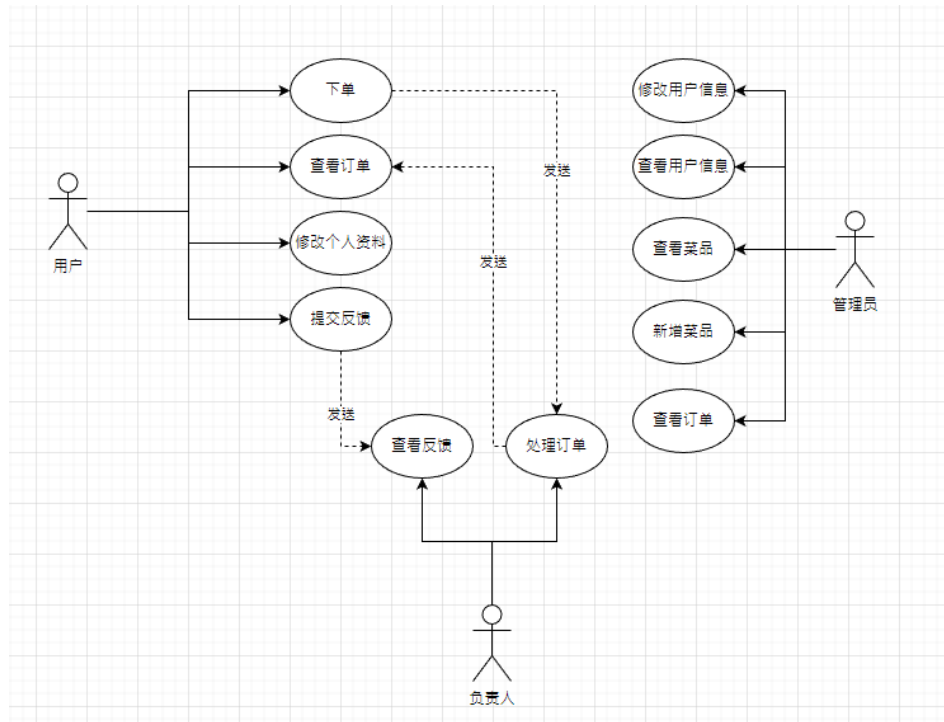
- (2) 改进：在第二个思想的下，我们采取了当加入商品时，便发起一次请求，在服务器端的 Session 保存 J,在更换页面时才请求 J,这样减少传送 J 的次数，确保是一个"有用功"行为
- 3. 分页功能实现
 - (1) 难点：需要编写一个换页的 nav。当页数过多时，会导致该 nav 非常之长。
 - (2) 改进：利用 bootstrap-table。并且使用其服务器端分页功能。
- 4. 管理员模块中需要反复对数据库更新操作，因此编写一个具有数据库的类函数很重要

测试帐号

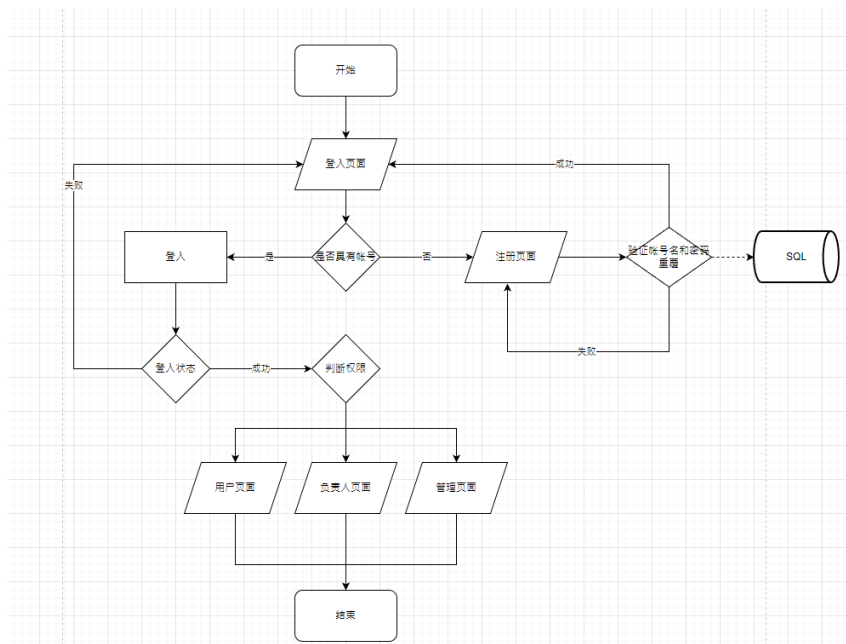
- 1. 用户测试
帐号：terence 密码：123
 - 2. 负责人测试
帐号：rest1 密码：123456
 - 3. 管理员
帐号：admin 密码：admin
- 注：用户帐号老师可选择直接注册

用例图与流程图

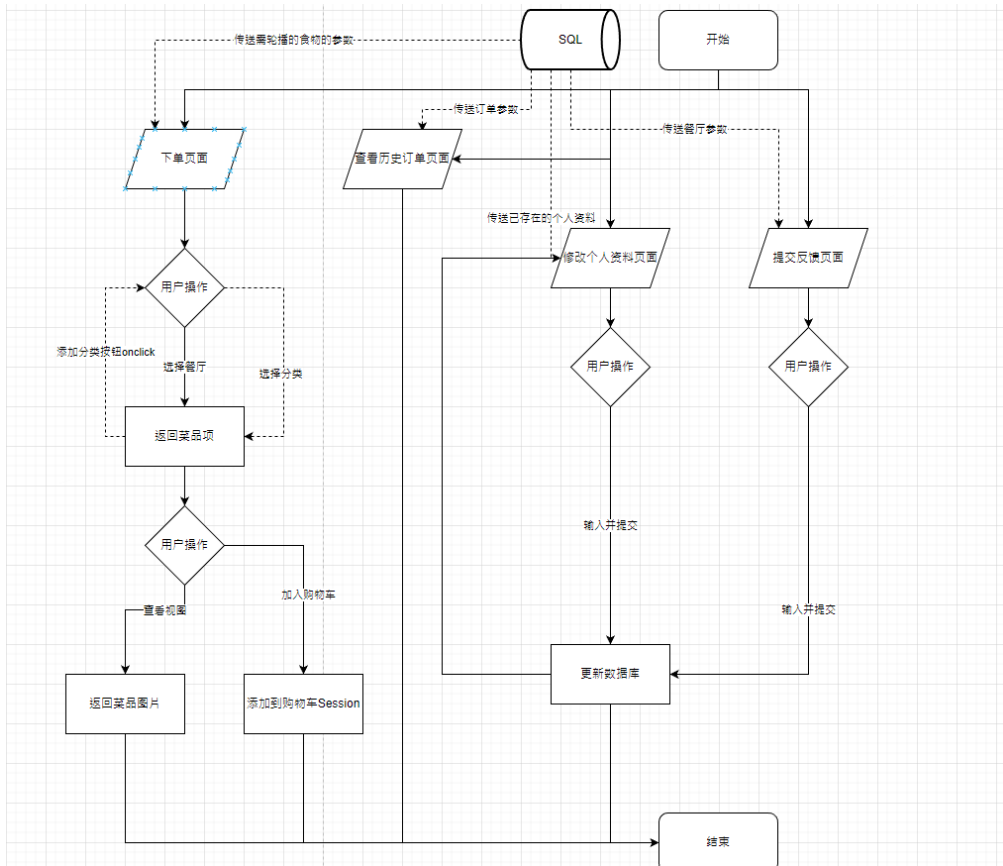
<https://app.diagrams.net/> 可以使用此网站打开文件夹中的.drawio 文件
用例图：



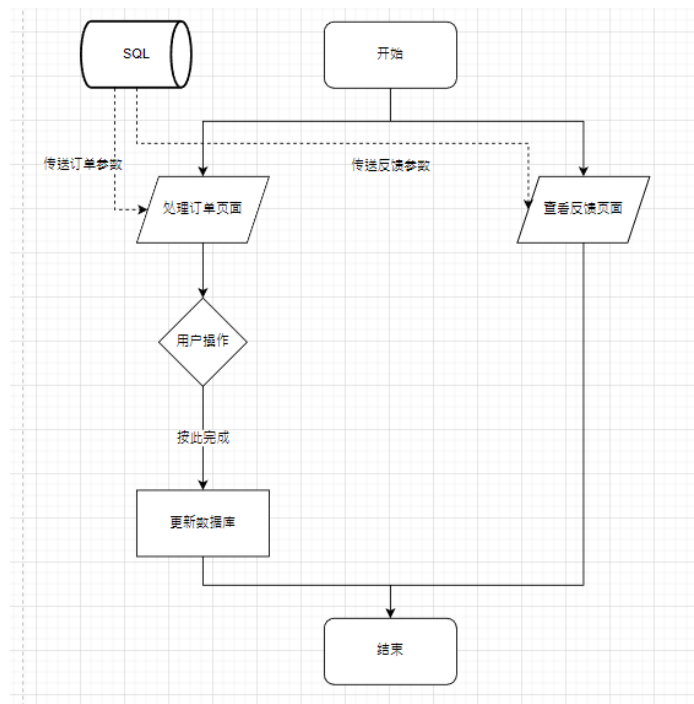
登入注册模块流程图：



用户模块流程图：



负责人模块流程图：



未来工作

1. 增加管理员的功能，包括更改用户的权限，添加食物类别，添加餐厅等功能
2. 完善用户结帐的功能，可以更仿真的结帐
3. 可以使用 bootstrap-suggest 增加用户对菜品搜索的功能

使用说明

1. 在该目录下打开终端
2. 运行 `pip install -r requirements.txt` 安装依赖包
3. 运行 `python main.py`

文档说明

static

- | **css** 存放网页中所使用到的 css 文件
- | **js** 存放网页中所使用到的 js 文件
- | **assets** 存放网页中所使用到的图片资源
- | **fonts** 存放网页中所使用到的字体资源

templates 存放不同的 html 资源

xxx.py 不同的后端文件

main.py 主文件

dbfunc.py 存放数据库函数

ENDE.py 存放 AES 加密函数

config.py 存放 flask config 的参数

requirements.txt python 依赖包

SomePic.drawio 可使用网页版的 drawio 打开，存放了用例图和流程图

updatePic.py 直接修改数据库图片函数

分工

袁仲贤：

- 登入注册模块
- 用户模块

- 负责人模块
- AES 加密函数
- 数据库设计
- 文档编写和画图

吴显宏：

- 管理员模块
- 数据库设计
- 文档编写和画图

总结与感言

在本次作业当中我们比较完善地完成了用户的功能，利用了 Js 进行网络开发，当中也涉及到不少的异步同步问题。采取了模块化的设计，减低了两者编程时版本不一至的问题。了解到不少方便网络开发的工具，例如 bootstrap, bootstrap-table 等等。有价值的地方在于，通过设计这个网站，希望提供一些实质性的东西来共同抗疫，虽然疫情在各界的配合之下开始平稳，但我们还需应该做好准备，防患于未然。此外，此类平台在学校仍然不存在，我认为未来的发展还存在着很大的空间，希望能籍此作业提供一个基本的想法。

最后，本个学期因疫情造成教学上很大的不便，从线下课转为线上课，无疑降低了不少课程的教学质量，但由于老师的语雀教案，能够更好地帮助学生们进行自习，这也是“防患于未然”的行为，非常感谢老师一学期的教导，希望课程越办越好!教安。