

# 基于 B/S 架构的外卖系统设计与实现

陈丹蕾 黄媛媛

四川大学锦城学院计算机与软件学院 四川 成都 611731

**【摘要】** 本文主要介绍了一个基于 B/S 架构的外卖系统的设计与实现，参考了一些常用的外卖类 app 搭建了一个外卖订购网页，该网页满足了不同年龄段人群的用餐需求，该项目提供了商品查看、收藏、购买、个人信息管理、注册、登录等功能。采用 Vue 和 Element-UI 搭建前端，后端使用 Node.js 及 Express 搭建服务器，采用 MongoDB 作为系统的数据库。

**【关键词】** 外卖系统；VueNode.js

## 前言

面对当下飞速建设与进步的新时代里，每个人忙着工作、学习，甚至利用碎片时间也在做些事情。很多人因为这样繁忙没有时间而忽略了吃饭，或者因为公司的餐食不合胃口而选择少吃。本外卖网页系统给予用户多种菜品选择，提供送餐服务，缩短用餐时间，提升生活质量。本文从主要使用技术、可行性分析、系统总体分析和系统实现设计几大部分进行介绍，展示本网页系统实现过程。

## 1 可行性分析

### 1.1 技术可行性分析

在技术方面，系统前端部分选用 Vue.js 进行搭建，更容易与其他库或项目进行整合，使用自定义接口对数据进行各种操作和使用现有的组件多次使用。相比其它的 MVVM 框架，Vue 更容易小巧且易于上手，简化了 Web 的开发也易于团队配合及维护。同时为了界面的美观以及开发效率，引入 Element UI 组件库，提升了用户体验感。

Node.js 具有强大软件包管理工具 npm，它的在线代码库包含功能齐全的代码模块<sup>[1]</sup>。而且统一了前后端的编程语言，开发人员只需掌握 JavaScript 语言就能够开发后端。数据库使用 MongoDB。MongoDB 也被称作非关系型数据库当中最像关系数据库的，是一个介于关系数据库和非关系数据库之间的产品<sup>[2]</sup>。适合实时数据的插入、修改、删除，提高项目开发的效率。

综上所述，技术可行。

### 1.2 开发模式分析

本系统使用前端与后端分别的单独开发模式，这也是当前项目的首选开发模式。运用 Vue 框架与 Element-UI 一起构建页面，后端使用 Node.js 和 Express 搭建服务器，数据库采用 MongoDB。这种模式下有众多优势，

前端只需展现页面，后端负责业务逻辑和数据的处理，这样前端即使页面有较大改动也不会影响到后端的数据，最大程度上保证了数据的信息安全性，提升了项目在开发上的效率，在分工上更明确清晰，也缩短了开发周期。用户的体验感更好。故开发模式可行。

## 2 外卖系统总体分析

### 2.1 需求分析

本外卖订餐项目主要具有用户登入与注册、订购菜品、个人中心管理、收藏管理、订单结算的功能。使用者进入网页前会要求用户进行注册，如果密码或使用名不在同一分组会有弹窗提示用户重新书写，进入系统后先填好使用者信息包括名字、地址等，若没有填写姓名、联系电话、地址则不能进行跳转并弹出提示框，填写完成后进入菜单页面，用户可选购菜品，购物车页面中可以改变菜品个数，或撤销当前菜品，选择完成后进行结算，在个人中心可以修改基本的个人数据，查看收藏菜品，展示订购的数据。一些常见问题在帮助页面查看。

### 2.2 系统架构分析

与传统点餐 app 相比，本系统无需安装，由于运用 B/S 架构进行搭建，只需浏览器就能够使用。从数据库中提取数据，提供数据服务，对数据操作，把结果返回到浏览器中。与传统的 C/S 结构相比，B/S 对数据的保护安全性更高且数据是实时性的。不必要安装专有的软件，通过系统给予的网址就可以访问，也可跨平台运行。

## 3 系统总体设计

### 3.1 功能模块设计

根据网页的需求分析即用户进行点餐、书写使用者的数据、加入购物车等操作下，系统可以划分为五个基本模块：登入与注册模块、个人中心模块、商品模块、购物车模块、结算模块。基本满足了网站用户的需求，具体的功能如图 2 所示：

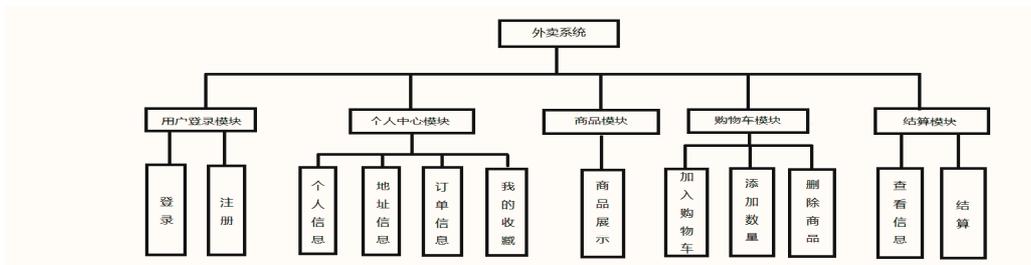


图 2 系统功能模块图

### 3.2 系统前端设计

使用 vue-router 制定不同子网页路径的跳转, 在对应的文件的 index.js 文件, 默认路径可以进入是登入页, 个人信息注册页面和导航条页面是同级路由, 在导航条页面嵌套子路由, 对应首页、购物车、个人中心、结算付款、品牌介绍子页面, 通过 router-link 进入对应子页面。在文件夹 views 中放置各个嵌套的子页面, 将注册、信息填写等几个一级路由页面放置在 components 文件夹下。当登入页面在数据比对成功后, 会跳到当前使用者的信息填写页面, 填写完成后访问到首页, 通过导航栏进而访问更多的子页。

### 3.3 系统后端设计

首先在项目新建 app.js 文件, 创建并启动 3000 端口的服务器。在本地的 MongoDB 数据库中, 使用图形化管理工具 Robomongo 创建数据库, 再以 BSON 格式插入一些数据集合如项目测试需要的用户信息, 网站菜品信息。创建数据库后, 在 app.js 文件中配合使用 Mongoose 来连接数据库, 编写对文档进行增删改查操作的几个接口供前端使用, 也有一些处理逻辑放在前端部分。

## 4 具体功能实现

### 4.1 登录模块

页面使用 element 标签页设计登录和注册页面, 当前使用者书写联系方式和密码记录登记信息接着登入。注册页面利用数据传输接口, 得到用户对应的信息, 确定按钮绑定注册函数, 利用查询接口接收已注册用户信息数据, 注册信息添加到用户信息数组中, 通过更新接口更新数据; 登录页面同理, 按钮绑定登录验证函数, 从后端查询到用户信息后, 利用循环语句对名称和密码进行判断, 若接收到登入成功信息则访问使用者的信息编写页, 若没有接收即收到错误指令, 则提示用户书写有误, 请重新填。其他静态元素由 html 标签实现, 结合 CSS 进行美化。

### 4.2 个人中心模块

个人中心页面有五子页: 我的订单、送餐地址、用户信息、菜品收藏、帮助。订单显示页面中, 用户没有订餐, 则无订购的菜品数据等, 如果订餐成功将显示对应的基本菜品信息和使用者信息。首先由查询接口接收订单数据, 若订单数据为空, 保持订单页面 display 为 none, 若接收到数据, 则设置 style 的 display 为 black, 将数据通过双向绑定渲染到页面, 订购的商品信息由 v-for 循环渲染至页面。其中订菜单中的名称、地址等在使用者基本信息在初始页面写入, 将数据传输到数据库中。

在送餐地址页面, 用户能够修改、添加、删除地址。首先结合 Element 渲染地址列表, 点击修改按钮, 弹出框 visible 状态变为 true, 显示填写信息框, 同时获得对应列表的索引, 点击确定通过索引找到对应的数据进行修改; 点击新增地址, 弹出新增弹框, 点击确定, 对应的单击操作捆绑的方法开始进行, 在地址数组中 push 方法插入最新一条地址, 同时通过接口传入数据库。

个人信息页面展示个人信息。从后端查询接收到用户的姓名、性别、联系电话、地址等信息, 初始输入框 disabled 属性为 true, 表示禁用。点击修改按钮, 触发

函数, 将禁用属性变为 false 表示可以修改, 修改后输入框失去焦点触发函数, 将现有的数据更新到数据库中。另外, 在网站进入菜品页前会先输入使用者信息, 在此页面填写名称、联系方式、地址等基本信息, 将信息放入数据库。

收藏页面中, 展示当前心仪菜品, 可以撤销喜欢。首先使用 ul 标签将从后端获得数据渲染出来, 包括商品的缩略图、名称、评分, 点击取消收藏时, 接收当前商品的 id, 运用 findIndex() 方法找到对应的数据索引, 使用 splice 方法删除, 在把修改的数据通过删除接口修改数据库中的收藏数据。帮助页面主要是针对一些用户常见问题的回答, 使用 Element 组件编写。

### 4.3 商品模块

在商品页面用户可以点单、收藏商品。首先页面从后端查询到商品信息, 存入变量数组中, 商品数组是一个二维数组, 按照分类渲染到页面中, 利用卡片组件展示商品信息, 其功能按钮包括加入购物车和加入收藏。点击加入按钮, 函数接收商品的索引, 通过索引判断是否已经在购物车列表中, 若不存在则添加进去, 若存在则找到对应的单价通过商品的数量和单价计算出总价, 将整个购物车信息到数据库中单独存储其信息。加入收藏功能与购物车类似, 也是通过索引取得当前商品信息, 不存在则将商品信息插入到收藏数组中, 并更新收藏信息。

### 4.4 购物车及结算模块

购物车页面中, 展示当前的购物信息, 能够对现有菜品进行撤销和数量管理。前端将获取到的购物信息存储, v-if 判断若菜品列表项数量为 0, 则显示没有放入任何菜品, 若有餐食存在, 则展示到页面中, 单个商品的数量默认为 1, 通过设置的增加和减少按钮改变其数量, 对应所有商品的总价随之改变。删除商品时, 通过传入的索引判断要删除的商品信息, 删除数组对应元素, 将更新的信息通过接口传入数据库。系统的结算页面, 显示用户的电话, 地址, 订购的商品, 总价及备注。点击返回, 回到上一级页面。单机结算, 可以进行进一步操作

## 5 结束语

本项目完成了外卖网站的商品信息展示、收藏、购买、个人信息管理、注册、登录等基本功能需求, 保证了多个年龄层段人群的用餐需求。网站操作简便, 清晰明了, 用户体验感良好, 有较强的实用价值。本系统也存在一些不足, 网站不具有商品评分展示及打分功能, 用户也无法对商品发表评论。在之后改进中会补充以商品详情页面, 在本页面可以对商品评分, 查看其他用户对商品的看法和建议, 也可发表对菜品的改进建议。

## 【参考文献】

- [1] 李丹清, 韩利峰, 李嘉曾, 吴丽梅, 张立园, 陈永忠. Node.js 平台下远程视频和信号监控系统的融合[J]. 仪器仪表用户, 2019, 26(03): 1-5+51.
- [2] 任明飞, 李学军, 崔蒙蒙, 杨双龙, 孙小奇. 基于 MongoDB 的非关系型数据库的设计与开发[J]. 电脑知识与技术, 2019, 15(34): 1-2. 10.14004/j.cnki.ckt.2019.4016.