

**今天中午吃什么**

**技术文档**

二〇一九年六月

今天中午吃什么

技术文档

|  |  |
| --- | --- |
| 组别： | 2-11 |
| 成员： | 胡恬雨 王汝芸 张盼盼 |

# 文档说明

## 目的

本文档是今天中午吃什么Web项目的指导性文档，该文档确定并描述为完成项目目标所需的各项任务（活动）范围，确定负责制性项目各项任务的全部人员，制定各项任务的时间进度表，阐明每项任务所需的人力、物力、财力，确定每项任务的预算。

## 范围

1. 该文档描述了今天中午吃什么Web项目的功能范围，最终确定的研发计划将按照这个产品范围进行开发，以及版本控制，也适用于本系统的设计人员和开发人员；
2. 本产品需求说明书涵盖了今天中午吃什么Web项目所有的功能需求，可用性需求、可靠性需求、性能需求等的详细描述。

## 文件版本说明

本技术文档前后经历多次修改，主要修改版本及修订内容如下表1所示。

表 1 技术文档版本说明

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 发布时间 | 修订章节 | 作者 |
| 0.0.1 | 2019-04-16 | 立项背景 | 胡恬雨 |
| 0.0.2 | 2019-05-14 | 系统需求分析、数据分析 | 胡恬雨 |
| 0.0.3 | 2019-05-28 | 系统设计 | 胡恬雨 |
| 0.0.4 | 2019-06-08 | 架构分析、技术细节 | 王汝芸 |

## 名词定义和缩略语说明

本技术文档名词定义和缩略语说明如下表2所示。

表 2 名词定义和缩略语说明

|  |  |  |
| --- | --- | --- |
| 名词或缩略语 | 英文全拼 | 说明 |
| Http | HyperText Transfer Protocol | 超文本传输协议 |
| JDBC | Java DataBase Connectivity | java数据库连接 |
| Servlet | Server Applet | 小服务程序或服务连接器 |
| DAO | Data Access Object | 数据访问对象 |
| Util | utiliy | 工具包、工具组 |
| MD5 | MD5 Message-Digest Algorithm | MD5消息摘要算法 |

## 读者对象

用户、技术支持工程师、售后服务人员

## 参考资料

[1] 黑马程序员.Java Web程序设计任务教程[M].北京：人民邮电出版社，2017.

[2] 明日科技.Java Web项目开发实战入门[M].吉林：吉林大学出版社，2017.

[3] 明日科技.MySQL从入门到精通[M].北京：清华大学出版社，2017.

[4] 谭云杰.大象Thinking in UML[M].北京：中国水利水电出版社，2012.

[5] 谢乾坤.左手MongoDB，右手Redis——从入门到商业实战[M].北京：电子工业出版社，2019.

目录

[第1章 背景与意义 1](#_Toc10923731)

[1.1 立项背景 1](#_Toc10923732)

[1.1.1 现实环境 1](#_Toc10923733)

[1.1.2 发展前景 1](#_Toc10923734)

[1.2 系统的开发意义 1](#_Toc10923735)

[1.2.1 提供决策方案 1](#_Toc10923736)

[1.2.2 优化方案设计 2](#_Toc10923737)

[第2章 开发环境与开发计划 3](#_Toc10923738)

[2.1 开发环境 3](#_Toc10923739)

[2.2 开发计划 3](#_Toc10923740)

[2.2.1 开发步骤 3](#_Toc10923741)

[2.2.2 项目角色分工 3](#_Toc10923742)

[第3章 需求分析 5](#_Toc10923743)

[3.1 业务需求 5](#_Toc10923744)

[3.2 用户需求 5](#_Toc10923745)

[3.3 功能需求 5](#_Toc10923746)

[第4章 系统设计 6](#_Toc10923747)

[4.1 业务流程 6](#_Toc10923748)

[4.2 数据分析 6](#_Toc10923749)

[4.2.1 实体 6](#_Toc10923750)

[4.2.2 项目E-R模型 7](#_Toc10923751)

[第5章 数据库设计 8](#_Toc10923752)

[5.1 数据表的结构设计 8](#_Toc10923753)

[5.2 字段设计说明 8](#_Toc10923754)

[5.2.1 店铺表 8](#_Toc10923755)

[5.2.2 店铺管理员表 8](#_Toc10923756)

[5.3 触发器设计 8](#_Toc10923757)

[第6章 技术细节 9](#_Toc10923758)

[6.1 目录结构 9](#_Toc10923759)

[6.1.1 总体目录结构 9](#_Toc10923760)

[6.1.2 src目录结构 9](#_Toc10923761)

[6.1.3 Web目录结构 10](#_Toc10923762)

[6.2 框架模式 10](#_Toc10923763)

[6.2.1 M（Model） 10](#_Toc10923764)

[6.2.2 V（View） 11](#_Toc10923765)

[6.2.3 C（Controller） 11](#_Toc10923766)

[6.3 数据库接入 11](#_Toc10923767)

[6.4 其他技术说明 11](#_Toc10923768)

[6.4.1 随机点餐 11](#_Toc10923769)

[6.4.2 登录 12](#_Toc10923770)

[6.4.3 Ajax技术 12](#_Toc10923771)

[6.4.4 邮箱验证 13](#_Toc10923772)

[6.4.5 级联下拉框 14](#_Toc10923773)

[6.4.6 后台数据管理 15](#_Toc10923774)

[6.4.7 MD5数据加密 16](#_Toc10923775)

[6.4.8 防SQL注入 16](#_Toc10923776)

[6.4.9 Filter登录验证 16](#_Toc10923777)

[第7章 总结与展望 17](#_Toc10923778)

[7.1 项目开发总结 17](#_Toc10923779)

[7.2 项目后期维护与展望 17](#_Toc10923780)

[7.2.1 Redis 17](#_Toc10923781)

[7.2.2 服务器端邮件发送 17](#_Toc10923782)

[7.2.3 前端界面 17](#_Toc10923783)

[7.2.4 移动端界面 18](#_Toc10923784)

[7.2.5 数据展示 18](#_Toc10923785)

[附录 19](#_Toc10923786)

[核心SQL语句 19](#_Toc10923787)

[项目Github地址 20](#_Toc10923788)

[项目URL 20](#_Toc10923789)

# 

# 背景与意义

## 立项背景

### 现实环境

在当今快节奏、碎片化的时代，人们生活的脚步越来越快，占据人们思维的问题越来越多，越来越繁杂。作为新时代的大学生，学习专业技能只是大学生活的一部分，还有一项很重要的任务便是适应当代社会的节奏，让自己尽快摆脱“学生思维”，成为能够独当一面，自立于社会的成年人。而且大学相较于中学时期按部就班的学习生活，更为丰富，更为“不可确定”。这使得大学结合了学习与工作两方面的生活，极为丰富且灵活。而在这样充实又匆忙的生活中，一直有一个问题困扰着我们——今天中午吃什么。

### 发展前景

民以食为天。中华饮食文化博大精深、源远流长，在世界上享有很高的声誉。中国人讲吃，不仅仅是一日三餐，解渴充饥，吃的文化已经超越了“吃”本身，获得了更为深刻的社会意义。就其深层内涵来看，可以概括成四个字：精、美、情、礼。这四个字，反映了饮食活动过程中饮食品质、审美体验、情感活动、社会功能等所包含的独特文化意蕴，也反映了饮食文化与中华优秀传统文化的密切联系。

综上，“今天中午吃什么”，是一个困扰我们已久且将继续困扰我们的亘古不变的问题。而我们的系统，就解决这个问题做出了初步的回答。

## 系统的开发意义

### 提供决策方案

决策问题是指实际状态与期望状态之间存在的一种需要缩小或排除的差距。每个人在考虑午饭的选择时都会有一个期望，或是一个具体的目标，或是一个大体的方向。而现实中解决这个问题的阻碍有两个：一是有具体目标但是并不可行；二是只有大体方向但是由于对实际情况的掌握情况不好，不能很好地规划出具体目标。而我们的系统就致力于解决这两个问题，通过数据给出一个符合个人期望又符合实际情况的选择。

### 优化方案设计

工程设计中最优化问题的一般提法是要选择一组参数，在满足一系列有关的限制条件下，使设计指标达到最优值。因此，最优化问题通常可以表示为数学规划形式的问题。这就将一个实际问题转化为了一个计算机可以解决的问题，我们的系统可以在现有条件下得到一个实际情况和个人期望下的最优解。

# 开发环境与开发计划

## 开发环境

[软件开发环境](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E7%8E%AF%E5%A2%83/2298281)（Software Development Environment，SDE）是指在基本硬件和数字软件的基础上，为支持[系统软件](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F%E8%BD%AF%E4%BB%B6" \t "_blank)和应用软件的工程化开发和维护而使用的一组软件，简称SDE。它由软件工具和环境集成机制构成，前者用以支持软件开发的相关过程、活动和任务，后者为工具集成和软件的开发、维护及管理提供统一的支持。

本项目开发环境见下表2-1。

表 2‑1项目开发环境

|  |  |
| --- | --- |
| 开发环境 | 具体内容 |
| 脚本 | HTML5+CSS+Java+JavaScripts |
| 框架模式 | MVC |
| 部署环境 | Tomcat7+腾讯云Linux CentOS7 |
| 数据库 | MySQL |
| IDE | JetBrains IntelliJ IDEA |

## 开发计划

### 开发步骤

1. 调查餐厅菜品安排信息以及同学们的点餐习惯；
2. 建立数据库；
3. 完善交互页面；
4. 系统发布测试。

### 项目角色分工

对于一个成功的项目来说，角色分工的重要性不言而喻，本小组项目角色分工如下表2-2所示。

表 2‑2项目角色分工

|  |  |
| --- | --- |
| 负责方向 | 负责人员 |
| 可行性分析 | 胡恬雨 |
| 文档编撰 | 胡恬雨、王汝芸 |
| 程序设计 | 王汝芸 |
| 项目框架设计 | 王汝芸 |
| 数据库设计 | 张盼盼、王汝芸、胡恬雨 |
| 需求分析 | 胡恬雨、张盼盼 |
| 运营和维护 | 王汝芸、张盼盼 |

# 需求分析

需求分析是指理解用户需求，就软件功能与客户达成一致，估计软件风险和评估项目代价，最终形成开发计划的一个复杂过程。在这个过程中，用户的确是处在主导地位，需求分析工程师和项目经理要负责整理用户需求，为之后的软件设计打下基础。

## 业务需求

业务需求反映了组织机构或客户对系统、产品高层次的目标要求。本系统业务需求包括模块化解决点餐问题、培养用户使用习惯以及商家入驻需求。

## 用户需求

用户需求是产品执行过程中，最有难度的需求挖掘。该跟随用户还是引导用户，该如何深层次的挖掘用户需求，用户的需求该如何转换为产品需求。

本系统项目需求分析如下图3-1所示。

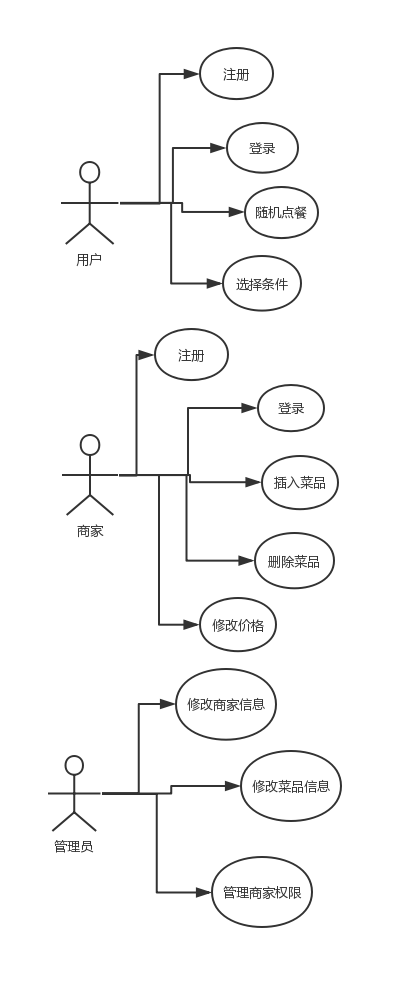
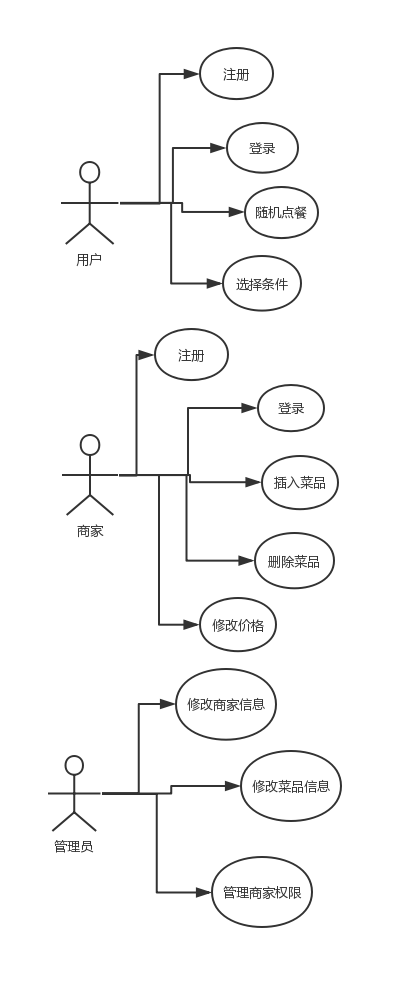
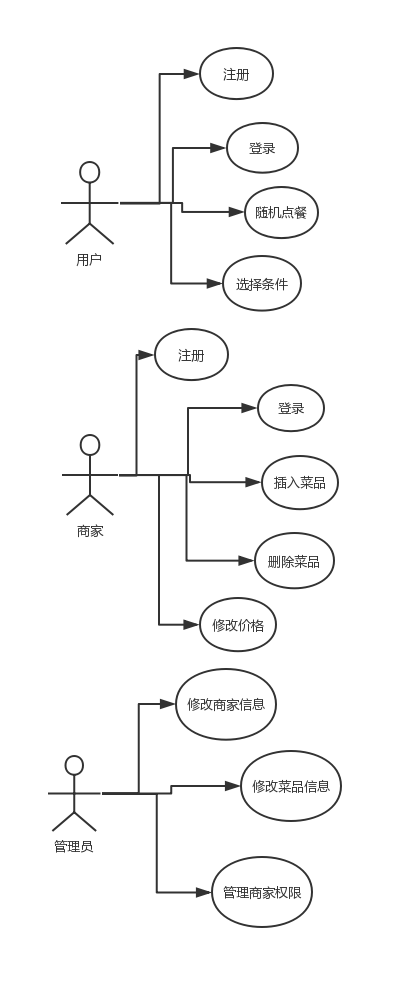


图 3‑1 项目用户需求分析

## 功能需求

功能需求规定开发人员必须在产品中实现的软件功能，用户利用这些功能来完成任务，满足业务需求。功能需求有时也被称作行为需求。

本项目功能需求包括管理员帐号的登录注册、数据查看，普通用户的使用打分、数据查看，后台数据增删改查及查询优化。

# 系统设计

## 业务流程

业务流程，是为达到特定的[价值目标](https://baike.baidu.com/item/%E4%BB%B7%E5%80%BC%E7%9B%AE%E6%A0%87/5941750" \t "_blank)而由不同的人分别共同完成的一系列活动。活动之间不仅有严格的先后顺序限定，而且活动的内容、方式、责任等也都必须有明确的安排和界定，以使不同活动在不同岗位角色之间进行转手交接成为可能。活动与活动之间在时间和空间上的转移可以有较大的跨度。

本项目业务流程图如下图4-1所示。

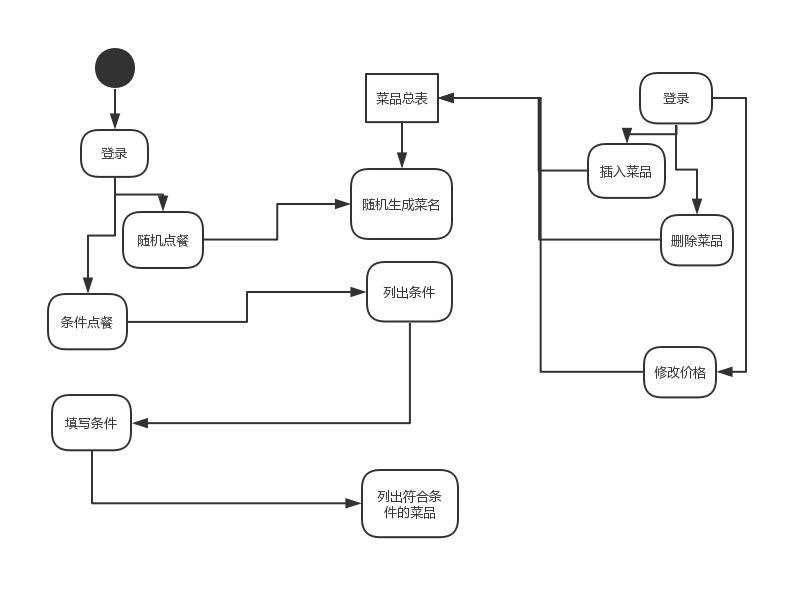


图 4‑1 业务流程图

## 数据分析

### 实体

实体是客观存在并可相互区别的事物。就数据库而言，实体往往指某类事物的集合。把每一类数据对象的个体称为实体。本项目共设计包括管理员、用户、商家三个实体。

### 项目E-R模型

ER模型，全称为实体联系模型、实体关系模型或实体联系模式图（ERD，Entity-relationship model）由美籍华裔计算机科学家[陈品山](https://baike.baidu.com/item/%E9%99%88%E5%93%81%E5%B1%B1" \t "_blank)提出，是[概念数据模型](https://baike.baidu.com/item/%E6%A6%82%E5%BF%B5%E6%95%B0%E6%8D%AE%E6%A8%A1%E5%9E%8B)的高层描述所使用的[数据模型](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%A8%A1%E5%9E%8B)或[模式图](https://baike.baidu.com/item/%E6%A8%A1%E5%BC%8F%E5%9B%BE)。

本项目E-R模型见下图4-2所示。

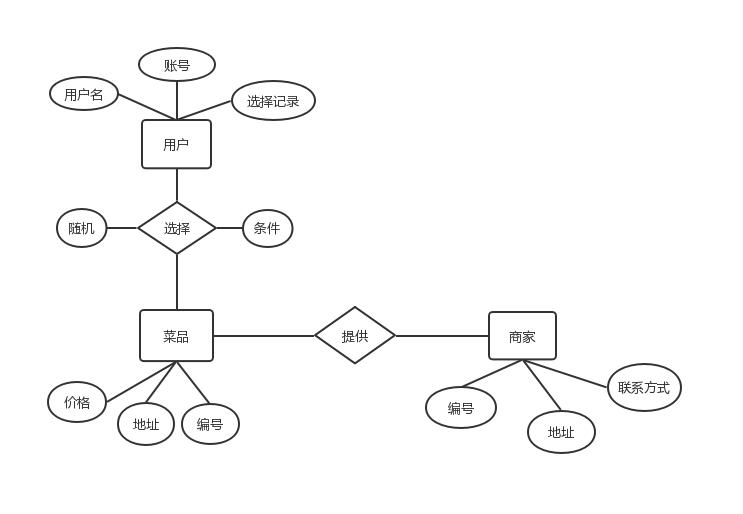


图 4‑2 E-R图

# 数据库设计

## 数据表的结构设计

本项目共创建三张数据表，分别存储实体及其属性。数据表见下表5-1、5-2及5-3所示。

表 5‑1 tb\_store

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | store\_id | store\_name | store\_loc |
| 字段含义 | 店铺编号 | 店铺名称 | 店铺地址 |
| 字段类型 | VARCHAR(9) | VARCHAR(n) | VARCHAR(n) |

表 5‑2 tb\_food

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | food\_id | food\_name | food\_price | food\_store\_id | star\_number |
| 字段含义 | 菜品编号 | 菜品名称 | 菜品价格 | 菜品所在店铺编号 | 菜品点赞数 |
| 字段类型 | INT | VARCHAR(n) | INT | CHAR(n) | INT |

表 5‑3 tb\_store\_manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | id | username | password | manager\_store\_id |
| 字段含义 | 店主编号 | 用户名 | 密码 | 所管理店铺编号 |
| 字段类型 | CHAR(9) | VARCHAR(n) | CAHR(n) | CHAR(n) |

## 字段设计说明

### 店铺表

store\_id为店铺编码，共9位。其中1—2位用于表示食堂，3—4位用于表示楼层，5—6位用于表示具体食堂（南北），7—9位用于区别具体窗口号。

### 店铺管理员表

店铺管理员id与manager\_store\_id数值相等，目前设计为一一对应关系。该设计可进行扩展，以应对日后一店多人情况。

## 触发器设计

数据表tb\_store\_manager中，manager\_store\_id指向该店铺管理员所管理的店铺，为数据表tb\_store外键，管理员与店铺为一一对应关系，当管理员账号注销时，店铺随之注销。

# 技术细节

该项目包含110个文件，总计6928行代码，初步完成了一个可行的Web项目。

## 目录结构

### 总体目录结构

项目总体目录结构如下图6-1所示。

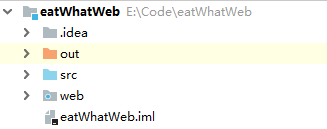


图 6‑1 项目总体目录结构

### src目录结构

项目/src结构如下图6-2所示。

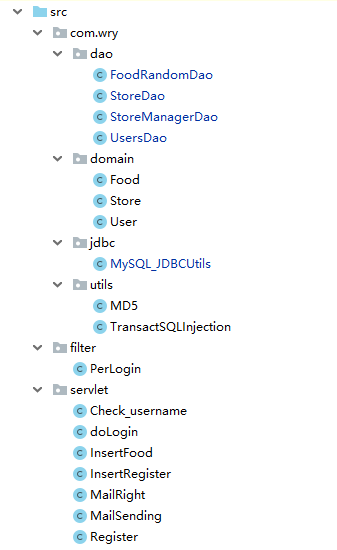


图 6‑2 src目录结构

### Web目录结构

项目/web目录结构如下图6-3所示。

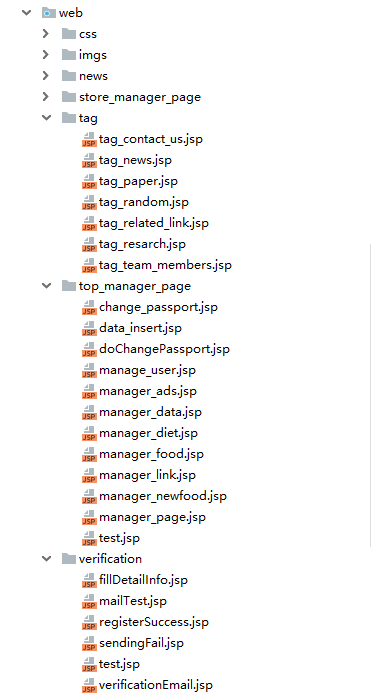
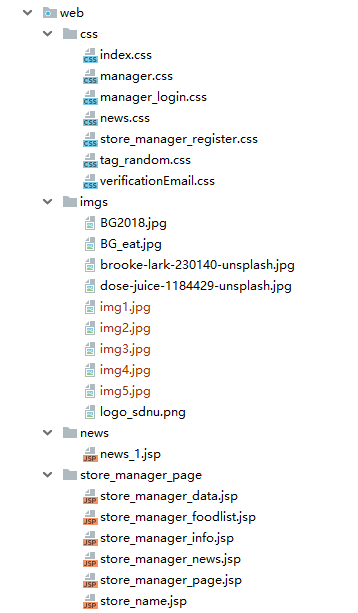


图 6‑3 Web目录结构

## 框架模式

采用MVC框架模式。[MVC](https://baike.baidu.com/item/MVC" \t "_blank)全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

### M（Model）

模型是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。在JavaWeb中，通常应用JavaBean及四种作用域作为M层。

### V（View）

视图是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。在JavaWeb中，通常采用HTML+CSS的方式进行页面的展示。

### C（Controller）

控制器是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。本项目采用Servlet作为控制器层技术。

## 数据库接入

JDBC是一种用于执行SQL语句的Java API，可以为多种关系数据库提供统一访问，它由一组用[Java语言](https://baike.baidu.com/item/Java%E8%AF%AD%E8%A8%80" \t "_blank)编写的类和接口组成。JDBC提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。采用Java与MySQL结合的方式，通过中间件JDBC进行连接，达到在java文件下直接写SQL语句即可实现对数据库操作的效果。

## 其他技术说明

### 随机点餐

随机点餐会从数据库中随机查询一条数据，并返回给前端显示，显示效果如下图6-4所示。



图 6‑4 随机点餐页面

同时随机点餐会提供无限刷新以及详细信息展示功能

1. 刷新

通过Java Scripts 下location.reload()方法实现网页刷新。

1. 详细信息

通过Java Scripts中alert弹窗实现详细信息展示，展示效果如下图6-5所示。



图 6‑5 详细信息展示

### 登录

页面底栏右侧设有管理员入口，登录页提供登录与注册两项业务。

其中，登录时会将表单传入数据与数据库进行比对，匹配则通过登录，同时在session域中记录用户名。若登录失败，会通过alert方式进行通知，并留在原地。

登录页面如下图6-6所示。

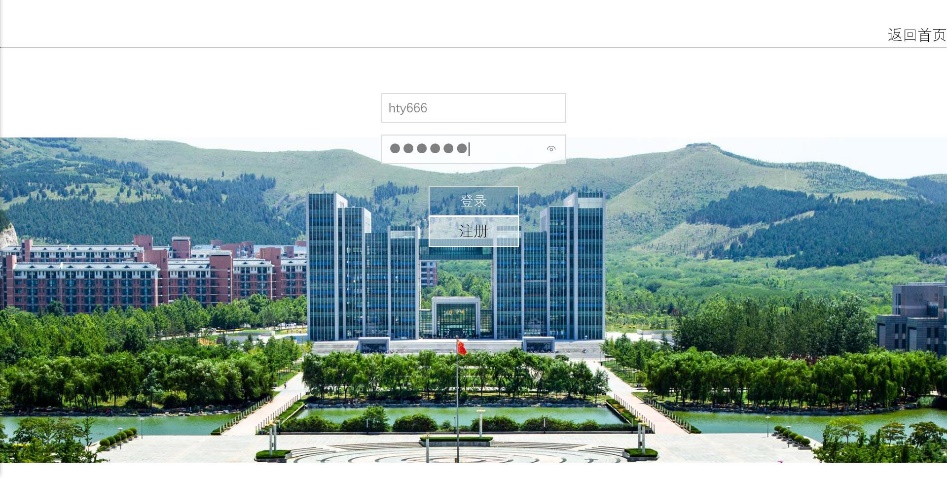


图 6‑6 登录界面

### Ajax技术

店铺可通过注册管理员的方式，实现店铺注册及管理

注册页面如下图6-7所示



图 6‑7 注册界面

其中，用户名验证通过Ajax技术，实现无需提交整个网页/表单，后台即可获取输入框值与数据库进行比对。由于整个项目未使用Spring框架，故该处Ajax通过Java Scripts与Servlet实现对数据的处理与反馈。

具体效果如下图6-8、6-9所示。



图 6‑8 用户名可以使用

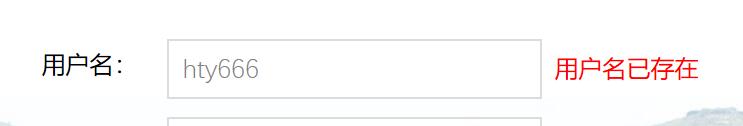


图 6‑9 用户名已存在

### 邮箱验证

后台通过接入Java邮件发送API，借助网易163服务器实现验证邮件的发送。

后台首先随机生成一组四位邮件验证码，发送邮件。验证页面填入验证码并与后台数据进行比对，若成功则进入店铺信息填写页面；若错误则留在原地提示信息；若发送失败则给出信息，返回邮箱填写页。

邮件验证页与邮件内容如下图6-10、6-11所示



图 6‑10 注册页面

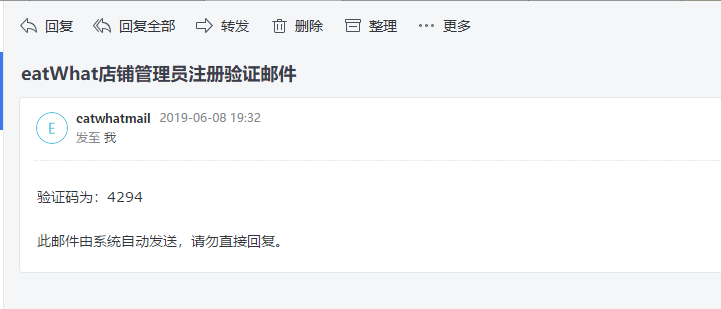


图 6‑11 验证邮件

### 级联下拉框

店铺注册页面通过Java Scripts级联下拉框实现食堂与具体位置的选择，如下图6-12、6-13所示。

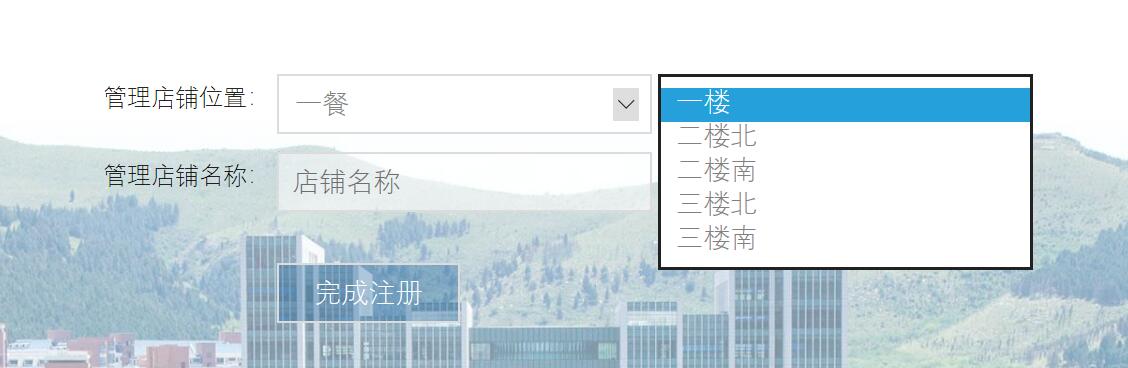


图 6‑12 级联下拉框1



图 6‑13 级联下拉框2

### 后台数据管理

店铺后台管理界面如下图6-14、6-15所示。



图 6‑14 网站高级管理员后台管理界面



图 6‑15 普通店铺管理员后台管理界面

其中，菜品管理可实现菜品数据的插入；推送管理目前未实现；数据查看如下图6-16所示。



图 6‑16 数据查看界面

店铺信息如下图所示。



图 6‑17 店铺信息查看界面

### MD5数据加密

网站密码通过MD5加密的方式存储在数据库中，避免明文存储造成数据泄露的严重事故。密码在数据库中存储信息如下图6-18所示。

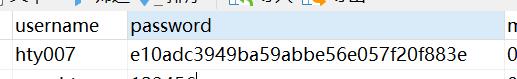


图 6‑18 MD5加密后密码在数据库中存储状态（该图中明文密码为123456）

### 防SQL注入

SQL注入攻击是Web安全史上的一个重要里程碑，它从1999年首次进入人们的视线，至今已经有十几年的历史了，虽然我们现在已经有了很全面的防范对策，但是它的威力仍然不容小觑，SQL注入攻击至今仍然是Web安全领域中的一个重要组成部分。

1. 页面在所有输入框位置前添加了一层Filter，通过java正则表达式检测、replace方法过滤异常输入。
2. 后台DAO层的所有SQL语句采用预编译语句集形式编写。

### Filter登录验证

网站在所有管理页面前都加了一层Filter，若用户未登录，即session中变量值为空，则直接拦截请求，并跳转至登录页面。

# 总结与展望

## 项目开发总结

本项目采用MVC开发模式，搭建了一个小型网站，项目基本实现了一个常规网站应有的功能，同时兼顾自身特点，设计了随机点餐模块。本项目实现了基本的前后端分离，某些短小的Java语句或JS方法选择嵌入在JSP中进行实现。项目在安全性方面作出了较为全面的考虑，实现了MD5密码加密、SQL防注入等功能。为后期继续完善提供基础保障。

本项目为解决大学生当下“今天中午吃什么”问题提供了一剂良方。这个困扰我们已久且将继续困扰我们的亘古不变的问题，由于本项目开发的完成，得到了初步的解决。

## 项目后期维护与展望

### Redis

目前项目后台数据库全部采用经典的关系型数据库MySQL实现，MySQL以强大稳定性与多平台适应性的特点为人称道，但在当下大数据的浪潮前，MySQL显得略微古板。后期拟引入强大的noSQL数据库Redis进行非重要数据的存储。Redis 是一个高性能的key-value数据库。 Redis的出现，很大程度补偿了[memcached](https://baike.baidu.com/item/memcached" \t "_blank)这类key/value存储的不足，在部分场合可以对关系数据库起到很好的补充作用。Redis支持主从同步。数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。这使得Redis可执行单层树复制。存盘可以有意无意的对数据进行写操作。由于完全实现了发布/订阅机制，使得从数据库在任何地方同步树时，可订阅一个频道并接收主服务器完整的消息发布记录。同步对读取操作的可扩展性和数据冗余很有帮助。

因此，许多经典的网站都采用Redis+MySQL进行后台数据的存储。本项目后期亦拟采取该架构，实现数据高速查询与稳定性的兼顾。

### 服务器端邮件发送

项目目前部署于腾讯云服务器，但由于网易163服务器协议原因，在服务器上无法完成邮件的发送，近期会着手解决此BUG。

### 前端界面

决定一个项目是否能被用户认可，很大的一方面是界面的美观性与实用性。本项目设计初期，借鉴了某官网布局，导致后来意识到该问题时项目已完成大半，前端的重构耗费将大量精力，因此只能在其后的每个模块尽力调整。未来如果项目反馈良好，将着手前端界面的重构。

### 移动端界面

由于使用场合的限制，“今天中午吃什么”更适合于移动端访问。因此，将尽快着手设计移动端。

### 数据展示

如何直观反映数据的变化，为商家提供简洁明了的决策支持，亦是本项目的努力方向之一。后期拟接入百度Echarts端口，为店铺管理员提供数据的查看。同时考虑接入Kibana进行数据分析，为项目开发人员提供运维、迭代方向支持。

# 附录

核心SQL语句

01 **SELECT** \* **FROM** tb\_food

02

03 **select** \* **from** tb\_food **where** food\_store\_id='"+storeID+"'

04

05 **insert into** **tb\_food** (food\_id,food\_name,food\_price,food\_store\_id,food\_pic) **values**('"

06 +food.getId()+"','"

07 +food.getFoodname() +"','"

08 +food.getFoodprice()+"','"

09 +food.getFood\_store\_id()+"','"

10 +food.getFoodpic()+"')

11

12 **SELECT** **COUNT**(\*) **FROM** tb\_food

13

14 **select** \* **from** tb\_store **where** store\_id='" + StoreID + "';

15

16 **insert into** **tb\_store** (store\_id,store\_name,store\_loc) **values**('"

17 + store.getStoreId() + "','"

18 + store.getStoreName() + "','"

19 + store.getStoreLoc() + "')";

20

21 insert into tb\_store\_manager (username,password,manager\_store\_id,email) values('"

22 +**user**.**getUsername**()+"','"

23 +**user**.**getPassword**()+"','"

24 +**user**.**getManager\_store\_id**()+"','"

25 +**user**.**getEmail**()+"')

26

27 SELECT \* FROM tb\_store\_manager

28

29 select \* from tb\_store\_manager where manager\_store\_id='"+manager\_store\_id+"'

30

31 select \* from tb\_store\_manager where username='"+manager\_name+"'

32

33 DELETE FROM tb\_store\_manager WHERE username='" + username+"'

34

35 UPDATE tb\_store\_manager SET password='"+**user**.**getPassword**()+"' WHERE username='"+**user**.**getUsername**()+"'

36

37 insert into tb\_store\_manager (id,username,password,manager\_store\_id,email) values('"

38 +**user**.**getManager\_store\_id**()+"','"

39 +**user**.**getUsername**()+"','"

40 +**user**.**getPassword**()+"','"

41 +**user**.**getManager\_store\_id**()+"','"

42 +**user**.**getEmail**()+"')

43

44 SELECT \* FROM tb\_store\_manager

45

46 select \* from tb\_store\_manager where username='"+username+"'

47

48 DELETE FROM tb\_store\_manager WHERE username='" + username+"'

49

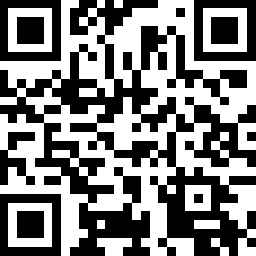
50 UPDATE tb\_store\_manager SET password='"+**user**.**getPassword**()+"' WHERE username='"+**user**.**getUsername**()+"'

51

52 select \* from tb\_store\_manager where id like'"+userID+"'

项目Github地址

<https://github.com/RuYunW/eatWhatWeb>



项目URL

[http://www.ruyunw.club/eatWha](http://www.ruyunw.club/eatWhatWeb/index.jsp)

[Web/index.jsp](http://www.ruyunw.club/eatWhatWeb/index.jsp)

