

天津大学

《软件工程综合实践》 结课报告

《软件工程综合实践》 结课报告



| | | |
|---|---|-------------------|
| 学 | 院 | <u>只能与计算学部</u> |
| 专 | 业 | <u>软件工程</u> |
| 年 | 级 | <u>2022 级</u> |
| 姓 | 名 | <u>张三</u> |
| 学 | 号 | <u>3022244455</u> |

2024 年 9 月 6 日

目 录

| | | |
|-----|-----------|----|
| 第一章 | 软件需求规格说明书 | 1 |
| 1.1 | 引言 | 1 |
| 1.2 | 需要分析 | 2 |
| 1.3 | 业务分析 | 3 |
| 1.4 | 功能性需求 | 7 |
| 1.5 | 非功能性需求 | 17 |
| 第二章 | 项目设计方案 | 23 |
| 2.1 | 数据库设计 | 23 |

第一章 软件需求规格说明书

1.1 引言

1.1.1 简介

本文档的目的是详细地介绍饿了么 APP 所包含的需求，以便客户能够确认产品的确切需求以及开发人员能够根据需求设计编码，以下叙述将结合文字描述、UML 图、ER 图等来描述饿了么 APP 的功能、性能、用户界面、运行环境、外部接口以及针对用户操作给出的各种响应。本文档的预期读者有用户及客户、项目经理、开发人员以及跟该项目相关的其他竞争人员。

1.1.2 背景

1.1.2.1 市场需求的转型

- **外卖需求增长：**随着城市化进程的加快和生活节奏的加速，越来越多的人选择外卖服务，尤其是在大城市，外卖已经成为人们日常生活中不可或缺的一部分。
- **餐饮行业的数字化转型：**传统餐饮行业面临转型升级的压力，许多餐厅需要通过线上渠道来扩大顾客群并提升销售额。
- **消费者行为的改变：**年轻一代更倾向于通过移动互联网进行餐饮消费，他们希望在方便快捷的同时拥有更多的选择。

1.1.2.2 技术进步

- **移动互联网的发展：**智能手机的普及和移动互联网技术的发展为在线外卖平台的出现提供了技术基础。用户可以随时随地通过手机下单。
- **定位与导航技术：**GPS 和北斗等地图导航技术的成熟，使得在线外卖平台能够准确定位用户位置，提供快速的配送服务。
- **支付系统的完善：**电子支付的普及为在线支付和快速交易提供了便捷的途径，这进一步促进了在线外卖业务的发展。

1.1.3 定义、缩略语

| 术语 | 解释 |
|-----|--|
| 购物车 | 用户在浏览商家菜单时选择的多个菜品和商品的临时集合，用户可以在下单之前将这些选定的商品放入购物车中。 |

表 1-1 术语解释

1.1.4 约束

1.2 需要分析

1.2.1 目标

饿了么旨在为用户以及商家提供一款便捷, 高效的在线外卖平台, 支持用户在线快速浏览附近餐厅, 下单并获得及时送餐服务, 最终做到让用户享受简单且快速的订餐体验, 同时提升餐厅的订单量和运营效率. 具体而言, 饿了么最终会实现以下内容:

1. 实现用户与商家的注册行为.
2. 用户和商家能够修改自身账号信息.
3. 用户能够完成从点餐、查看购物车、修改订单地址到付款的点餐流程.
4. 商家能够完成对自身商品的上架、下架、修改操作.
5. 设置管理员登录, 管理商家信息.

1.2.2 涉众分析

本系统只涉及第 1.2.3,4 类涉众, 没有第 5 类涉众

| 序号 | 涉众 | 代表人物 | 待解决的问题/对系统的期望 |
|----|----------|------|---|
| 1 | 顾客 | 张三 | 1. 能注册账号 2. 能管理账号和个人信息 3. 能点餐, 管理购物车、送货地址, 下单支付 |
| 2 | 商家 | 李四 | 1. 能注册账号 2. 能管理商家信息 3. 能上架、下架、管理待售商品 |
| 3 | 管理员 | 王五 | 1. 能管理用户数据 |
| 4 | 审核员 | 赵六 | 1. 审核商家上架商品 |
| 5 | 配送员 (骑手) | 李七 | 1. 能获取订单信息 2. 获取商家以及用户地点进行取餐, 送餐 |

表 1-2 涉众分析表

1.2.3 范围

本系统不与实际支付接口对接, 不提供针对配送员的业务流程

1.3.1.4 顾客下单

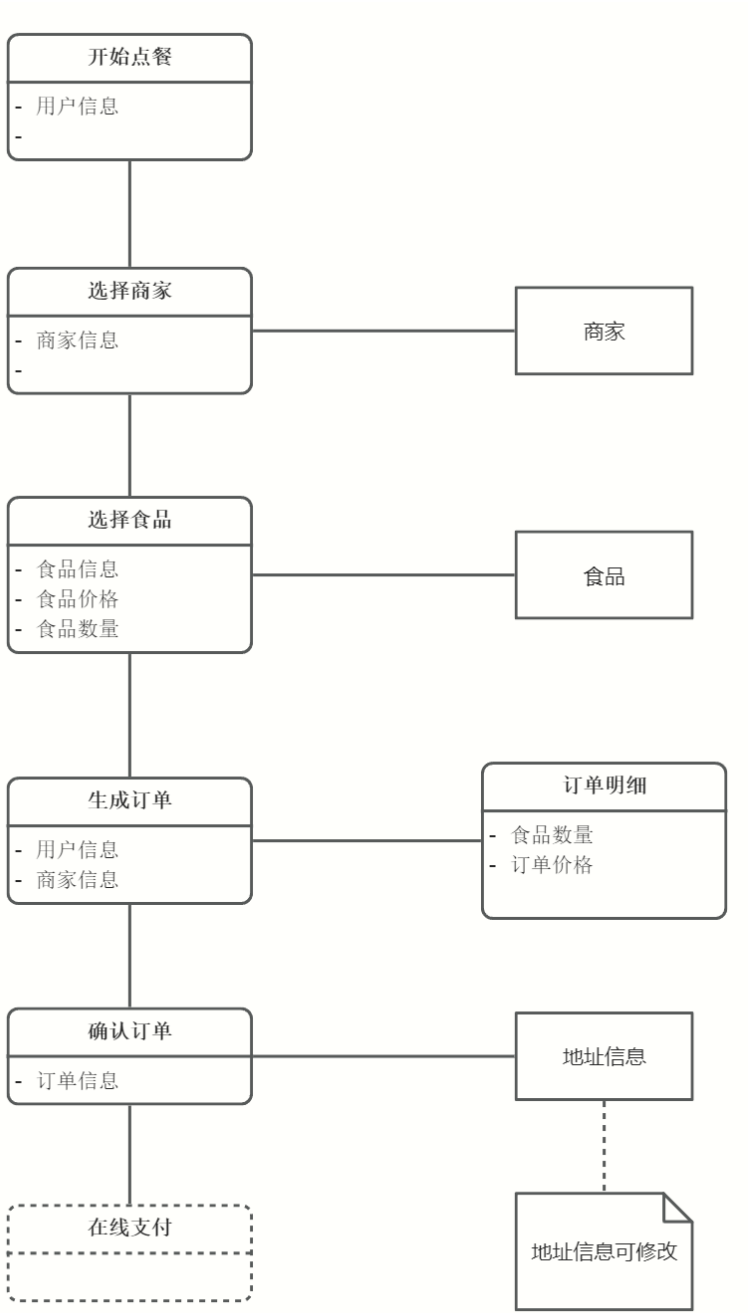


图 1-3 顾客下单

1.3.2 商家管理商品

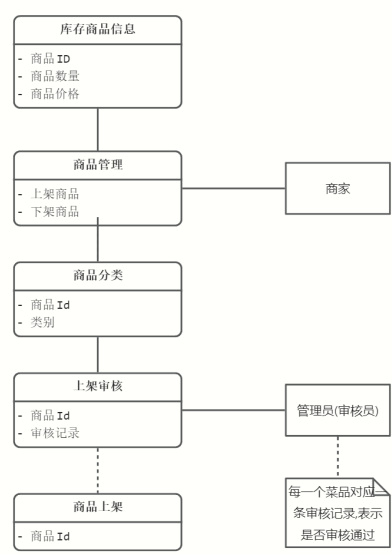


图 1-4 商家管理商品

1.3.3 业务流程分析 (UML 活动图)

1.3.3.1 用户登录流程

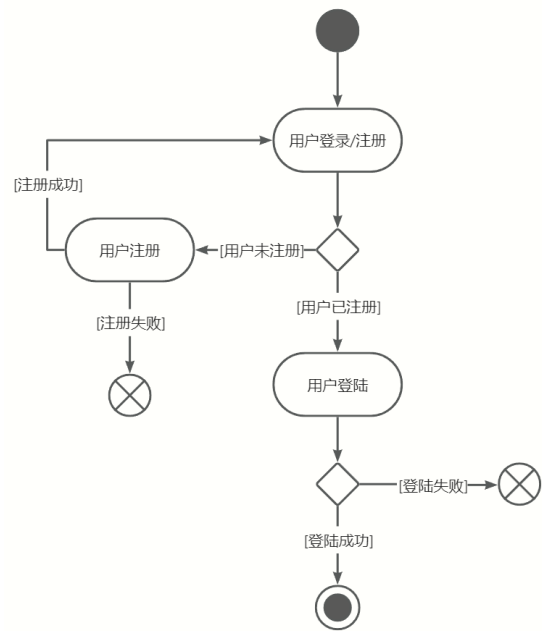


图 1-5 用户登录流程

1.3.3.2 顾客下单流程

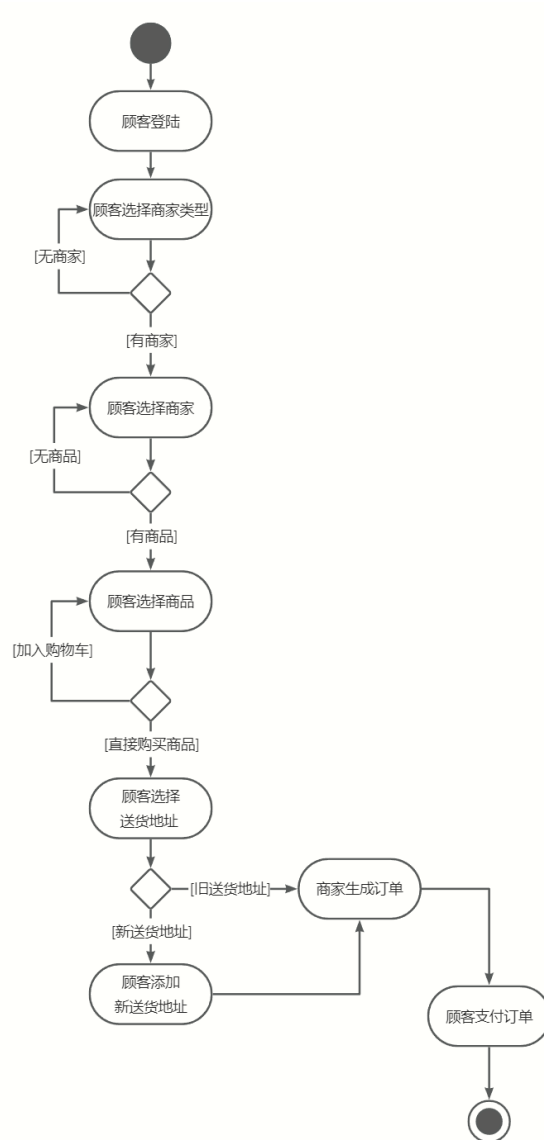


图 1-6 顾客下单流程

1.3.3.3 商家管理商品流程

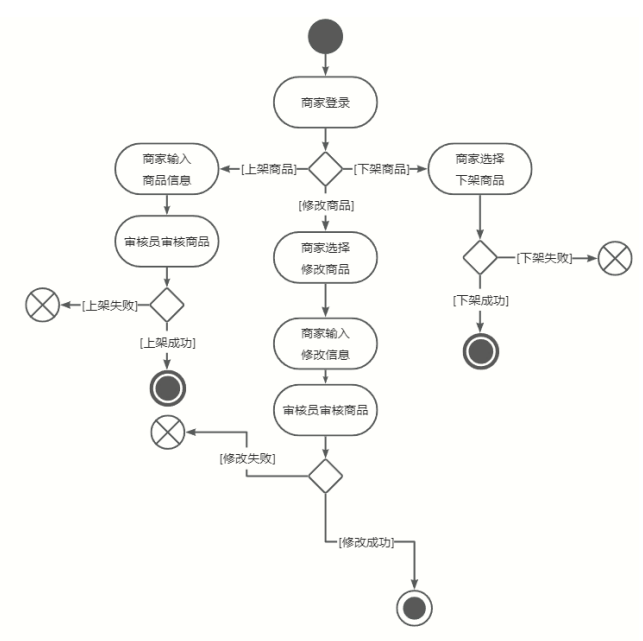


图 1-7 商家管理商品流程

1.4 功能性需求

1.4.1 执行者分析

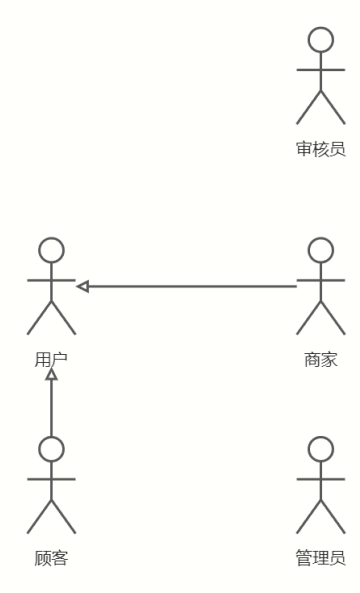


图 1-8 执行者分析

1.4.2 总用例图

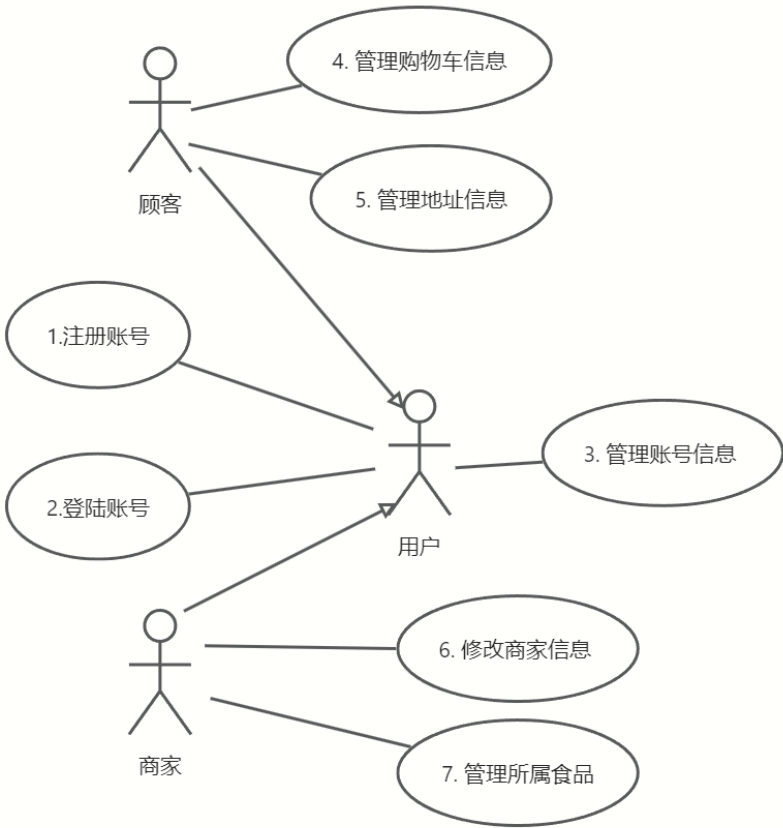


图 1-9 总用例图

1.4.3 用户的用例

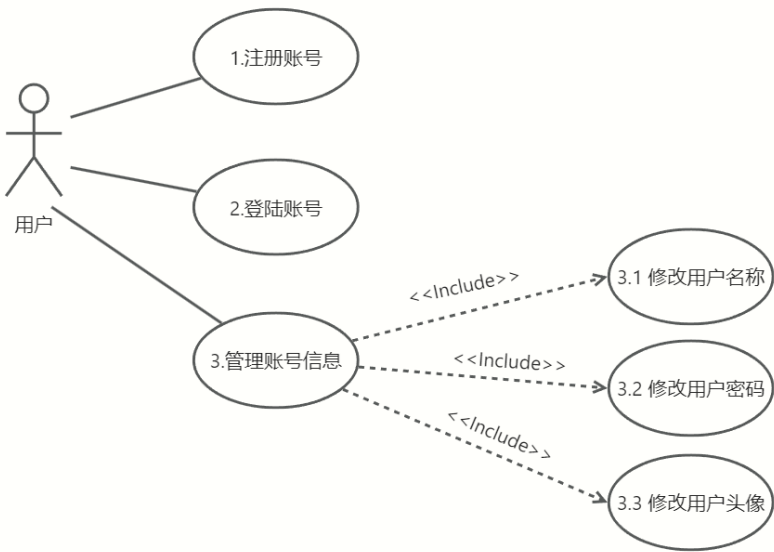


图 1-10 用户的用例

| | | | |
|------|--|-----|------|
| 编号 | 1 | 名称 | 注册账户 |
| 执行者 | 普通用户 | 优先级 | 高 |
| 描述 | 用户使用手机号进行账户注册, 设置账户密码 | | |
| 前置条件 | 输入正确的手机号码 | | |
| 基本流程 | 1. 输入手机号码 2. 输入密码 3. 显示注册成功信息 | | |
| 结束状况 | 注册成功进入登录界面 | | |
| 可选流程 | | | |
| 异常流程 | 1. 手机号验证不通过, 要求重新填写手机号 2. 用户已经注册显示用户已存在信息 | | |
| 描述 | | | |

表 1-3

| | | | |
|------|--|-----|------|
| 编号 | 2 | 名称 | 登录账户 |
| 执行者 | 普通用户 | 优先级 | 高 |
| 描述 | 用户使用手机号进行账户登录 | | |
| 前置条件 | 1. 提交正确的手机号码 2. 提交正确的密码 | | |
| 基本流程 | 1. 输入手机号码 2. 输入密码 3. 显示登录成功信息 | | |
| 结束状况 | 1. 登录成功进入主界面 (注册时) 2. 登陆成功返回上一个页面 | | |
| 可选流程 | 注册账号 | | |
| 异常流程 | 1. 手机号验证不通过, 要求重新填写手机号 2. 用户未注册进入用户注册流程 3. 密码验证不通过 | | |
| 描述 | | | |

表 1-4

| | | | |
|------|--|-----|--------|
| 编号 | 3.1 | 名称 | 修改用户名称 |
| 执行者 | 普通用户 | 优先级 | 低 |
| 描述 | 用户修改个人信息中的名称 | | |
| 前置条件 | 无 | | |
| 基本流程 | 1. 输入新用户名 2. 判断是否满足要求 3. 显示修改成功界面 | | |
| 结束状况 | 用户界面显示修改后的用户名 | | |
| 可选流程 | 1. 用户名称不符合要求，重新输入 2. 用户名称已经存在，显示用户名称已经被使用 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-5

| | | | |
|------|---|-----|--------|
| 编号 | 3.2 | 名称 | 修改用户密码 |
| 执行者 | 普通用户 | 优先级 | 低 |
| 描述 | 用户修改个人信息中的密码 | | |
| 前置条件 | 1. 旧密码输入正确 | | |
| 基本流程 | 1. 输入新密码并再次输入确保无误 2. 判断是否满足密码要求 3. 显示密码修改成功界面 | | |
| 结束状况 | 用户退出登录，需要重新输入新密码 | | |
| 可选流程 | 1. 用户密码不符合要求，要求重新输入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-6

| | | | |
|------|--|-----|--------|
| 编号 | 3.3 | 名称 | 修改用户头像 |
| 执行者 | 普通用户 | 优先级 | 低 |
| 描述 | 用户修改个人信息中的头像 | | |
| 前置条件 | | | |
| 基本流程 | 1. 导入新头像 2. 判断是否满足大小要求 3. 显示头像修改成功界面 | | |
| 结束状况 | 用户界面显示新头像 | | |
| 可选流程 | 1. 图片大小不符合要求，要求重新导入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-7

1.4.4 顾客的用例

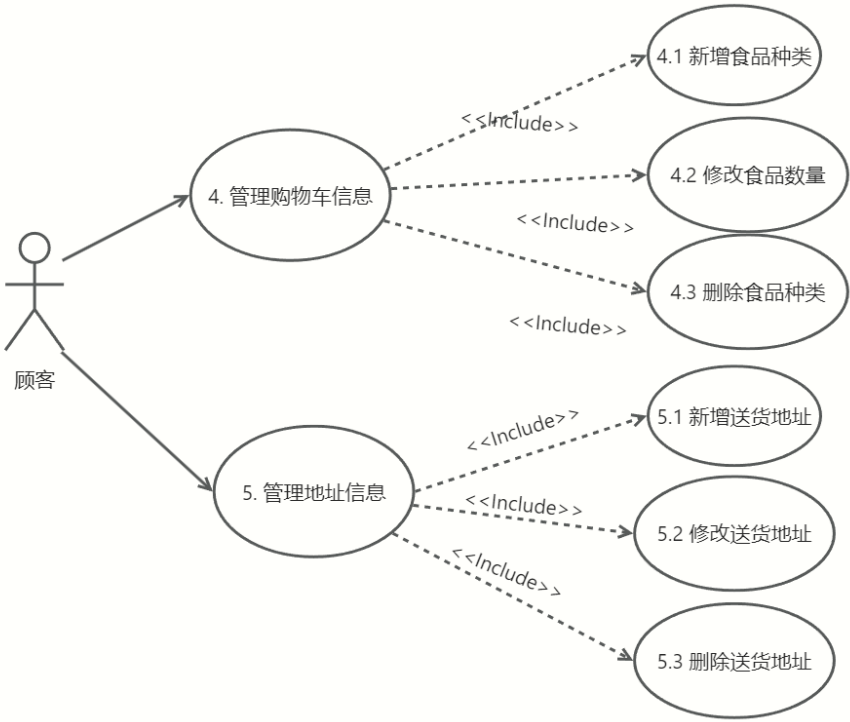


图 1-11 顾客的用例

| | | | |
|------|---|-----|---------|
| 编号 | 4 | 名称 | 管理购物车信息 |
| 执行者 | 顾客 | 优先级 | 低 |
| 描述 | 用户对购物车中商品进行新增, 修改, 删除操作 | | |
| 前置条件 | 登录用户 | | |
| 基本流程 | 1. 向购物车中添加商品 2. 修改商品的数量 3. 统计购物车内商品的总价格 | | |
| 结束状况 | 得到一个完整的购物车内容 超过起送费即可生成订单 | | |
| 可选流程 | 1. 删除购物车中的商品. | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-8

| | | | |
|------|--|-----|--------|
| 编号 | 5.1 | 名称 | 新增送货地址 |
| 执行者 | 顾客 | 优先级 | 低 |
| 描述 | 顾客新增地址列表中的送货地址 | | |
| 前置条件 | | | |
| 基本流程 | 1. 点击地址界面的新建用户地址 2. 依次填入地址信息满足正常地址规范 3. 显示新建地址成功界面 | | |
| 结束状况 | 用户界面出现新地址的简略版本 | | |
| 可选流程 | 1. 用户输入的地址不符合正常的地址规范, 需要重新输入 2. 地址中的必填信息未填写, 需要写入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-9

| | | | |
|------|--|-----|--------|
| 编号 | 5.2 | 名称 | 修改送货地址 |
| 执行者 | 顾客 | 优先级 | 低 |
| 描述 | 顾客修改地址列表中的送货地址 | | |
| 前置条件 | | | |
| 基本流程 | 1. 点击地址界面的用户地址 2. 修改地址信息满足正常地址规范 3. 显示修改地址成功界面 | | |
| 结束状况 | 用户界面出现修改后的新地址的简略版本 | | |
| 可选流程 | 1. 用户输入的地址不符合正常的地址规范，需要重新输入 2. 地址中的必填信息未填写，需要写入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-10

| | | | |
|------|---|-----|--------|
| 编号 | 5.3 | 名称 | 删除送货地址 |
| 执行者 | 顾客 | 优先级 | 低 |
| 描述 | 顾客删除地址列表中的送货地址 | | |
| 前置条件 | 1. 地址列表中有已经存在的地址 | | |
| 基本流程 | 1. 点击地址界面的用户地址，进入用户的地址列表界面 2. 选中需要删除的地址，点击删除，询问是否删除 3. 显示删除地址成功界面 | | |
| 结束状况 | 用户的地址列表界面原地址消失 | | |
| 可选流程 | 1. 误触点到删除，询问是否删除时点击否，返回用户地址列表界面 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-11

1.4.5 商家的用例

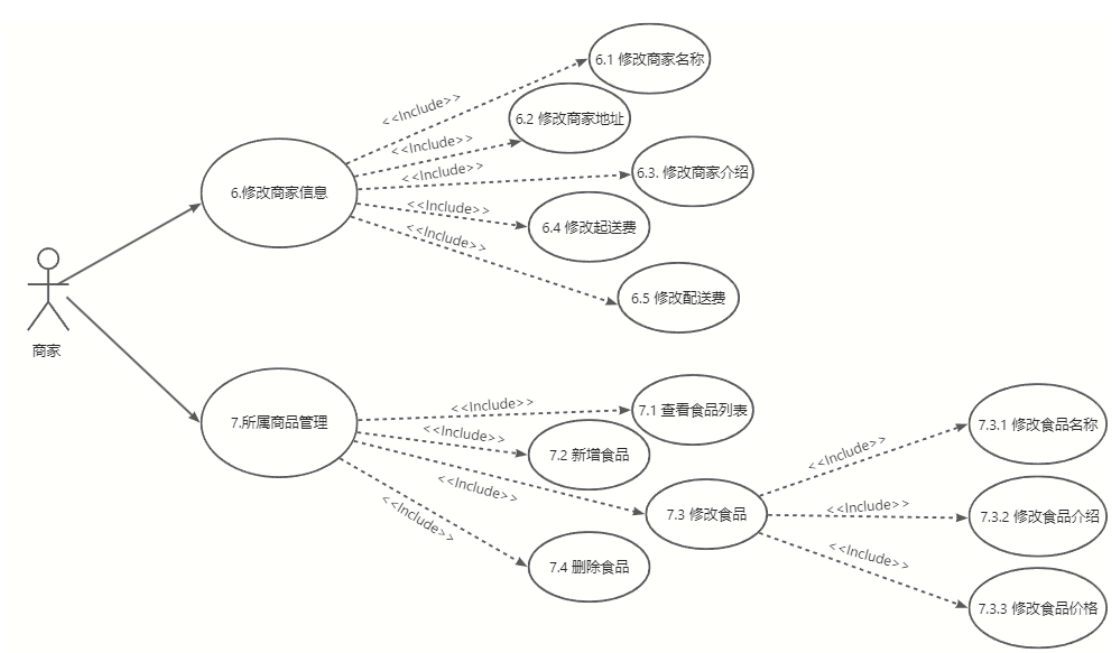


图 1-12 商家的用例

| | | | |
|------|--|-----|--------|
| 编号 | 6.1 | 名称 | 修改商家名称 |
| 执行者 | 商家 | 优先级 | 低 |
| 描述 | 商家修改商家信息中的名称 | | |
| 前置条件 | 无 | | |
| 基本流程 | 1. 输入新商家名 2. 判断是否满足要求 3. 显示修改成功界面 | | |
| 结束状况 | 商家界面显示修改后的商家名 | | |
| 可选流程 | 1. 用户名称不符合要求，重新输入 2. 用户名称已经存在，显示用户名称已经被使用 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-12

| | | | |
|------|--|-----|--------|
| 编号 | 6.2 | 名称 | 修改商家地址 |
| 执行者 | 商家 | 优先级 | 低 |
| 描述 | 商家修改商家信息中的地址 | | |
| 前置条件 | 无 | | |
| 基本流程 | 1. 输入商家的新地址 2. 判断是否满足正常的地址规范 3. 显示地址修改成功界面 | | |
| 结束状况 | 商家界面显示修改后的商家地址 | | |
| 可选流程 | 1. 商家地址不符合要求，重新输入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-13

| | | | |
|------|---|-----|--------|
| 编号 | 6.3 | 名称 | 修改商家介绍 |
| 执行者 | 商家 | 优先级 | 低 |
| 描述 | 商家修改商家信息中的介绍 | | |
| 前置条件 | 无 | | |
| 基本流程 | 1. 输入商家的新介绍 2. 判断是否满足要求（字数要求，敏感词） 3. 显示修改成功界面 | | |
| 结束状况 | 商家界面显示修改后的商家介绍 | | |
| 可选流程 | 1. 商家介绍不符合要求，重新输入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-14

| | | | |
|------|--|-----|---------|
| 编号 | 6.4 | 名称 | 修改商家起送费 |
| 执行者 | 商家 | 优先级 | 低 |
| 描述 | 商家修改商家信息中的起送费 | | |
| 前置条件 | 无 | | |
| 基本流程 | 1. 输入商家的新起送费 2. 判断是否满足要求（选择：免起送费，固定金额） 3. 显示修改成功界面 | | |
| 结束状况 | 商家下单界面显示修改后的新起送费 只要满足订单金额大于起送费时，订单才允许跳转到支付界面 | | |
| 可选流程 | 1. 商家起送费不符合要求，重新输入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-15

| | | | |
|------|--|-----|---------|
| 编号 | 6.5 | 名称 | 修改商家配送费 |
| 执行者 | 商家 | 优先级 | 低 |
| 描述 | 商家修改商家信息中的配送费 | | |
| 前置条件 | 无 | | |
| 基本流程 | 1. 输入商家新的配送费 2. 判断是否满足要求（金额要求） 3. 显示修改成功界面 | | |
| 结束状况 | 商家下单界面显示修改后的配送费 | | |
| 可选流程 | 1. 商家配送费不符合要求，重新输入 | | |
| 异常流程 | | | |
| 描述 | | | |

表 1-16

1.5 非功能性需求

1.5.1 系统架构要求

1.5.2 接口 API

business

- businesses/orderType/orderTypeId
 - 请求方法:GET
 - 参数: orderTypeId
 - 返回值: business 数组
 - 功能: 根据点餐分类编号查询分类商家信息
 - 后端函数名:listBusinessByOrderTypeId
- businesses/businessId
 - 请求方法:GET
 - 参数:businessId
 - 返回值:businessId 对象
 - 功能: 根据商家编号查询商家信息
 - 后端函数名:getBusinessById
- businesses
 - 请求方法:POST
 - 参数:businessName、businessImg、orderTypeId
 - 返回值:int(返回行数)
 - 功能: 向商家表中添加一条字段
 - 后端函数名:registerBusiness
- businesses
 - 请求方法:PUT
 - 参数:businessId、businessName、businessAddress、businessExplain、businessImg、orderTypeId、starPrice、deliveryPrice
 - 返回值:int(返回影响的行数)
 - 功能: 向商家表中更新一条记录
 - 后端函数名:updateBusiness
- 5. businesses
 - 请求方法:DELETE
 - 参数:businessId
 - 返回值:int(删除成功)
 - 功能: 根据商家编号删除一条商家记录 (将 delTag 设为 0)

- 后端函数名:deleteBusiness

food

- foods/business/businessId
 - 请求方法:GET
 - 参数:businessId
 - 返回值:food 数组
 - 功能: 根据商家编号查询所属食品信息
 - 后端函数名:listFoodByBusinessId
- foods/foodId
 - 请求方法:GET
 - 参数:foodId
 - 返回值:food 对象
 - 功能: 根据食品编号查询食品信息
 - 后端函数名:getFoodById
- foods
 - 请求方法:POST
 - 参数:foodName、foodExplain、foodImg、foodPrice、businessId、quantity
 - 返回值:int
 - 功能: 向食品表中添加一条记录
 - 后端函数名:addfood
- foods/foodId
 - 请求方法:PUT
 - 参数:businessId、foodId、foodName、foodExplain、foodImg、foodPrice、quantity
 - 返回值:int
 - 功能: 根据商家编号、食品编号更新一条食品记录（检测商家是否匹配，更新前拷贝）
 - 后端函数名:updateFood
- foods
 - 请求方法:PATCH
 - 参数:businessId、foodId、soldOut
 - 返回值:int
 - 功能: 根据商家编号、食品编号更新食品是否售罄
 - 后端函数名:setFood

- foods/foodId
 - 请求方法:DELETE
 - 参数:businessId、foodId
 - 返回值:int
 - 功能: 根据商家编号、食品编号删除一条食品记录（将 delTag 设为 0，检测商家是否匹配，删除前拷贝）
 - 后端函数名:removeFood

cart

- carts/user/userId/businessId
 - 请求方法:GET
 - 参数:userId、businessId（可选）
 - 返回值:cart 数组（多对一：所属商家信息、所属食品信息）
 - 功能: 据用户编号和商家编号，查询此用户购物车中某个商家的所有购物车信息
 - 后端函数名:listCart
- carts
 - 请求方法:POST
 - 参数:userId、businessId、foodId
 - 返回值:int
 - 功能: 向购物车表中添加一条记录
 - 后端函数名:saveCart
- carts
 - 请求方法:PUT
 - 参数:userId、businessId、foodId、quantity
 - 返回值:int
 - 功能: 根据用户编号、商家编号、食品编号更新数量
 - 后端函数名:updateCart
- carts
 - 请求方法:DELETE
 - 参数:userId、businessId、foodId（可选）
 - 返回值:int
 - 功能: 根据用户编号、商家编号、食品编号删除购物车表中的一条食品记录
 - 根据用户编号、商家编号删除购物车表中的多条记录
 - 后端函数名:removeCart

deliveryAddress

- delivery-addresses/user/userId
 - 请求方法:GET
 - 参数:userId
 - 返回值:deliveryAddress 数组
 - 功能: 根据用户编号查询所属送货地址
 - 后端函数名:listDeliveryAddressByUserId
- delivery-addresses/daId
 - 请求方法:GET
 - 参数:daId
 - 返回值:deliveryAddress 对象
 - 功能: 根据送货地址编号查询送货地址
 - 后端函数名:getDeliveryAddressById
- delivery-addresses
 - 请求方法:POST
 - 参数:contactName、contactSex、contactTel、address、userId
 - 返回值:int
 - 功能: 向送货地址表中添加一条记录
 - 后端函数名:saveDeliveryAddress
- delivery-addresses
 - 请求方法:PUT
 - 参数:daId、contactName、contactSex、contactTel、address、userId
 - 返回值:int
 - 功能: 根据送货地址编号更新送货地址信息（更新前拷贝）
 - 后端函数名:updateDeliveryAddress
- delivery-addresses/daId
 - 请求方法:DELETE
 - 参数:daId
 - 返回值:int
 - 功能: 根据送货地址编号删除一条记录（删除前拷贝）
 - 后端函数名:removeDeliveryAddress

order

- orders/user/userId
 - 请求方法:GET
 - 参数:userId

- 返回值:orders 数组（包括多对一：business；一对多：订单明细信息）
- 功能: 根据用户编号查询此用户的所有订单信息
- 后端函数名:listOrdersByUserId
- orders/orderId
 - 请求方法:GET
 - 参数:userId
 - 返回值:orders 数组（包括多对一：business；一对多：订单明细信息）
 - 功能: 根据订单编号查询订单信息，包括所属商家信息，和此订单的所有订单明细信息
 - 后端函数名:getOrdersById
- orders
 - 请求方法:POST
 - 参数:userId、businessId、daId、orderTotal
 - 返回值:int
 - 功能:
 1. 用户编号、商家编号、订单总金额、送货地址编号向订单表中添加一条记录, 并获取自动生成的订单编号
 2. 根据用户编号、商家编号从购物车表中查询所有数据, 批量添加到订单明细表中
 3. 根据用户编号、商家编号删除购物车表中的数据。
 - 后端函数名:createOrders

user

- users/login
 - 请求方法:POST
 - 参数:userId、password
 - 返回值:user 对象
 - 功能: 根据用户编号与密码查询用户信息
 - 后端函数名:getUserByIdByPass
- users/userId
 - 请求方法:GET
 - 参数:userId
 - 返回值:int
 - 功能: 根据用户编号查询用户表返回的行数

- 后端函数名:getUserById
- users/register
 - 请求方法:POST
 - 参数:userId、password、userName、userSex
 - 返回值:int
 - 功能: 向用户表中添加一条记录
 - 后端函数名:saveUser
- users
 - 请求方法:PUT
 - 参数:userId、password、userName、userSex、userImg (可选)
 - 返回值:int
 - 功能: 根据用户编号与密码更改用户信息
 - 后端函数名:updateUser
- users
 - 请求方法:DELETE
 - 参数:userId、password
 - 返回值:int
 - 功能: 根据用户编号与密码删除用户 (将 delTag 设为 0)
 - 后端函数名:deleteUser

1.5.3 安全性

1.5.4 性能

1.5.5 界面

第二章 项目设计方案

2.1 数据库设计

在原有的饿了吧 V1.0 的数据库结构基础上, 饿了吧 V2.0 对数据库进行了如下修改:

1. business 表添加 delTag 字段
2. food 表添加 delTag 字段
3. deliveryaddress 表添加 delTag 字段
4. delTag 表示删除标记, 其中 0: 正常, 1: 删除
5. food 表添加 soudOut 字段
6. user .password 成员的长度修改为 70

2.1.1 DB 一览表

| NO | 表名称 | 中文名 | 说明 |
|----|-----------------|--------|-------------------|
| 1 | business | 商家表 | 存储所有商家信息 |
| 2 | food | 食品表 | 存储每个商家拥有得食品得的所有信息 |
| 3 | cart | 购物车表 | 存储每个用户的购物车中的食品信息 |
| 4 | deliveryaddress | 送货地址表 | 存储每个用户的所有送货地址信息 |
| 5 | orders | 订单表 | 存储每个用户的所有订单信息 |
| 6 | orderdetail | 订单明细吧表 | 存储每个订单中的所有食品信息 |
| 7 | user | 用户表 | 存储所有用户的信息 |

表 2-1 数据库一览表

2.1.2 表结构

约束类型标识: PK: primary key 主键 FK: foreign key 外键 NN: not null 非空 UQ: unique 唯一索引 AI: auto increment 自增长列

删除标记: 数据类型 int, 正常:0 删除:1

点餐分类: 1: 美食、2: 早餐、3: 跑腿代购、4: 汉堡披萨、5: 甜品饮品、6: 速食简餐、7: 地方小吃、8: 米粉面馆、9: 包子粥铺、10: 炸鸡炸串

2.1.2.1 business(商家表)

| NO | 字段名 | 数据类型 | size | 默认值 | 约束 | 说明 |
|----|-----------------|------------|-------|------|----------|------|
| 1 | businessId | int | | | PK AI NN | 商家编号 |
| 2 | businessName | varchar | 40 | | NN | 商家名称 |
| 3 | businessAddress | varchar | 50 | | | 商家地址 |
| 4 | businessExplain | varchar | 40 | | | 商家介绍 |
| 5 | businessImg | mediumtext | | | NN | 商家图片 |
| 6 | orderTypeId | int | | | NN | 点餐分类 |
| 7 | starPrice | decimal | (5,2) | 0.00 | | 起送费 |
| 8 | deliveryPrice | decimal | (5,2) | 0.00 | | 配送费 |
| 9 | remarks | varchar | 40 | | | 备注 |
| 10 | delTag | int | | 0 | NN | 删除标记 |

表 2-2 商家表

2.1.2.2 food(食品表)

| NO | 字段名 | 数据类型 | size | 默认值 | 约束 | 说明 |
|----|-------------|------------|-------|-----|----------|--------|
| 1 | food | int | | | PK AI NN | 食品编号 |
| 2 | foodName | varchar | 30 | | NN | 食品名称 |
| 3 | foodExplain | varchar | 30 | | NN | 食品介绍 |
| 4 | foodImg | mediumtext | | | NN | 食品图片 |
| 5 | foodPrice | decimal | (5,2) | | NN | 食品价格 |
| 6 | businessId | int | | | FK NN | 所属商家编号 |
| 7 | remarks | varchar | 40 | | | 备注 |
| 8 | soldOut | int | | | NN | 是否售罄 |
| 9 | delTag | int | | 0 | NN | 删除标记 |

表 2-3 食品表

2.1.2.3 orders（订单表）

| NO | 字段名 | 数据类型 | size | 默认值 | 约束 | 说明 |
|----|------------|---------|-------|------|----------|-------------------|
| 1 | orderId | int | | | PK AI NN | 订单编号 |
| 2 | userId | varchar | 20 | | FK NN | 所属用户编号 |
| 3 | businessId | int | | | FK NN | 所属商家编号 |
| 4 | orderDate | varchar | 20 | | NN | 订购日期 |
| 5 | orderTotal | decimal | (7,2) | 0.00 | NN | 订单总价 |
| 6 | daId | int | | | FK NN | 所属送货地址编号 |
| 7 | orderState | int | | 0 | NN | 订单状态（0：未支付；1：已支付） |

表 2-4 订单表

2.1.2.4 orderdetailt（订单明细表）

| NO | 字段名 | 数据类型 | size | 默认值 | 约束 | 说明 |
|----|----------|------|------|-----|----------|--------|
| 1 | odId | int | | | PK AI NN | 订单明细编号 |
| 2 | orderId | int | | | FK NN | 所属订单编号 |
| 3 | foodId | int | | | FK NN | 所属食品编号 |
| 4 | quantity | int | | | NN | 数量 |

表 2-5 订单明细表

2.1.2.5 user（用户表）

| NO | 字段名 | 数据类型 | size | 默认值 | 约束 | 说明 |
|----|----------|------------|------|-----|-------|---------------|
| 1 | userId | varchar | 20 | | PK NN | 用户编号 |
| 2 | password | varchar | 70 | | NN | 密码 |
| 3 | userName | varchar | 20 | | NN | 用户名称 |
| 4 | userSex | int | | | NN | 用户性别（1：男；0：女） |
| 5 | userImg | mediumtext | | | | 用户头像 |
| 6 | delTag | int | | | NN | 删除标记 |

表 2-6 用户表