本科学生毕业论文（设计）

|  |
| --- |
| 题目(中文)： |
| **基于Flutter的在线店铺管理App的设** |
|  | **计与实现** |
| （英文）： | **The Design and Implementation of** |
|  | **Online Store Management App Based** |
|  | **on Flutter** |
| 姓 名 | **李辉** |
| 学 号 | **201606050302** |
| 院 （系） | **电子与信息工程学院** |
| 专业、年级 | **软件工程2016级** |
| 指导教师 | **何琛（讲师）** |

2020年5月6日

基于Flutter的在线店铺管理App的设计与实现

摘 要

随着电商行业的高速发展和扩张的情况下，由于电商市场资源有限的情况下，电商行业发展正处于一个发展的瓶颈阶段，寻找一条新的道路去扩大市场资源成为了电商行业先进最大的难题。受到电商冲击的线下传统零售行业也随着销售利润的逐年降低，传统零售行业也在寻找新的销售模式。马云在阿里巴巴的云栖会上所提出的“新零售”成为了电商行业和传统零售行业的突破口。新零售利用互联网的先进技术将各种商品的研发[生产](https://wiki.mbalib.com/wiki/%E7%94%9F%E4%BA%A7)、物流与[销售的全过程](https://wiki.mbalib.com/wiki/%E9%94%80%E5%94%AE%E8%BF%87%E7%A8%8B)和各个环节进行升级和技术改造，并对线上线下的服务体验以及[现代商品物流](https://wiki.mbalib.com/wiki/%E7%8E%B0%E4%BB%A3%E7%89%A9%E6%B5%81)进行高度结合。

**【关键词】**Flutter；新零售；店铺管理；Java；App；Web

**Design and Implementation of Online Store Management App Based on Flutter**

**Abstract**

With the advent of Industry 4.0,big Data has become the theme of this era.With the development of basic information and digital industry in China.

Medical care is a major event in the national economy and people’s livelihood.

【**Keywords**】Flutter;Java;App;Web;Store Management

目 录

# 1 绪论

1.1 课题研究的背景

从20世纪90年代的萌芽开始，直至今日经历了20多年的成长和发展，电子商务逐渐走向成熟。其从具体技术的应用发展到相关产业的形成，最后到产业生态圈的衍生，在互联网新时代下，新时代技术、应用的高速发展大浪潮以及国内“互联网+”战略的实施，电子商务一点点融入到国民生活习惯的各个方面。随着电子商务的普及，国民生活习惯逐渐向互联网的转移，因为电子商务在互联网交易的各个环节能够有效的降低交易成本并且有效的提高消费。然而随着电子商务高速的发展和国民消费习惯的转变，导致线下实体零售店铺遭受的了非常大的冲击。根据实体零售行业的几年的销售额和盈利率来看，整体是不断下滑的趋势，由此越来越多的线下传统零售行业的商业领袖意识到如果不寻求突破点就将面临淘汰的局面。马云在2016年阿里巴巴的云栖会上所提出的“新零售”，让各界的领袖都聚焦到这个新的概念“新零售”上，尤其是急于寻求突破口传统零售行业。这无疑是让面临困境的传统零售行业找到了新的发展方向。

新零售是一种[企业](https://wiki.mbalib.com/wiki/%E4%BC%81%E4%B8%9A)根据[互联网](https://wiki.mbalib.com/wiki/%E4%BA%92%E8%81%94%E7%BD%91)为基础，通过运用[大数据](https://wiki.mbalib.com/wiki/%E5%A4%A7%E6%95%B0%E6%8D%AE)、[人工智能](https://wiki.mbalib.com/wiki/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD)等先进的高科技技术手段，对各种商品的研发[生产](https://wiki.mbalib.com/wiki/%E7%94%9F%E4%BA%A7)、物流与[销售的全过程](https://wiki.mbalib.com/wiki/%E9%94%80%E5%94%AE%E8%BF%87%E7%A8%8B)和各个环节进行升级和技术改造，并对线上线下的服务体验以及[现代商品物流](https://wiki.mbalib.com/wiki/%E7%8E%B0%E4%BB%A3%E7%89%A9%E6%B5%81)进行高度结合的新型销售模式。简而言之，新零售就是以大数据为核心技术驱动，通过新一代的技术科技和企业[用户体验](https://wiki.mbalib.com/wiki/%E7%94%A8%E6%88%B7%E4%BD%93%E9%AA%8C)的升级，对传统零售行业的营销模式进行全面的改造升级。[[1]](#endnote-1)新零售的崛起，将引入大批的传统零售店铺商家和新的寻找机遇的新零售店铺商家，而对于店铺的管理和数据统计的需求则是这些商家急需的需求之一。

根据上述电子商务和新零售行业的分析研究，本课题就是在发展迅猛的电子商务和线下零售店铺结合的O2O模式新零售崛起的背景下衍生，为商家提供一个简单便利的店铺管理应用。因此基于Flutter的在线店铺管理App应运而生，可以为购物、订餐、零售等网络购物的多个平台提供店铺管理、商品控制、物流支持等通用功能支持。为了能够提供给用户好的体验、减少企业开发和维护成本以及提高开发速度，本系统采用Flutter + SSM开发模式。

1.2 国内外发展现状

随着国民经济收入稳步提高、互联网技术的高速发展，电子商务行业在国内的发展状态可谓是百花齐放，各自争辉。电商平台在市场上的地位已经趋于稳定，国内的大批用户已经被稳定瓜分，主要的平台由淘宝、闲鱼、天猫、苏宁易购等由阿里巴巴投资的平台；京东、拼多多、唯品会等由腾讯投资的平台。此外，网易、滴滴等也通过多种方式纷纷涉足电商行业，从用户量排名来看，淘宝网的用户量最高，其次是京东、拼多多以及唯品会。[[2]](#endnote-2)

国外电商源于美国的亚马逊书店凯里，由亚马逊开始掀起了美国电商的浪潮。[[3]](#endnote-3)因此在亚马逊的引领下，美国诸多知名网站或服务接入点都引人网络销售，美国还涌现许多网络购物平台。良好的发展优势，让美国的传统零售行业也开始逐渐向电子商务进行转变。国际上其他国家的电商行业也开始蓬勃发展，如2014年英国网络销售交易金额和互联网连接费用达到600亿英镑；2015年德国网络购物用户渗透率到达63.3%以及法国网络销售金额增长至314亿欧元。[[4]](#endnote-4)

有上述的国内外电商发展状况可以分析出，电商市场的利益瓜分正在进入一个瓶颈阶段。其原因是由于电商的特性，物流时间长，产品更加偏向于对时间要求不高的产品如衣服、保质期长的食品、家具等。现阶段大部分电商产品的使用期限是较长的，而人民需求是有限的，所以大部分的电商红利因为市场需求的限制而受限，所以寻找新的电商市场和盈利模式是电商行业当前面临的大难题，而由马云第一提出的新零售，是解决这一难题的优秀方案，许多大型公司都在新零售的发展道路上进行研究和试水。新零售的出现将会给电商行业带来重大的变革，电商行业发展朝着用户体验人性化、商品丰富、数据精准、配送效率高、价格便宜等多方面进行提升，可以用四个字来形容“多、快、好、省”。由此可以发现行业发展趋势由粗糙变得越来越精细。

1.3 课题研究的主要内容

通过对电商和新零售的现状分析和根据新零售商家的对店铺管理需求结合新型前端技术Flutter进行分析和研究。本系统主要开发一个基于Flutter的多端（主要是IOS、Android、Web端）的店铺管理平台，利用MVC的设计模型，使用多种计算模型以及数据可视化的设计方法对本系统进行开发、实现和完成。

1.4 设计组织结构

本文根据基于Flutter的店铺管理系统App的设计，主要将论文工作分为七个章节，各章的内容具体如下：

第一章 绪论，主要是介绍了课题的开发背景及意义、国内外背景、课题主要开发内容与论文结构。

第二章 相关技术简介，介绍本课题所开发系统所使用相关的技术，包含包含前后端框架、开发语言、数据库、设计模式等。

第三章 系统分析，详细介绍了整个系统功能需求、系统技术可行性分析和性能分析，并且对所使用的框架技术进行了详细的描述。

第四章 系统设计，详细对根据需求对系统进行各个层面的设计，包括总体设计原则与思路、系统架构设计、系统功能模块设计与系统数据层的设计。

第五章 系统实现，详细的介绍了系统开发和运行环境以及描述了各个功能模块的实现。

第六章 系统测试，对系统进行功能性、兼容性、安全性等多方面的测试，并且用表格来详细描述功能测试用例和其测试过程和结果。

第七章 总结与展望，总结开发过程中遇到的问题与解决方法，考虑系统拓展性，并对可提高的地方作描述。

1.5 本章小结

通过本章节的描述，阐述了网络购物背景与意义，明确了本次课题研究的主要内容，对整体研究内容进行了初步的说明，对国内外的研究作了相关的了解，同时规划了论文的整体架构和章节划分。

# 2 相关技术简介

基于Flutter的在线店铺管理App项目的实现分为前端和后端端两个子项目，下面将简要介绍实现该项目需要用到的核心技术及其优势。

2.1 Flutter简介

Flutter框架是由Google推出的，它能够让开发者使用一套代码来构建优美且性能卓越的大前端应用，大前端包含移动端（IOS、Android）和Web端，也是是Google开发的全新的Fuchsia操作系统的默认开发平台。在没有Flutter之前，跨平台开发的解决方案主要有两种方式，第一种方案是向原生应用中嵌入WebView，第二种方案则是封装原生控件并为其提供跨平台参数。然而Flutter从另一个全新角度出发，不依靠浏览器技术或者原生控件，而是依靠高性能渲染引擎Flutter Engine，以及利用Dart、C/C++和Ski来构建Widget。Flutter只需要通过使用相同的框架、渲染器和同一组Widget，就能够达到同时构建大前端应用，不需要维护多套独立代码库。[[5]](#endnote-5)

Flutter 的主要特点如下：

1. Dart开发语言。Dart语言由Google自主开发，能够很好的支持面向对象，并且学习成本低易于使用。
2. Skia渲染引擎。Android自带该引擎，在iOS中Flutter也会将该引擎打包集成到应用之中，如此来实现快速高性能的渲染。
3. 丰富的视图组件。Flutter包含Android、IOS常用UI风格组件，如Material Design、 Cupertino等常用UI风格视图组件。
4. 热部署。开发时可以实时看到效果并且进行修改。[[6]](#endnote-6)

Flutter的理念和整体架构结构的设计都极具发展潜力和想象力。对比其他可以跨平台框架，它能够确保各平台体验和实现效果的一致性，实现了真正意义上的跨平台。同时，Flutter凭借其极具特色的工具库和热加载功能，能够给开发者带来了良好的开发体验以及开发效率上的提高，并且Flutter框架的学习成本低，开发者能够快速上手使用。

2.2 SSM简介

SSM主要由Spring、SpringMVC、MyBatis三个部分组成，其框架集由Spring和MyBatis两个框架组合而成，SpringMVC是Spring中的部分内容。SpringMVC框架负责对表现层进行请求和响应操作，通过Controller控制层对其进行请求转发和数据解析。MyBatis框架根据Service服务层的业务逻辑来对数据进行封装处理。即Dao持久层的职责所在。而Spring框架则贯穿其中，它负责将SpringMVC框架、MyBatis框架进行相关的整合和协调工作，并对各个层次进行解耦，使之职责明确。

2.2.1 Spring框架

Spring是为了降低企业级应用开发复杂度而创建的一种轻量级Java开发的框架。Spring框架的主要目的是将开发过程中的各层进行分离，让开发过程中的各层能够灵活自由的选择各种组件依赖，并且还为JAVAEE开发提供集成框架。Spring能够使用Bean去实现以前只能通过EJB完成的工作。并且Spring框架不仅对服务端的开发有益处，还在松散耦合、测试性、简单性等多方面的都能够从中获取优势。Spring框架主要提供了AOP编程、IOC机制、可集成多种持久层的框架以及根据自身特性定制的MVC框架等众多优势功能。因此可知Spring框架提供的不是专注于某一层的解决方案，而是提供J2EE应用开发中各层的解决方案。[[7]](#endnote-7)

2.2.2 SpringMVC框架

SpringMVC是一种基于Java实现MVC设计模式的轻量级开源Web框架。该框架根据MVC模式将WEB层进行职能解耦，其主要职责为控制器、对象模型、处理程序、调度程序等，职责解耦有利于开发者使用每个角色能够更加灵活、减轻开发难度与复杂度以及提高开发效率。[[8]](#endnote-8)简而言之，Spring MVC框架是Spring基于MVC设计模型的用于构建WEB应用的一种实现产物。

SpringMVC框架提供MVC（Model-View-Controller）架构和现成的组件用来开发灵活、松散耦合的WEB应用。MVC模式将应用程序分离成输入逻辑、业务逻辑和UI逻辑三个方面。SpringMVC框架是围绕处理所有HTTP请求和响应的DispatcherServlet设计的。

2.2.3 MyBatis框架

MyBatis是一个开源的、轻量级的持久性框架，它是一种优于JDBC和Hibernate的替换框架。MyBatis框架可以自动执行SQL数据库与Java、.NET和Ruby on Rails中对象之间的映射，其通过将数据库语句打包在配置文件中，将映射与应用程序逻辑分离。它抽象了几乎所有的JDBC代码，并减轻了手动设置参数和检索结果的负担。它提供了一个简单的API与数据库进行交互。它还提供对自定义SQL，存储过程和高级映射的支持。[[9]](#endnote-9)

MyBatis框架将关系型数据库与面向对象的应用程序的结合使用变得更加方便且高效。对比其他关系映射工具，简单性是MyBatis框架的最大优势。要使用MyBatis，需要依赖数据对象、XML和SQL。并且开发者几乎不需要花费大量的时间就可以使用简单使用MyBatis和轻松掌握其SQL和存储过程的全部功能。

2.3 编程语言

2.3.1 Dart简介

Dart是由Google开发的针对WEB端和移动客户端开发进行专门优化的语言。Dart加强和精简数据类型、清理语法、扩充完善开发工具库，使得大前端的开发变得更加灵活和高效。

Dart的设计实现同时借鉴了Java和JavaScript。Dart在静态语法方面和Java非常相似，如类型定义、函数声明、泛型等，而在动态特性方面又和JavaScript很像，如函数式特性、异步支持等。除了融合Java和JavaScript语言之所长之外，Dart也具有一些其它具有表现力的语法，如可选命名参数、级联运算符、条件成员访问运算符等。其实，对编程语言了解比较多的读者会发现，在Dart中其实看到的不仅有Java和JavaScript的影子，它还具有其它编程语言中的身影，如命名参数在Objective-C和Swift中早就很普遍，而??操作符在PHP 7.0语法中就已经存在了，因此我们可以看到Google对Dart语言给予厚望，是想把Dart打造成一门集百家之所长的编程语言。

2.3.2 Java简介

Java是Sun Microsystems于1995年首次发布的一种面向对象高级编程语言与计算平台。Java的设计吸取了之前的高级编程语言的经验和优点，如C、C++，因此Java具有两大特征：功能强大和简单易用。Java作为面向对象高级编程语言的代表，因此很好地实现了面向对象的理论，这让开发者能够使用简单且逻辑清晰的代码来编写复杂编程。

Java具有面向对象、平台独立、平台可移植性、简单性、安全性、动态性等众多特点。其中平台独立是java的最大优点，能够增加了代码的利用率，因此同一套Java程序可以在多个不同的软硬件平台上运行[[10]](#endnote-10)。Java还能够开发桌面、分布式系统、Web等多种类型应用。

2.4 MariaDB数据库

MariaDB Server是现今最受欢迎的开源关系数据库之一。它由MySQL的原始开发人员制作，并保证保持开源。因此MariaDB继承了MySQL的众多特点，并且在社区的维护下快速迭代，拥有了更多更好的特性。[[11]](#endnote-11)并且MariaDB的使用方式与在MySQL下[几乎没有任何区别](https://mariadb.com/kb/en/mariadb-versus-mysql-compatibility/)，她们有相同的界面、命令以及拥有MySQL中的库的API，因此MariaDB与MySQL是相互兼容的，如果需要更换数据库都不需要转换。

与 MySQL 相比较，MariaDB 更强的地方在于：更多的存储引擎、更多的扩展、更快的查询处理、更少的bug、更快的运行速度、更好的功能测试等多种新型的高性能的优势。

2.5 MVC模式

MVC 是一种程序开发设计模式,它实现了显示模块与功能模块的分离，具有更好的代码复用率、低耦合度、使用简单Java逻辑实现页面、较强的可维护性、高度灵活等优势。[[12]](#endnote-12)MVC设计模式主要运用在J2EE平台上，适合Web项目开发，使用Model（模型）、View（视图）、Controller（控制器）对业务逻辑进行分离式的设计。

1. Model封装了应用程序的数据，它们通常由POJO组成。
2. View负责呈现模式数据的显示效果。
3. Controller负责处理用户请求，并且建立合适的会话，将其传递给其渲染的视图。

2.6 本章小结

本章主要对该系统开发使用的前端框架Flutter、后端框架SSM、编程语言Dart和Java 程序编程语言、数据库MariaDB和开发模式MVC做了简单的介绍。简要概述了响应技术来来源和作用，为系统的后续开发奠定了坚实的理论基础。

# 3 系统分析

3.1 系统定位与概述

基于Flutter的店铺管理App是为新零售的商家用户群体的需求设计的系统，提供了商家订单管理、商品管理、店铺管理、账户管理、数据统计五大模块。基本上将每个店家的基本需求包含在内。

3.2 功能性需求分析

基于Flutter的在线店铺App的是面向广大的商家群体，提供商家便利的、简易的App，让商家能有灵活的对订单、商品、网上店铺等网购商家群体必须的基本需求进行管理和操作。结合该系统的业务分析，本系统的功能需求主要包括订单管理模块、商品管理模块、店铺管理模块、账户管理模块和数据统计模块。

3.2.1 订单管理模块

订单管理模块主要包含浏览不同状态订单列表功能、新订单的接单/拒单功能、浏览订单详情功能、待配送订单配送功能、配送订单物流跟踪功能。用户能够快速根据订单当前不同状态进行查看订单列表，每一个订单都能够查看其详细信息。其中比较核心的操作有订单的物流操作，即在根据订单情况对订单进行发货和查看物流等功能；还需要能够对订单进行接单、拒单、修改以及删除等操作。此外，该模块也可以根据相关信息对订单进行搜索，还需要对较多的数据信息进行直接定位。

3.2.2 商品管理模块

商品管理模块主要业务是对商品进行相关的操作管理。这个模块的需要包括对商品、商品信息管理和查看商品信息等相关操作。查看商品信息包括对不同状态（在售、待售、下架）商品进行浏览。商品信息管理包括商品分类定义、商品数量等其相关的一系列信息进行编辑和修改操作。商品管理包括可以添加、删除商品、上架或者下架商品以及修改或者删除商品信息。此外，该模块需要拥有根据条件搜索商品的操作，对较多的数据信息进行直接定位。

3.2.3 店铺管理模块

店铺管理模块主要功能是店铺申请、店铺信息管理、店铺流水信息查询和店铺状态控制功能。店铺申请需要填写申请店铺的电话、地址、店铺类型等详细信息。店铺信息管理包括对店铺类型定义、店铺名称等相关信息进行修改操作。店铺流水信息查询需要对店铺的销售情况进行统计记录并显示。

3.2.4 账户管理模块

账户管理模块主要功能是账户的注册或登陆、账户个人信息管理、账户资金管理。注册或登陆功能是系统必不可少的，如果没有注册用户将无法登陆系统进行其他业务。账户个人信息管理需要账户名称、账户密码、显示头像等信息进行添加修改操作。账户管理功能能够查询账户流水、管理账户提现的银行卡进行管理控制和修改提现密码、查询提现记录等功能。

3.2.5 数据统计模块

数据统计模块主要功能是订单数据统计、交易金额数据统计、商品数据统计。系统需要将上面三个数据统计功能以图表的表现效果显示在应用上。

3.3 非功能性需求分析

本系统的非功能性需求分析将从安全性、易用性、可维护性三个方面进行分析。

1. 安全性。对于在线店铺App的安全性分析主要需求如下：一是保证用户信息的安全性。用户的个人信息应是私密的，需要妥善的监管用户重要信息尤其涉及到金钱的信息、用户电话等个人隐私信息。确保用户信息不会在访问过程中被非法获取和滥用；二是在线支付安全，需要保证用户网上支付过程中银行以及系统的支付信息的安全；三是数据可靠性，保证双方数据安全可用即使在交易过程发生意外中断时或者其他异常需要确保尽快恢复数据。
2. 易用性。易用性需要注重用户体验即是一种以使用者为中心的设计概念，好的用户体验能直接给企业带来直观的收益。[[13]](#endnote-13)本系统的易用性需求提现主要是设计思想，系统的整体效果设计会遵统一的颜色主题，做到突出重要信息。系统的布局将会根据客户的习惯和易用的原则进行设计与实现，并且让界面简洁有条理的展示给顾客使用。
3. 可维护性。该系统会随着用户需求和复杂的业务逻辑需要对系统进行复杂的维护，为了便于日后的维护和运营，所以需要采用自动化运维工具。该工具主要的两个任务是新版本的发布和日常的运维。

3.3 可行性分析

3.3.1 经济可行性分析

本系统的经济可行性分析主要从项目成本、项目收益、人民生活等方面进行分析。

1. 项目成本降低。本系统采用的技术，能够使开发和维护成本大大降低。Flutter的设计理念是实现了一套代码部署于Mac OS、Windows、Linux、Android、iOS以及嵌入式多个平台，大大减少了开发、维护的工作量，实现了真正的“跨平台”。这样能够减少大量的开发和维护人员，因此其也就成本大大降低了。

2. 项目收益提高。基于新零售模式的大浪潮，线下众多的传统零售商家将进入新零售的队伍中，而其中线上店铺管理的需求将会大大提升，该App将满足商家需求，搭建一个良好的生态环境和便利简洁的操作，以此来引进许多的商家入住平台，就能获取更多的用户，来增加收益。

3. 提高人民生活水平。未来可期的新零售模式将会给电商行业带来大的变革，让消费者生活更加便利，也能提供商家更多的创业机会，比如说由于商品找不到销售途径的偏僻地区农民或厂家和受到电商冲击的线下传统零售商家。

3.3.2 技术可行性分析

基于Flutter的在线管理App分为前后端两个子项目。本项目将从前后端项目的开发技术进行可行性分析。

前端主要采用Flutter框架，虽然该框架是比较新型的框架，但是已经有很多大公司比如阿里巴巴、腾讯等都在研究应用该技术，并且已经市场上有采用该技术框架的产品如咸鱼、今日头条、美团、饿了么等产品都应用了该框架。而且Flutter框架已经有了稳定的开发版本，可以着手使用。并且由于Flutter的开发采用Dart语言，Dart其本身结合了静态和动态语言的特性，吸取了许多其他高级编程语言的优点，对于以前学习过其他高级语言如Java、JavaScript等经验的开发者来说该语言的学习门槛并不高，只需要不到一周的时间就能快速上手。

后端项目采用了传统的SSM框架，其中的SpringBoot和Mybatis也是十分成熟的大型框架，很多产品已经上线。该框架的技术应用已经应用十分成熟，其编程语言采用了非常成熟且适合企业级项目的开发，其适合团队开发，软件工程可以做到比较规范，这是一个难以比拟的优势。Java语言在现有市场中，占有比重较大，且易上手，还拥有非常丰富的开源库。

综上所述，对项目的前后端进行技术分析，不管从使用框架还是编程语言，在其技术上都是可行的。

3.4 本章小结

本章定位系统的服务群体——商家并对其进行需求、可行性进行分析，了解的该系统具有良好的开发价值，并且其中采用的前端框架Flutter是本系统的一大亮点。

随着技术的改革更新，系统也要进行相应的改革，从技术角度来说，Flutter的加入是大势所趋，因为它良好的性能，多平台通用等优势是公司关注的热点。从经济角度来说，该系统为网购市场服务，而电商市场具有良好的前景。

# 4 系统总体设计

4.1 总体设计原则及方法

为了保证系统的可持续发展和稳定性，在系统的设计和实现过程中我们遵循如下几个原则：

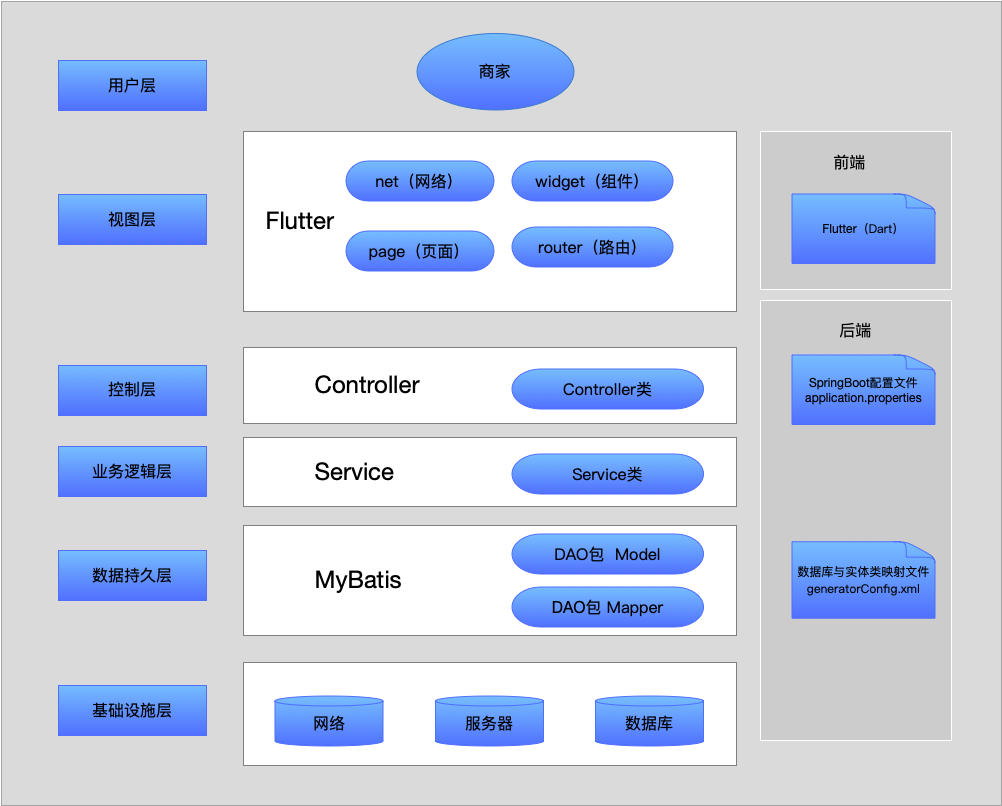
1. 整体性原则。设计和实现系统的过程中，不管是业务流程还是数据建模都需要从全局出发，从长远角度考虑。
2. 先进性原则。系统架构采用成熟、具有国内外先进性的、并符合国际发展趋势的架构，其系统的架构分为前端和后端两套架构。
3. 性能比和价格原则。在当下紧张日益激烈的产品，为了能够赢得先机就需要拥有较短的开发周期以及出色的性能和可靠性。在满足性能和速度的同时还需要遵循减少开发和维护成本。

4.2 系统架构设计

基于Flutter在线店铺管理App系统利用成熟且稳定的网络条件和数据库技术，实现的系统的自动化、远程化和数据统一化。系统采用兼有前瞻性和稳定性特点的Flutter和SpringBoot框架的开发模式和前后端分离模式，前端采用先进且有众多优势的Flutter框架去搭建项目，而后端采用成熟的SSM框架。采用Flutter + SpringBoot框架的开发模式，加速了系统的开发扩展，具有多平台、开发快、维护和升级简单灵活、成本低、数据安全等优势。[[14]](#endnote-14)

本系统采用B/S架构，遵从MVC设计思想，并对系统做详细的划分。系统采用的分层设计结构，既降低了系统各部分之间的耦合性，又使得系统易于维护和升级、具有良好的扩展性。如图4-1所示，主要包括用户层、视图层、控制层、业务逻辑层、数据持久层。分层的优点是能够让各层各司其职，遵从开闭原则，便于修改和扩展。[[15]](#endnote-15)

图4-1 系统总体架构图



下面对各层进行简要的介绍：

1. 用户层。用户层包含系统的各类用户，本系统主要服务对象只有商家。商家通过客户端与服务的进行网络通信，从而完成所需的工作。
2. 视图层。本系统为B/S架构的多平台（主要是IOS、Android、Web端））App。视图层使用的Flutter框架，以及根据用户习惯和审美设计的简洁明了的自定义布局。通过Dio（Dart的网络框架）与服务器的控制层进行交互。
3. 控制层。控制层主要通过SpringMVC进行对网络请求进行数据解析，本层接收客户端的请求，根据协定的API将请求数据注入到相应Service类中进行业务逻辑处理。
4. 业务逻辑层：业务逻辑层接收来自控制层的请求数据，并处理相应的业务，完成各个模块之间的信息流转，并通过数据持久化层对数据库进行相应操作。
5. 数据持久化层：数据持久化层可以概述为数据库的交互，使用MyBatis框架对数据库进行来自业务层的相应操作。Dao层通过MyBatis技术实现java实体类与数据库关系映射表的映射；
6. 基础设施层：该层罗列出该系统的核心设备：网络、服务器、数据库。

4.2.1 客户端设计

基于Flutter在线店铺管理App采用手机移动端（IOS、Android）和Web端多端并用的技术，用户可以通过浏览器访问系统网址或者使用移动端App进行访问。移动端App将后台业务代码加载到移动端，再通过业务代码操作本系统，发送网络请求，实现客户端和服务端进行业务交流和数据同步。

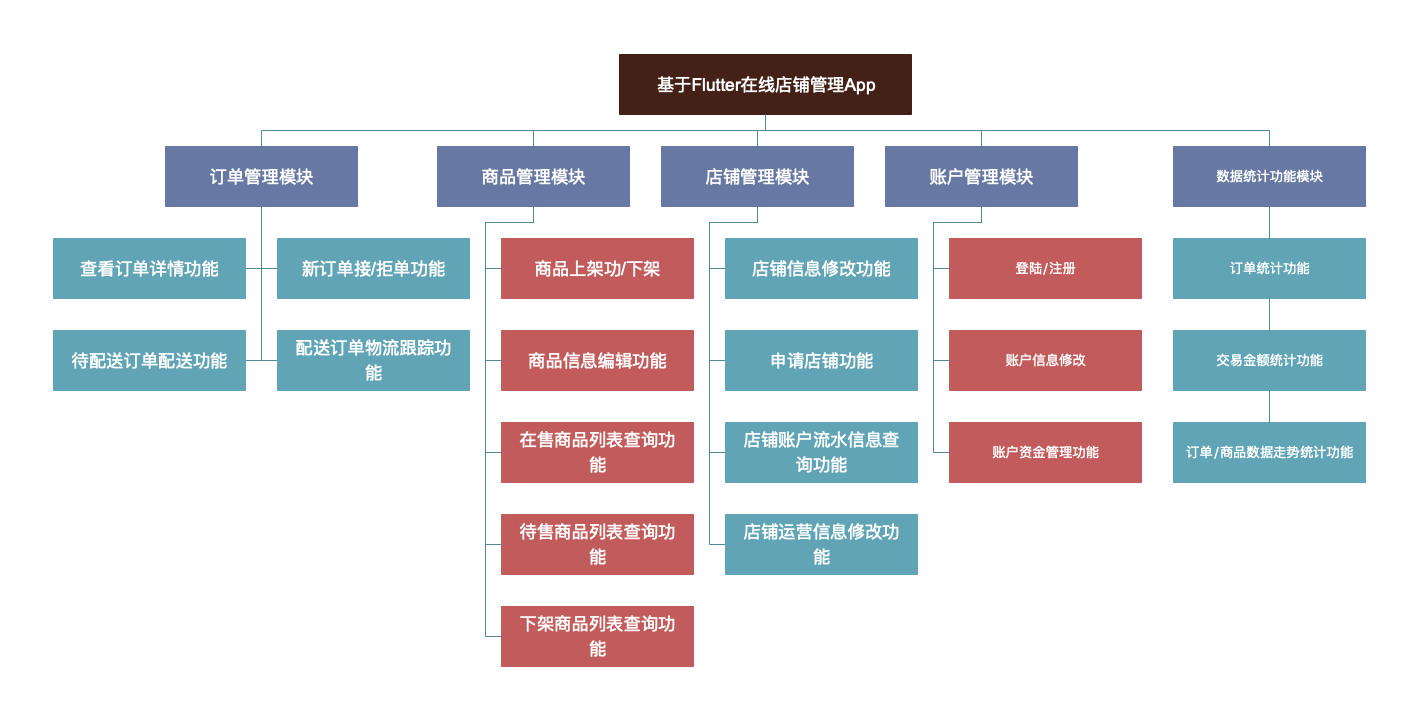
4.2.2 服务端设计

服务器部署环境为CentOS系统,应用服务器采用Apache,数据库采用开源的MariaDB数据库对各类信息进行存储和管理,因其具有跨平台 安全性高和存储容量大等特点 服务器端主要是将系统的订单、商品、账户、店铺信息进行汇集,并存储到数据库中 服务器端的程序采用Java语言进行开发,因其适用于WEB开发领域,在服务器启动后用户可以通过浏览器直接访问。

4.3 系统功能结构设计

基于Flutter在线店铺管理App按照业务种类划分，可以分成订单管理模块、商品管理模块、店铺管理模块、账户管理模块、数据统计模块五个部分。功能结构设计图如4-2所示。

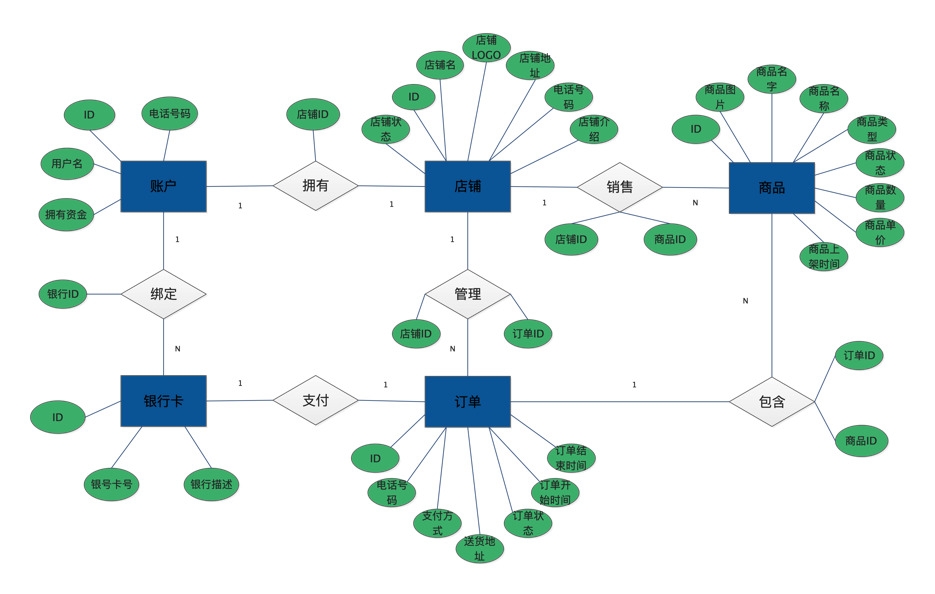
图4-2 功能结构设计图



4.4 系统数据库设计

4.4.1 数据库概念设计

根据系统的需求分析阶段的用户功能需求对其进行信息世界结构的抽象，然后将其根据概念模型的要求进行数据建模，数据库概念模型的优点是能够真实的反应现实世界中事物之间的联系以及满足用户对数据处理的要求、易于用户和开发者理解以及当应用需求和环境改变时易于对模型进行修改和扩充。下面将用E-R复合模型图来展示在线店铺管理数据库的概念设计。[[16]](#endnote-16)总体E-R图如图4-3所示。

图4-3 总体E-R图

基于Flutter在线店铺管理App的数据库采用“账户——店铺——商品——订单——银行卡”五位一体的数据模型，其中账户、店铺、商品、订单、银行卡作为数据模型的基本概念实体。店铺实体是整体数据模型的核心，围绕其展开。账户与店铺的关系是一对一的关系，一个账户绑定已经店铺对其进行管理。店铺与商品是一对多的关系，店铺拥有多种类型的商品。店铺与订单关系是一对多的关系，店铺管理多个订单。每个订单包含一个或多个商品，其支付方式是一对一的关系。

4.4.2 数据库结构设计

数据库物理设计阶段主要任务是根据微生活的具体特征，为己给定的数据库模型选择比较合适的存储结构与存取方式。以下主要对数据库表的详细设计进行说明。

1. login登陆表，用于存储账户账号（id）、电话号码（phone）、登陆密码（password）以及账户注册时间（register\_time）（注册的时候自动生成），具体结构表4-1所示。

表4-1 登陆表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| phone | varchar(20) | 非空 | 电话号码 |
| password | varchar(100) | 非空 | 密码 |
| register\_time | timestamp | 非空 | 注册时间 |

1. account账户表，用于存储登陆ID（id）账户电话（phone）、拥有店铺ID（shop\_id）、用户名（user\_name）和总资产信息（money），具体结构表4-2所示。

表4-2 账户表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| phone | varchar(20) | 非空 | 电话号码 |
| shop\_id | integer | 非空 | 店铺ID |
| user\_name | varchar(100) | 非空 | 账户名 |
| money | DECIMAL(6,2) | 非空 | 拥有金额 |

1. shop店铺表， 存储店铺ID（id）、店铺名称（shop\_name）、店铺图片（shop\_logo）、店铺所在地址（address）、店铺联系电话（phone）、店铺运营状态（status\_id）和店铺介绍信息（introduction），具体结构表4-3所示。

表4-3 店铺表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| shop\_name | varchar(50) | 非空 | 店铺名 |
| shop\_logo | varchar(200) |  | 店铺LOGO |
| address | varchar(200) | 非空 | 店铺地址 |
| phone | varchar(20) | 非空 | 店铺联系电话 |
| status\_id | integer | 非空 | 店铺运行状态ID |
| introduction | varchar(200) |  | 店铺介绍 |

1. shop\_status 店铺状态表，存储店铺运营的状态ID（id）和该ID对应的名称（name）与店铺表中的店铺状态相关联，具体结构表4-4所示。

表4-4 店铺状态表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| name | varchar(20) | 非空 | 状态命名 |

1. orders订单表，存储订单的销售店铺ID、下单客户的联系电话、客户的支付类型、客户地址、订单状态ID、订单生成时间（插入数据的时候自动生成）和订单结束时间，具体结构表4-5所示。

表4-5 订单表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| shop\_id | integer | 非空 | 店铺ID |
| phone | varchar(20) | 非空 | 顾客联系电话 |
| pay\_type | integer | 非空 | 支付类型 |
| address | varchar(200) | 非空 | 送货地址 |
| status\_id | integer | 非空 | 订单状态ID |
| start\_time | timestamp | 非空 | 订单开始时间 |
| end\_time | timestamp |  | 订单结束时间 |

1. order\_status订单状态表，存储订单的状态ID和该ID对应的名称与订单表中的订单状态相关联，具体结构表4-6所示。

表4-6 订单状态表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| name | varchar(20) | 非空 | 状态命名 |

1. goods商品表，存储商品的显示图片、商品名称、商品所属商店、商品类型、商品状态、商品数量、商品单价、商品上架时间（添加商品时自动生成），具体结构表4-7所示。

表4-7 商品表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| icon | varchar(200) |  | 商品图片 |
| title | varchar(200) | 非空 | 商品名字 |
| shop\_id | integer | 非空 | 店铺ID |
| type | integer | 非空 | 商品类型ID |
| status | integer | 非空 | 商品状态ID |
| count | integer | 非空 | 商品数量 |
| price | DECIMAL(6,2) |  | 商品单价 |
| start\_time | timestamp | 非空 | 商品上架时间 |

1. good\_status商品状态表，存储商品运营的状态ID和该ID对应的名称与商品表中的商品状态相关联，具体结构表4-8所示。

表4-8 商品状态表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| name | varchar(20) | 非空 | 状态命名 |

1. good\_type商品类型表，存储商品类型ID和该ID对应的名称与商品表中的商品类型相关联，具体结构表4-9所示。

表4-9 商品类型表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| name | varchar(20) | 非空 | 类型命名 |

1. order\_good订单商品表，存储订单和商品关系、购买商品数量、订单总价，具体结构表4-10所示。

表4-10 订单商品表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| order\_id | integer | 非空 | 订单ID |
| good\_id | integer | 非空 | 商品ID |
| count | integer | 非空 | 购买商品数量 |
| price | DECIMAL(6,2) |  | 订单总价 |

1. bank银行表，存储能够绑定的银行名与银行描述。具体结构表4-11所示。

表4-11 银行表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| bank\_name | varchar(50) | 非空 | 银行名称 |
| first\_letter | varchar(100) |  | 银行描述 |

1. account\_bank账户银行表，存储账户和银行关系，具体结构表4-12所示。

表4-12 账户银行表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| bank\_id | integer | 非空 | 银行ID |
| account\_id | integer | 非空 | 账户ID |

1. account\_running账户流水表，存储银行与账户关系、账户流水信息和账户该笔流水生成时间（账户金额变动触发自动生成），具体结构表4-13所示。

表4-13 账户流水表

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 类型 | 约束 | 注释 |
| id | integer | 非空 | ID号 |
| bank\_id | integer | 非空 | 银行ID |
| account\_id | integer | 非空 | 账户ID |
| withdrawal | DECIMAL(6,2) | 非空 | 提现/充值金额 |
| w\_time | timestamp | 非空 | 提现时间 |

4.5 本章小结

通过本章的描述，从总体、功能、数据三个方面进行设计说明。阐述了本系统的总体设计原则与方法，对系统整体架构作出设计，对需求提出的功能进行相应的模块划分与开发设计，对数据层进行相应的设计，为系统开发提供了总纲。

# 5 系统实现

5.1 系统开发及运行环境

5.1.1 开发环境

1. 硬件环境

表5-1 开发硬件环境表

|  |  |
| --- | --- |
| 硬件环境 | 名称 |
| 操作系统 | MacOS Mojava 10.14.6 |
| 处理器 | 1.8 GHz Intel Core i5 |
| 内存 | 24 GB 2400 MHz DDR4 |

1. 软件环境

表5-2 开发软件环境表

|  |  |
| --- | --- |
| 软件环境 | 名称 |
| 开发语言 | Java、Dart |
| 前端开发框架 | Flutter |
| 服务的开发框架 | SpringBoot |
| 代码编辑器 | Visual Studio Code、[IntelliJ IDEA](https://www.jetbrains.com/idea/) |
| 调试工具 | Debug |
| 版本管理 | Git、Github |
| 依赖管理 | Maven、Flutter Pub |
| 数据库 | MariaDB |

5.1.2 运行环境

表5-3 运行环境表

|  |  |
| --- | --- |
| 运行环境 | 名称 |
| 前端运行环境 | IOS、Android、浏览器 |
| 后端运行环境 | CentOS、MacOS |

5.2 系统功能模块实现

5.2.1 订单管理模块

用户登陆后进入系统的首页，首页默认显示的是订单管理页面。该页面包含订单列表显示导航和订单搜索按钮。

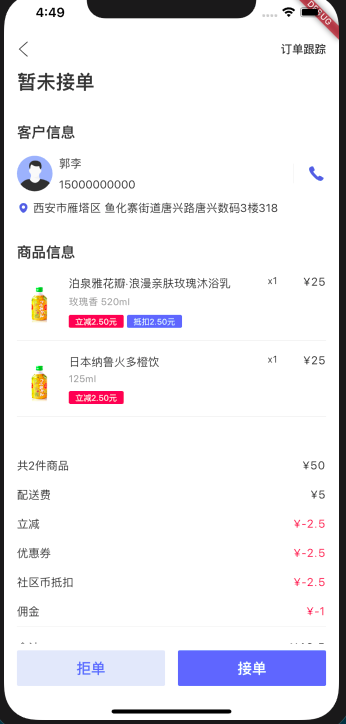
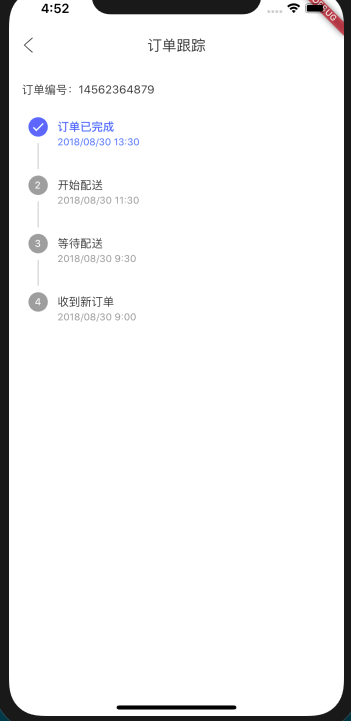
用户可以通过点击订单控制导航的订单不同状态（新订单、待配送、待完成、已完成、已取消）来跳转不同状态订单页面。用户点击搜索按钮，页面会跳转到搜索页面。订单管理模块页面如图5-1所示。

图5-1 订单管理页面



每一件订单卡片，点击就会跳转到订单信息管理页面，该页面展示了订单的详细信息，如商品数量、优惠、下单顾客头像等信息。顾客联系电话能够点击触发拨打电话、物流跟踪可以跳转到订单物流进度页面。订单信息管理页面如图5-2所示。

图5-2 订单信息管理页面

、 

5.2.2 商品管理模块

点击首页的模块控制导航的商品管理，页面会跳转到商品管理页面。该页面主要由商品列表页面、搜索商品、添加商品、商品类型筛选、商品功能部分组成。点击商品列表的不同状态商品导航，页面会跳转。商品管理页面如5-3所示。

图5-3 商品管理页面

触发添加商品按钮，跳转到添加商品页面，该页面需要添加商品的信息包括商品名称、商品图片、商品价格等信息。触发商品编辑按钮，显示商品编辑页面，可以对商品的一下基本信息进行修改更新。商品添加/编辑页面如图5-4所示。

图5-4 商品添加/编辑页面



5.2.3 店铺管理模块

进入店铺管理模块，该模块可以查看店铺账号、店铺名称、店铺头像信息、账户管理模块、店铺设置功能。店铺管理页面如图5-5所示。

图5-5 店铺管理页面



通过触发店铺设置页面，可以对店铺营业状况进行控制、查看店铺基本信息并修改低店铺信息。店铺设置页面如图5-6所示。

图5-5 店铺设置页面

5.2.4 账户管理模块

通过店铺管理模块可以进入账户管理模块，该模块有账户流水、资金管理、提现账户三个功能。账户管理模块如图5-6所示。

图5-6 账户管理页面



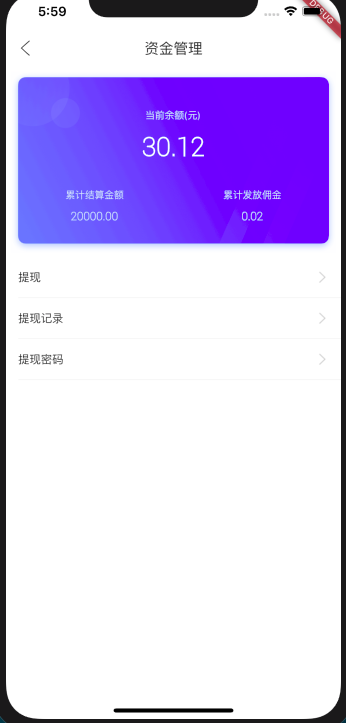
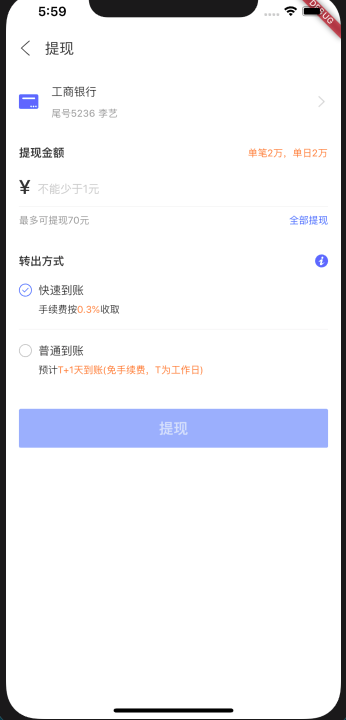
账户流水页面显示了店铺每笔账单的时间、金额、描述等基本信息。账户流水页面如图5-7所示。

图5-7 账户流水页面



资金管理页面有显示账户余额、提现、提现记录、提现密码功能。提现页面能输入提现金额，选择提现方式和选择提现的银行或微信。提现记录页面显示每笔提现金额的详细信息。提现密码能够修改密码和忘记密码功能。资金管理页面如图5-8所示。

图5-8 资金管理页面

提现账户页面显示账户绑定的银行卡和添加绑定银行卡。添加绑定银行需要选择账户类型，然后根据账户类型显示填写信息。提现账户页面如图5-9所示。

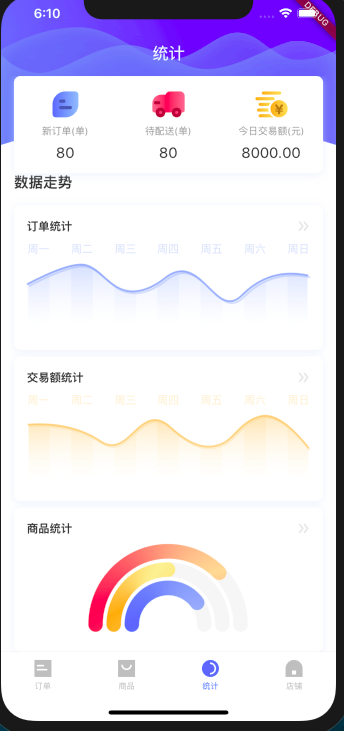
图5-9 提现账户页面

5.2.5 数据统计模块

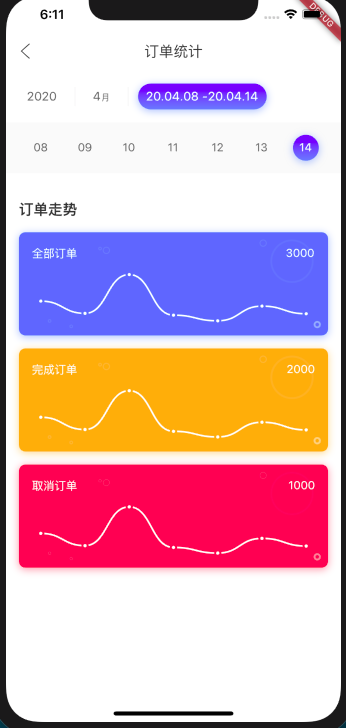
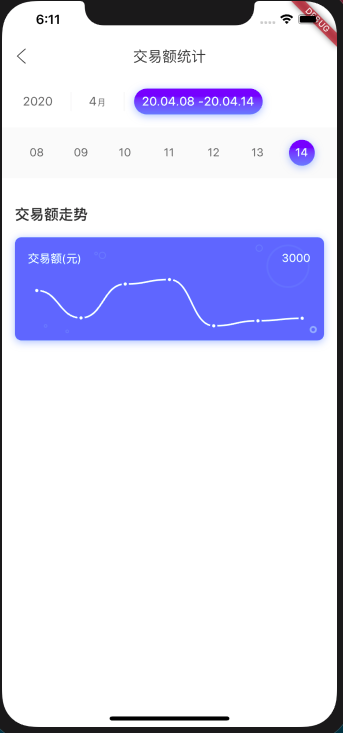
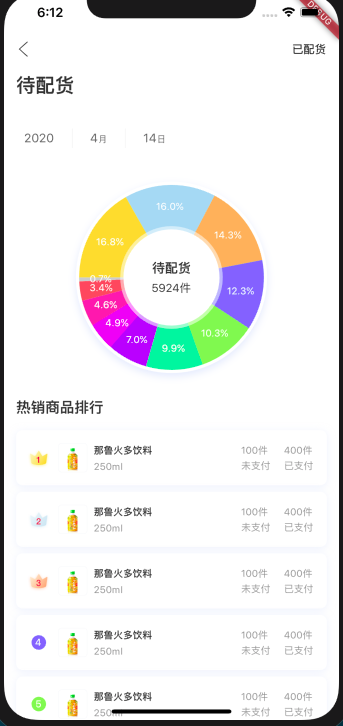
进入数据统计模块，会显示今日销售数据和数据走势两个部分。数据走势包括订单统计、交易额统计和商品统计。数据统计页面如图5-10所示。

图5-10 数据统计页面



订单统计页面能够选择日期，去查看当日的全部订单、完成订单、取消订单的走势图。交易金额统计显示选择日期的交易金额走势图。商品统计页面会以扇形图以及数据形势显示当日商品数据。数据走势页面如图5-11所示。

图5-11 数据走势页面

5.3 本章小结

本章节基于第四章的系统设计，实现了基于Flutter在线店铺管理App，首先介绍了该系统的开发和运行环境，然后按照功能模块展示了系统的页面实现效果和其具体的用法。

# 6 系统测试

基于Flutter在线店铺管理系统在测试方法上以系统设计的功能设计的功能设计部分为基础，在系统进行测试的时候通过对系统业务流程的设计测试用例。首先能流程为基础，在进行测试时，是通过对流程的设计测试用例来进行的。首先对每个模块的总体功能测试，是通过输入输出的结果进行对比，然后根据模块业务流程的设计，对于一些能够满足全部条件分支的测试用例进行设计和规划。由此能够对系统的全部功能进行总体体测试，将错误和不足找出来并进行纠正改进。[[17]](#endnote-17)

6.1 测试方案

6.1.1 功能性测试

基于Flutter在线店铺管理系统的每一个独立的模块都是系统业务的参与者、实施者，并且模块在参与的过程中，其业务在系统内部完成资源流转、资源共享，保障了系统的安全性、流畅性。在系统App的层面来看，要推动本系统的线上运营，功能性测试是必不可少的。在功能性测试的同时，测试人员需要测试系统的准确性、有效性、实用性，同时还需要对存在于本系统中的缺陷错误加以修复。因此，测试人员必须要对系统的五个独立的大模块分别展开功能性测试。

6.1.2 兼容性测试

兼容性即要确保系统可以跟其它平台保持有序的、无障碍的交互沟通。因此系统操作需要能够在不同类别移动端和浏览器上进行测试，从而防止发生系统不兼容，无法正常运行状况的产生。本系统将采用MacOS自带的IOS模拟器Simulator测试系统在各个移动设备上的兼容性。测试的机型包括iPhone5、iPhone5、iPhone6/7/8、iPhone 11 Pro。

6.1.3 用户友好性测试

用户友好性是系统操作界面布局需要将字体大小、格局布置、颜色配比等要素设计得简洁美观，由此用户使用系统的时候才会有更加舒适的体验；除此之外，系统还需要具友好性有可视化操作、合理的快捷键设计、增加适当的文字照片作为辅助说明，以此让用户对系统的功能能够更加简单的理解，让用户流畅容易的完成系统操作。

6.1.4 安全性测试

安全性测试(Security Testing)是指有关验证应用程序的安全等级和识别潜在安全性缺陷的过程，其主要目的是查找软件自身程序设计中存在的安全隐患，并检查应用程序对非法侵入的防范能力，安全指标不同，测试策略也不同。

但安全是相对的，安全性测试并不能最终证明应用程序是安全的，而只能验证所设立策略的有效性，这些对策是基于威胁分析阶段所做的假设而选择的。例如，测试应用软件在防止非授权的内部或外部用户的访问或故意破坏等情况时的运作。

6.2 测试用例分析

测试用例的主要对象是系统的功能测试。根据对功能模块的测试来确认功能的信息交换的准确性和有效性。下面将会对系统功能模块进行测试。

6.2.1 订单管理模块

订单管理模块的测试用例如表6-1所示

图6-1 订单管理模块测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预期结果 | 是否通过 |
| 订单导航 | 已登陆并绑定店铺 | 1. 点击待配送导航 2. 点击待完成导航 3. 点击已完成导航 4. 点击已取消导航 | 1. 订单列表刷新成待配送列表 2. 订单列表刷新成未完成列表 3. 订单列表刷新成已完成送列表 4. 订单列表刷新成已取消列表 | 是 |
| 订单卡片 | 已登陆并绑定店铺、至少存在一个订单 | 1. 点击卡片除去按钮外的部分 2. 点击联系电话 | 1. 页面跳转到订单详细信息页面 2. 提示拨打联系人，确认后联系顾客 | 是 |
| 搜索组件 | 已登陆并绑定店铺 | 1. 点击搜索组件，2. 输入搜索条件，并确认 | 1. 页面跳转到搜索页面  2. 搜索页面显示搜索列表 | 是 |

6.2.2 商品管理模块

商品管理模块的测试用例如表6-2所示

图6-2 商品管理模块测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预期结果 | 是否通过 |
| 商品导航 | 已登陆并绑定店铺 | 1. 点击待售导航 2. 点击下架导航 | 1. 商品列表刷新成待售列表 2. 商品列表刷新成下架列表 | 是 |
| 商品类型选择 | 已登陆并绑定店铺 | 1. 点击商品类型下拉栏 2. 选择商品类型 | 1. 显示所有商品类型 2. 商品列表根据商品类型刷新 | 是 |
| 商品卡片功能组件 | 已登陆并绑定店铺 | 1. 点击卡片功能组件 2. 点击编辑功能 | 1. 显示商品功能 2. 页面跳转到商品编辑页面 | 是 |
| 搜索组件 | 已登陆并绑定店铺 | 1. 点击搜索组件，2. 输入搜索条件，并确认 | 1. 页面跳转到搜索页面  2. 搜索页面显示搜索列表 | 是 |
| 添加商品组件 | 已登陆并绑定店铺 | 1. 点击添加组件 2. 编辑商品信息，并确认提交 | 1. 页面跳转商品添加页面 2. 商品添加成功 | 是 |

6.2.3 店铺管理模块

店铺管理模块的测试用例如表6-3所示

图6-3 店铺管理模块测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预期结果 | 是否通过 |
| 店铺设置 | 已登陆并绑定店铺 | 1. 点击店铺设置组件 2. 修改店铺信息，然后提交 | 1. 页面跳转到店铺设置页面 2. 店铺信息修改成功 | 是 |

6.2.4 账户管理模块

账户管理模块的测试用例如表6-4所示

图6-4 账户管理模块测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预期结果 | 是否通过 |
| 账户流水 | 已登陆并绑定店铺 | 点击账户流水 | 页面跳转账户流水页面 | 是 |
| 资金管理 | 已登陆并绑定店铺 | 1. 点击资金管理 2. 点击提现 3. 填入提现信息，并确认提现 4. 点击提现密码 5. 点击忘记密码 6. 修改密码，并确认 7. 点击提现记录 | 1. 页面跳转资金管理页面 2. 页面跳转提现页面 3. 提现成功 4. 页面跳转提现密码页面 5. 页面跳转忘记密码页面 6. 修改密码成功 7. 页面跳转提现记录页面 | 是 |
| 提现账号 | 已登陆并绑定店铺 | 1. 点击提现账号 2. 点击添加组件 3. 选择提现账号类型 4. 填写提现账号信息，并确认 | 1. 页面跳转提现账号页面 2. 页面跳转添加提现账号页面 3. 页面根据提现帐号类型刷新需要填写的信息 4. 成功添加提现帐号 | 是 |

6.2.5 数据统计模块

数据统计模块的测试用例如表6-5所示

图6-5 数据统计模块测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 预置条件 | 测试步骤 | 预期结果 | 是否通过 |
| 订单统计 | 已登陆并绑定店铺 | 1. 点击订单统计 2. 选择日期 | 1. 页面跳转订单统计页面 2. 订单统计页面根据所选日期刷新订单走势图 | 是 |
| 交易额统计 | 已登陆并绑定店铺 | 点击交易额统计 | 页面跳转交易额统计页面 | 是 |
| 商品统计 | 已登陆并绑定店铺 | 点击商品统计 | 页面跳转商品统计页面 | 是 |

6.3 本章小结

通过对本章的描述，对系统进行多层次，多方法的测试。通过对订单管理商品管理、店铺管理等模块进行测试用例分析，同时将测试结果以展示表格形式展示，对整个系统运行效果进行测试。

# 7 总结与展望

7.1 全文工作总结

本文使用前后端分离的模式对基于Flutter在线店铺App进行设计与实现，其目的是为了解决随着网络购物产业发展和互联网技术不断更新产生的性能跟不上，用户需求无法满足、开发维护成本高的多平台多套代码的软件产品问题。本系统既能够让网络购物中的商家群群体更加方便简洁的操作网上业务，又能够为企业开发和维护系统降低成本，因此采用了最新的前端Flutter技术，这是展露新头角的一种尝试。项目开展前期从技术、热门网购平台等多方面进行调研，作出系统需求分析和可行性分析。然后根据调研结果整理的需求分析，结合开发技术确定详细的系统设计。最后完成系统总体开发后，考虑系统的完善性、可维护性、用户友好性和安全性等方面，对系统进行优化。

具体工作如下：

1. 系统设计方面，前期通过对当前热门技术和热门平台所使用的技术框架进行调研和分析。然后根据调研结果来进行技术框架选型，最后确定系统的最终技术体系。在研究学习项目需要的技术体系后，搭建系统基础架构。
2. 页面布局设计方面，前期调研市场移动软件和Web端系统的排版和设计，根据调研结果分析用户对颜色、主题、布局等方面的喜好，最后确定本系统的 UI 设计图和用户交互图。设计的UI 设计图和用户交互设计图结合前端Flutter框架的组件库，最后完成系统的页面布局定制。
3. 业务功能方面，参考了热门电商平台和外卖平台等网络购物平台，将电商功能和外卖功能结合商家的实际需求确定本系统的所需的全部功能并完成需求分析和系统功能设计。最后通过详细的系统设计，完成系统的基本功能开发。
4. 系统完成后，参考其它热门网购应用的数据，编纂模拟数据，并将其导入数据库中，测试系统整体功能和性能。对于测试过程中的 bug，进行修复和完善，最终实现整个系统。

7.2 对未来的展望

对于基于Flutter的在线店铺管理App系统来说，是以实际商家管理店铺的业务流程为基础，对商品和订单管理效率进行提升。该系统可以结合缺少商家管理系统的购物系统进行组合，通过数据库展开信息的共享与交互，从而使沟通和开发成本降低，让信息交换与反馈都能快速进行，从而促进了业务工作的开展。与此同时，基于Flutter在线店铺管理系统还有很多不足和弊端，比如，在部分模块功能上还是非常的简单；系统的非功能性能还有很多可以加强的部分；其数据库的设计和建立简单以及数据量还是很小等。然而对于互联网技术而言，正处于互联网技术高速发展的时代，有着开发语言和技术框架逐渐丰富和完善、界面布局设计的发展趋于美观、网络的功能和性能逐渐得到优化等优势，处于这样一个最好的时代让系统在发展的过程中，能够趋于多样性和丰富性，从而在未来的发展上对于店铺管理系统将会更加的完善，也会创造更高的价值。

# 参考文献

1. 王娟.新零售背景下生鲜电商O2O模式分析——以小象生鲜为例[J].中国储运,2019,(12):106-109.DOI:10.16301/j.cnki.cn12-1204/f.2019.12.041 [↑](#endnote-ref-1)
2. 2020年中国网络购物市场分析报告-产业现状与未来规划分析 [↑](#endnote-ref-2)
3. 基于Vue.js框架的掌上零售系统的设计与实现 （硕士论文） [华中科技大学](http://navi.cnki.net/KNavi/PPaperDetail?pcode=CMFD&logo=GHZKU) .2019 [↑](#endnote-ref-3)
4. 何海波.国内外网络零售市场发展状况对比分析[J].商业经济研究,2017,(07):133-135. [↑](#endnote-ref-4)
5. 基于Flutter的跨平台移动APP开发前景研究（期刊）[信息与电脑(理论版)](http://navi.cnki.net/KNavi/JournalDetail?pcode=CJFD&pykm=XXDL).2019(15) [↑](#endnote-ref-5)
6. 基于Flutter的配网监控APP研究与设计（期刊）[电子世界](http://navi.cnki.net/KNavi/JournalDetail?pcode=CJFD&pykm=ELEW).2019(20) [↑](#endnote-ref-6)
7. 基于springmvc与ibatis的轻量级web应用研究 [刘军](http://www.oalib.com/search?kw=%E5%88%98%E5%86%9B&searchField=authors),[戴金山](http://www.oalib.com/search?kw=%E6%88%B4%E9%87%91%E5%B1%B1&searchField=authors) [↑](#endnote-ref-7)
8. [The design and implementation of the framework for Spring+SpringMVC+MyBatis in the development of Web application](https://x.cnki.net/read/article/xmlonline?filename=JKDZ201912012071&tablename=IPFDTOTAL&dbcode=IPFD&topic=&fileSourceType=1&taskId=&from=&groupid=&appId=CRSP_BASIC_PSMC&rdts=1586846011) [↑](#endnote-ref-8)
9. MyBatis在《Web中间件技术》课程中的应用（期刊）[教育教学论坛](http://navi.cnki.net/KNavi/JournalDetail?pcode=CJFD&pykm=JYJU).2019(38) [↑](#endnote-ref-9)
10. 基于eclipse的毕业设计流程系统的设计与开发（期刊）[智库时代](http://navi.cnki.net/KNavi/JournalDetail?pcode=CJFD&pykm=ZKSD).2020(13) [↑](#endnote-ref-10)
11. 校园网中私有云盘的实现与应用(期刊)广东职业技术教育与研究.2019(06) [↑](#endnote-ref-11)
12. 基于J2EE和MVC模式的Web应用研究（期刊）[软件](http://navi.cnki.net/KNavi/JournalDetail?pcode=CJFD&pykm=RJZZ).2020(03) [↑](#endnote-ref-12)
13. 姚媛, 许天才. 移动图书馆用户体验评价结构模型研究. 国家图书馆学刊, 2018, 27(05):32-43 [↑](#endnote-ref-13)
14. 基于SpringBoot和Vue框架的教代会提案系统的设计与实现（期刊）[科技创新与应用](http://navi.cnki.net/KNavi/JournalDetail?pcode=CJFD&pykm=CXYY).2020(03) [↑](#endnote-ref-14)
15. [唐菁敏，高张宝.基于B/S架构的企业综合管理系统的研究与设计[J].信息技术，2019(3):144-148.](https://x.cnki.net/read/article/xmlonline?filename=CXYY202003037&tablename=CJFDTOTAL&dbcode=CJFD&topic=&fileSourceType=1&taskId=&from=&groupid=&appId=CRSP_BASIC_PSMC&rdts=1585486132) [↑](#endnote-ref-15)
16. 梁少岗.基于ArcGIS的不动产登记数据库系统设计与实现[D].西安科技大学,2019. [↑](#endnote-ref-16)
17. 基于B/S的烟草销售管理系统设计与实现 （硕士论文） [华东交通大学](http://navi.cnki.net/KNavi/PPaperDetail?pcode=CMFD&logo=GHDJU) .2020

    致 谢

    历时多个月，从论文开题、资料收集、系统设计、系统实现以及论文的反复修改，到现在的本论文已经基本完成。

    首先需要感谢的是我的我的指导老师何琛老师。在毕业设计的选题和实现、论文的编纂过程中，何琛老师给我提出了许多宝贵的意见。导师在论文开题报告阶段与我交流我的选题构思和系统定位，之后的论文编纂以及系统实现阶段，反复督促和严格要求我们提交自己的完成进度，并提供一些宝贵的建议。我从何琛老师身上不仅学到了很多专业知识，还有他对学生的认真负责的态度。非常感谢导师在毕业设计给我的极大帮助，使得我对系统设计的总体思路以及论文格式的要求有了很大的把握。

    然后需要感谢大学期间教授我们专业知识的老师们，通过大学四年的专业知识的学习，让我奠定了实现毕业设计的基础，更加重要的是老师们教授的学习方法和经验，这将在我以后的职业以及做人处事上有着很大的引导下帮助。

    最后感谢我的父母以及学校的朋友们，父母提供了我学习的机会，朋友们陪伴我成长。非常感谢父母朋友在学习上对我的鼓励和支持。 [↑](#endnote-ref-17)