

玛尔沃超市管理系统

**可行性分析报告**

**版本：2.0**

编写： 阳玉洁

校对： 王娜

审核： 方亚群

**西北工业大学－玛尔沃超市管理系统服务小队**

**2020年6月28日**

**目 录**

[1 文档简介 3](#_Toc296)

[1.1文档名称 3](#_Toc30074)

[1.2 项目概述 3](#_Toc10381)

[1.3 文档概述 3](#_Toc13866)

[1.4 参考文档 4](#_Toc21508)

[2 可行性分析的前提 4](#_Toc13523)

[2.1 项目背景及内容 4](#_Toc30719)

[2.2 项目要求 5](#_Toc32676)

[2.3 项目目标 6](#_Toc12137)

[2.4 项目的环境、条件、假定 6](#_Toc23914)

[2.5 系统功能 7](#_Toc10026)

[2.6 进行可行分析的方法 12](#_Toc27538)

[3 可选方案 13](#_Toc8850)

[3.1 整体技术方案 13](#_Toc30047)

[3.2 功能实现方案 13](#_Toc18496)

[3.3 选择最终方案的准则 40](#_Toc5553)

[4 所建议的系统 40](#_Toc9040)

[4.1 对所建议的系统的说明 40](#_Toc9)

[4.2 数据流程和处理流程 41](#_Toc25291)

[5 可行性分析 43](#_Toc29363)

[5.1 经济可行性 43](#_Toc29093)

[5.2 技术可行性 45](#_Toc9020)

[5.3 用户可行性 45](#_Toc5091)

[5.4 法律可行性 46](#_Toc16285)

# 

# 1 文档简介

## 1.1文档名称

中文名称：《可行性分析报告》

英文名称：“Feasibility Analysis Report”

英文缩写：“FAR”

文档版本：“2.0”

文档编码：“Marvel-XB(G)-FAR-2.0”

## 1.2 项目概述

本文档适用于“玛尔沃超市管理系统开发项目”的开发过程。该项目由西北工业大学软件学院17级学生提出并自主设计实现。项目标识号为“Marvel”，软件产品版本号为“2.0”。

项目内容为：玛尔沃超市管理系统面向三类用户：超市经理、售货员以及超市顾客。

目标在于实现超市商品的智能化、信息化和简单化；实现售货员对商品信息的添加、删除、编辑、查询和对商品以及顾客购买信息的管理；提高超市经理以及售货员工作信息提交和反馈的效率，减轻管理人员的工作负担，提高管理效率；同时方便顾客线上查看商品信息及购物记录等等。

## 1.3 文档概述

本文档依据《国标GB/T 8567-2006计算机软件文档编制规范》制定，详细描述了玛尔沃超市管理系统的目标和具体要求、数据流程和各用户角色业务处理流程，同时列出可以选择的技术方案及其不同方案的优缺点对比，然后对系统的经济、技术、法律和用户使用可行性进行分析，以供后续管理和开发参考。

## 1.4 参考文档

* 《GB/T 8567-2006计算机软件文档编制规范》，国家标准
* 《Marvel-XB(G)-SDS-2.0 软件文档规范》，项目成员编写

# 2 可行性分析的前提

## 2.1 项目背景及内容

随着我国经济实力不断增强，超市行业也得到了蓬勃发展。此时，超市的竞争也进入到了一个全新的领域，竞争已不再是规模的竞争，而是技术的竞争、管理的竞争和人才的竞争。然而我国大部分地区仍然是中小型城市、城镇和农村，这些地区的超市也是以中小型超市为主（即使是大型城市仍分布着许多中小型超市）。长期以来，这些超市一直使用着最简单的人工管理，超市经营商品的销售、管理及核算的手工信息管理使得工作量巨大、服务质量差、工作效率低。同时对市场反馈的信息不能及时处理，从而导致销售与市场脱节，最终的结果只能使超市在经营中立于不利的地位。在此背景下，我们项目小组开发了玛尔沃超市管理系统，该系统软件容易被接受，具有简单易学性，便于管理等功能，是对超市管理的一种有效工具。

本项目旨在开发一个科学高效的超市管理系统，实现超市运作的规范化、信息化和系统化管理，降低管理成本。玛尔沃超市管理系统具有强大的系统功能，包括商品管理、顾客管理、售货员管理、消费购物记录、系统公告和财务管理等多种功能。在方便顾客查看购物记录的同时，实现货物商品信息管理的智能化、信息化和简单化；提高售货员的工作效率，降低管理难度；建立起规范化、智能化的超市管理新模式。

## 2.2 项目要求

### 2.2.1基本要求

本超市管理项目基于B/S架构，通过Web网站为用户提供服务。本项目的最终用户为超市经理、售货员和顾客，他们是经常性用户；系统维护人员是计算机专业人员，熟悉操作系统和数据库，是间隔性用户。

1. 经理：经理能够管理（添加、删除、编辑、查询）售货员的账号信息且对新增售货员无数量限制，可以发布、编辑和删除系统公告，同时可以查看超市营业收入信息，实现对超市整体运行情况的统一有效管理；经理不能注册顾客账户，不能编辑或删除顾客信息以及查看其购买记录。
2. 售货员：售货员可以注册顾客账户且无数量限制，能够管理（添加、编辑、删除、查询）商品信息，同时对商品的类别及位置进行管理，也有权限编辑和删除顾客信息，并查看顾客的购买记录和超市的消费记录。
3. 顾客：顾客注册后能够对自己的信息进行维护和管理，同时能够查看购物记录；未注册顾客不能查看购物记录。

### 2.2.2系统要求

性能：可以方便快捷有效地完成查询、注册等各项操作，录入数据合法性校验程序高，查询速度快（不超过5秒）。保证信息的正确和及时更新，并降低信息访问的成本，技术可靠性高。

可扩展性：能够适应应用要求的变化和修改，具有灵活的可扩展性。

数据安全和保密：具有较高的安全性。系统对不同的用户提供不同的功能模块，只有注册了的顾客才能查看个人的购物记录；只有具有一定权限的售货员用户才能管理顾客信息，只有具有高级权限的部门管理者或维护人员用户才能对用户进行管理；还应具有一定的保护机制，防止系统被恶意攻击，信息被恶意修改和窃取；有完善的备份机制，如果系统被破坏应该能快速恢复。

## 2.3 项目目标

本超市管理项目主要提供超市商品管理和用户基本信息的维护以及查看消费记录等功能。本项目旨在提高超市运营工作的效率，减少相关人员的工作量，使资源和人力真正做到科学、合理地规划以及高效地分配。

1. 开发的界面美观大方、用户友好、可扩展性强。
2. 实现商品信息更新过程，及其完整数据跟踪信息。
3. 提供灵活、合理的权限划分，使整个系统的管理过程清晰。
4. 项目的文档完备，包含管理、技术、产品类文档。
5. 项目需提供供需双方都满意的结果，降低超市管理维护费用，提高工作效率，改善服务质量，为超市运营发展决策提供支持，该项目的设计目标尽量达到人力和设备的节省，并且处理数据速度提高。

## 2.4 项目的环境、条件、假定

1. 软件环境

* 操作系统软件Microsoft Windows 10，64位家庭版。
* 版本控制软件：Git。
* 集成开发环境：IntelliJ IDEA。
* 数据库：PostgreSQL。
* 浏览器：Firefox、IE或者Google Chrome。

1. 硬件环境：2G Hz 双核 CPU、4G 内存、2G 自由硬盘空间、Windows10系统。
2. 条件

* 拥有能快速响应超市管理业务的服务器。
* 开发团队有4名以上精通Java开发技术的程序员。
* 开发的整个过程必须使用版本控制软件并保证每天的进度。

1. 假定

* 项目所需要的资金和资源充足。
* 项目团队包括熟悉超市管理业务逻辑的开发人员和当前参与者。

## 2.5 系统功能

### 2.5.1功能1 - 登录与登出

经理、售货员和已经注册的顾客在进入本系统主界面后，可以选择“LogIn”按钮进入登陆界面，在输入正确的用户名和密码后即可进入相应的个人操作界面；若输入为用户名和密码不匹配、用户名/密码不符合规范、用户名/密码为空等非正确输入状态下的情况，则会提示相应的错误信息，登录失败。

已经登录的用户若要退出系统，可点击“LogOut”按钮登出系统。

### 2.5.2（经理）功能2 - 添加售货员账户

经理登录进入经理页面后，可以进行添加售货员账户的操作。通过输入要添加的售货员的账户名、售货员姓名、性别、电话和邮箱等基本信息，在确认无误后即可提交注册信息，系统接收到信息后反馈注册是否成功。若注册成功，经理可以在售货员信息列表中看见新添加的售货员信息；若不成功，则需要根据系统反馈的不成功原因重新注册。

### 2.5.3（经理）功能3 - 搜索、编辑和删除售货员信息

经理可以在售货员管理模块进行搜索、编辑和删除售货员信息的操作。

经理在搜索栏输入售货员账户名或售货员姓名即可查询到具体的售货员信息；在售货员信息的操作选项中有编辑和删除两个功能，经理可以编辑售货员除账户名外的其他信息或删除售货员信息，系统给与确认提示和操作反馈。

### 2.5.4（经理）功能4 - 修改经理账户密码

在经理修改密码模块中，输入原密码、新密码和确认新密码后可以修改经理登录密码，系统验证原密码无误且两次密码输入一致后反馈操作成功信息，否则反馈密码修改失败提示信息。

### 2.5.5（经理）功能5 - 发布、编辑和删除系统公告

玛尔沃超市管理系统具有信息发布功能，包括超市介绍、商品促销和通知等，目的是让系统用户及时获得超市的相关消息。

经理在系统公告管理模块可以发布、编辑和删除系统公告。经理编辑要发送的公告的标题和内容后提交给系统审核，若审核通过，系统反馈成功发布信息，同时公告会展示在系统主界面中；否则，系统反馈发布公告失败提示信息。

### 2.5.6（经理）功能6 - 查看收入信息（每日/月/年）

经理在系统的财务管理模块中可以通过选择不同的区间范围查看玛尔沃超市每日/月/年的财务收入状况。财务收入状况同时提供折线图和表格两种可视化展示效果。

### 2.5.7（售货员）功能7 - 修改售货员密码

售货员可在其个人信息管理模块选择修改登录密码，输入要修改的售货员账户的邮箱地址，并在邮箱中获取系统发送的验证码；之后输入验证码和两次新密码后，提交系统，若修改成功，系统反馈修改成功信息，否则系统反馈密码修改失败提示信息。

### 2.5.8（售货员）功能8 - 添加商品

售货员可以通过使用商品的条形码或手动输入商品信息两种方式添加商品信息至系统中。若采用使用商品条形码的方式，扫描商品条形码后，系统会解析商品条码所携带的信息并将其存储至商品信息库中；若采用人工输入方式，则需要售货员手工输入商品的各项信息。

### 2.5.9（售货员）功能9 - 查看商品信息

售货员可以进入商品模块查看所有商品的信息和库存状态。

### 2.5.10（售货员）功能10 - 搜索、编辑和删除商品信息

售货员可以在商品管理模块进行搜索、编辑和删除商品信息的操作。售货员可以选择关键字，然后在搜索栏中输入搜索内容来查看指定商品的信息和库存状态。在商品信息的操作选项中有编辑和删除两个功能，售货员可以编辑商品信息或删除商品信息，在删除商品信息时，系统会反馈确认删除该商品的提示信息。

### 2.5.11（售货员）功能11 - 商品类别管理

经理在商品类别管理模块中，可以添加商品的类别、编辑类别信息和删除类别信息。在删除商品类别信息时，系统会反馈确认删除该商品类别提示信息

### 2.5.12（售货员）功能12 - 商品位置管理

售货员在商品位置管理模块中，可以添加位置信息、编辑位置信息和删除位置信息。售货员在删除位置信息时，如果被删除的位置仍存在商品，则此操作会将被删除位置信息的这些商品的位置重置为默认位置。

### 2.5.13（售货员）功能13 - 注册顾客账户

售货员登录进入售货员页面后，在顾客管理模块可以进行注册顾客账户的操作。通过输入要注册的顾客账户的账户名、顾客姓名、密码和邮箱等基本信息并确认无误后，即可提交注册信息，系统接收到信息后反馈注册是否成功的信息。若注册成功，售货员可以在顾客信息列表中看见新添加的顾客账户信息；否则，售货员需要根据系统反馈的错误提示信息重新注册顾客账户。

### 2.5.14（售货员）功能14 - 编辑、删除顾客信息和查看购物记录

售货员可以在顾客管理模块进行编辑、删除顾客信息和查看顾客购物记录的操作。在顾客信息的操作选项中有编辑、删除和查看购物记录三个功能，售货员可以编辑顾客信息或删除顾客信息，系统给予确认删除该顾客信息提示和操作反馈。售货员在对某一顾客选择查看购物记录操作后，系统界面显示要查看的指定顾客的购物记录，即可以看到该顾客购物记录的所有相关信息。

### 2.5.15（售货员）功能15 - 记录消费

售货员可以在顾客管理模块进行查看消费记录的操作。售货员在对某一顾客选择查看消费记录操作后，系统界面显示要查看的指定顾客的购物消费记录，即可以看到该顾客消费记录的所有相关信息；同时售货员也可以查看当前的所有的消费记录。

### 2.5.16（顾客）功能16 - 信息反馈

如果顾客有任何问题或想法，可以在系统中获取地址、邮件联系方式或留言给玛尔沃超市管理系统服务小队。

### 2.5.17（顾客）功能17 - 修改个人信息

顾客可以在个人信息界面修改自己的个人信息，包括姓名、电话和邮箱地址。修改信息提交后，系统会反馈修改是否成功的反馈，若修改成功，系统反馈成功并可在个人界面中看到修改后的个人信息，否则系统反馈修改失败提示信息，系统中展示的仍是修改前的信息。

### 2.5.18（顾客）功能18 - 修改顾客账号密码

顾客在登录账户之后，可以在个人信息界面修改自己的账户密码，输入原密码、新密码和确认新密码后顾客可以修改个人登录密码，系统验证原密码无误、两次密码输入一致且新密码与原密码不同后反馈操作成功信息。

### 2.5.19（顾客）功能19 - 找回顾客账号密码

顾客如果忘记了自己的账户密码后，可以在找回密码界面找回自己的账户密码，输入账户名和邮箱后，系统重置该账户密码并发送密码至顾客邮箱，顾客使用收到的密码即可登录自己的账户。

### 2.5.20（顾客）功能20 - 查看消费记录

顾客在登录账户之后，可以在个人信息界面查看自己的购物消费记录。

## 2.6 进行可行分析的方法

进行可行性分析的方法主要有战略分析、调查研究、预测技术、系统分析、模型方法等，本项目合理运用这些方法分析原有系统，进行商品市场的调查，组织团队成员进行头脑风暴，将项目可行性分析分为以下三个阶段：

1. 机会鉴定阶段。项目团队通过对社会需求、技术发展趋势和资源状况的分析。团队发现高质量的在线超市管理系统需求量极大，而符合要求的软件并不多；并且，BS设计模式已经很成熟，被用户所信赖；开发团队以软件学院为依托，因此拥有强大的技术支持和资金保障。该项目研发投入不大，但收益极高。
2. 初步可行性分析阶段。在项目前景研究的基础上，团队寻找可行解决方案和研发方向，进一步考察在线超市管理系统软件市场，在技术上进行实验和中间试验，发现可行性极高。
3. 技术经济可行性论证阶段。团队在全面分析、计算、比较、论证的基础上，对项目进行可行性定性分析，最终决定该项目可行。

# 3 可选方案

## 3.1 整体技术方案

本系统将以Windows 10为操作系统，IntelliJ IDEA为开发平台，PostgreSQL作为后台数据库。采用前后端分离的技术，前端显示页面主要为静态的html页面，后端与数据库进行交互以处理相应的业务逻辑，前后端的数据交互方式主要有Ajax的异步交互技术。在Java Web技术体系中，以Servlet充当控制器（Control）完成功能“导航”，JSP充当视图（View）完成内容的呈现。

该基本方案技术路线比较清晰，但也存在一些问题，即当业务规模比较大时，有两个明显的弊端，第一个弊端是Servlet会变得非常臃肿；第二个弊端是性能扩展性较差。因此在基础Java Web开发体系当中如果基本方案技术路线不再能满足系统需求，我们将采用Spring Boot框架来解决这两个问题，同时可以明显提升团队的开发效率，更加方便地完成Web开发。

## 3.2 功能实现方案

### 3.2.1（经理）功能1 - 添加售货员账户

经理在向有格式或长度要求的输入框中输入数据和最终全部信息填写完成提交系统时，在JavaScript事件处理函数中编写逻辑，实时验证输入合法性与正确性；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。

具体过程为：**（1）**在JSP页面中的Form表格中输入要添加的相应的信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面。

### 3.2.2（经理）功能2 - 搜索、编辑和删除售货员信息

**3.2.2.1方案一**

搜索：前端根据搜索条件以及搜索内容做请求操作到后端，然后从数据库查询，数据库链接为本地数据库，并返回前端展示结果；

编辑：若想修改某一售货员数据的信息，首先找到要修改的售货员，其次跳转到修改页面（须先获取到原来售货员的信息）；

删除：点击某售货员操作中的删除按钮时获取要删除信息的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

**3.2.2.2方案二**

搜索：制作较小的Form，提交表格后控制器得到搜索的字条，传给服务层组成一个SELECT 语句，返回结果。具体过程为**（1）**输入要查询的信息，进行粗略和精确查询，将输入的内容通过Form表格传值给Servlet；**（2）**Servlet中查询search方法调用dao层的查询search方法，在数据库中进行模糊或者精确查询，将查询到的信息装在List中传给Servlet；**（3）**JSP返回最终查询的信息；

编辑：**（1）**显示全部的数据，点击要修改的一组数据，传值（此值为要修改的id）给Servlet；**（2）**Servlet中调用Dao层的update修改方法，update方法修改数据库中数据；**（3）**操作结果返回至Servlet中，JSP弹出操作结果；

删除：**（1）**选择要删除的售货员，然后传值到Servlet中；**（2）**在Servlet中通过调用Dao层的查询方法，在数据库中查找该售货员信息；**（3）**经理确定是否要删除，若要删除则继续传值给Servlet，调用Servlet中的删除del方法，该方法调用Dao层的del函数，实现数据库中的数据的删除；**（4）**操作结果返回至Servlet中，在Servlet中判断输出一个JSP结果页面。

**3.2.2.3方案三**

搜索：全文检索框架，使用Lucene 和Solr来实现搜索系统；

编辑：选择Ajax异步或者同步提交表单数据到后台，后台进行业务逻辑处理后更新数据库里的表；

删除：点击某售货员操作中的删除按钮时获取要删除信息的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

**3.2.2.4优缺点对比**

方案一：简单基础，易于实现，适用于直查数据库数据量少的情况，数据量大的时候会带来查询速度越来越慢的问题，因为涉及到所有数据的加载和检索，效率较低。可以添加细节，来完善程序，对输入框中输入的信息要进行判断，通过Form表格进行onsubmit的判断，判断输入的内容是否合理，是否为空。

方案二：JSP文件为显示界面，包括增删改查四个界面，然后使用form或者href超链接的方法将网页获得的值传到Servlet中的方法里；而Servlet方法中调用Dao层里面的包括增删改查等方法的对象，以此实现对数据库里面的数据的增删改查，最后返回页面输出操作的结果。简单易于实现，适用于直查数据库数据量少的情况，当数据量增加甚至较大时查询效率会降低，必要时可以使用Ajax请求到后台进行查询。

方案三：搭建简单，可以直接导出数据库中表数据为文档，在文档中查询，效率极高，适用于数据量较大的系统查询。

### 3.2.3（经理）功能3 - 修改经理账户密码

#### 3.2.3.1方案一

对输入的原密码的正确性、新密码的位数进行判断，两次密码的输入内容是否一致进行比较。点击"保存"按钮后，系统提示密码修改成功。使用Ajax将前台的数据参数传到后台Controller的指定方法中，后台Controller方法接收参数id和password(修改的新密码)进行处理，在相应的Dao数据层接口中创建实现更改数据库用户密码操作的方法，里面的内容是对数据库进行更改密码的具体代码实现。

#### 3.2.3.2方案二

用JS检测两次输入的密码是否一致；**（1）**将经理输入的新密码获取到，并交给Ajax去处理，Ajax将密码参数传给后台；**（2）**添加注解@ResponseBody接收Ajax中json类型的数据参数；**（3）**在Controller的方法中使用@RequestParam接收前台传来的密码参数值；**（4）**生成6位随机数，把前台传来的密码进行MD5加密处理，再加盐值即为最终数据库需要存储的用户密码；**（5）**将最终经过加密后的密码加入到map中；**（6）**在Dao数据层接口中的实现类DaoImpl中进行实现，实现类里是对数据库用户密码修改的具体操作的代码实现；**（7）**在Service业务层接口中创建与Dao数据层接口中一致的方法，在Service业务层接口的实现类ServiceImpl中实现Service业务层接口中的此方法，方法体里调用的是Dao数据层接口中的方法；**（8）**在Controller中进行调用Service接口的此方法即可，在调用之前需要使用注解@Autowired 把Service接口装配进来；**（9）**保存用户新密码明码和随机生成的6位盐值存储数据库中。

#### 3.2.3.3优缺点对比

方案一：工作量相对较小，实现起来较容易可以满足需求。

### 方案二：工作量较大，可能无法实现全部功能，但可以防止数据库泄露，减少数据库泄露带来的损失，避免攻击者轻松获取用户名和密码，可以保护系统重要方面的密码。

### 3.2.4（经理）功能4 - 发布、编辑和删除系统公告

#### 3.2.4.1方案一

发布：经理向输入框中输入要发布的系统公告内容信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要发布的公告信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：若想修改某系统公告信息，首先找到要修改的系统公告位置，其次跳转到修改页面（须先获取到原来系统公告的信息）；

删除：点击某系统公告操作中的删除按钮时获取要删除系统公告的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.4.2方案二

发布：经理向输入框中输入要发布的系统公告内容信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要发布的公告信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：**（1）**显示全部的系统公告信息，点击要修改的一组数据，传值（此值为要修改的id）给Servlet；**（2）**Servlet中调用Dao层的update修改方法，update方法修改数据库中数据；**（3）**操作结果返回至Servlet中，JSP弹出操作结果；

删除：**（1）**选择要删除的系统公告，然后传值到Servlet中；**（2）**在Servlet中通过调用Dao层的查询方法，在数据库中查找该系统公告信息；**（3）**经理确定是否要删除，若要删除则继续传值给Servlet，调用Servlet中的删除del方法，该方法调用Dao层的del函数，实现数据库中的数据的删除；**（4）**操作结果返回至Servlet中，在Servlet中判断输出一个JSP结果页面。

#### 3.2.4.3方案三

发布：经理向输入框中输入要发布的系统公告内容信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要发布的公告信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：选择Ajax异步或者同步提交表单数据到后台，后台进行业务逻辑处理后更新数据库里的表；

删除：点击某系统公告操作中的删除按钮时获取要删除的系统公告的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.4.4优缺点对比

方案一：简单基础，易于实现，可以添加细节，来完善程序与系统功能。

方案二：简单易于实现，易于扩展功能。JSP文件为显示界面，然后使用form或者href超链接的方法将网页获得的值传到Servlet中的方法里；而Servlet方法中调用Dao层里面的包括增删改查等方法的对象，以此实现对数据库里面的数据的增删改查，最后返回页面输出操作的结果。

方案三：减轻服务器的负担，按需取数据，最大程度地减少冗余请求，用户体验相对较好，在不需要刷新页面的情况下，就可以产生局部刷新的效果，是一种交互式、快速动态网页应用。但是Ajax大量使用了JavaScript和Ajax引擎，这些取决于浏览器的支持，在编写的时候要考虑对浏览器的兼容性。

### 3.2.5（经理）功能5 - 查看收入信息（每日/月/年）

#### 3.2.5.1方案一

完成JSP连接数据库并将数据库中超市收入数据展示在浏览器页面。连接数据库分为四步**（1）**注册驱动；**（2）**获取连接；**（3）**获取Statement；**（4）**执行SQL语句并获得结果集。最终在页面上显示数据库数据。

每日/月/年财务统计表使用商业级数据图表ECharts，ECharts是一个纯JavaScript图表库，底层依赖于轻量级的Canvas类库ZRender，基于BSD开源协议，是一款非常优秀的可视化前端框架。通过标签方式直接引入构建好的 Echarts 文件，在绘图前需要为 ECharts 准备一个具备高宽的DOM容器，然后就可以通过 echarts.init 方法初始化一个 Echarts 实例并通过 setOption 方法生成一个简单图表。

#### 3.2.5.2方案二

使用Ajax，JSP页面编写Ajax脚本（jQuery API文档）访问Servlet层，Servlet获取Ajax传递的参数，传递给Service, Service层将参数传递给Dao层，利用JDBS连接数据库获取数据库中超市每日/月/年的收入信息，各级返回，利用Gson将bean对象转化为JSON对象，在JSP页面的Ajax中解析并显示到页面上。

#### 3.2.5.3优缺点对比

方案一：实现难度相对较小，同时ECharts 兼容性好，基于HTML5，有着良好的动

画渲染效果且提供直观、生动、可交互、可高度个性化定制的数据可视化图表。创新的拖拽重计算、数据视图、值域漫游等特性大大增强了用户体验，赋予了用户对数据进行挖掘、整合的能力。但缺点是界面刷新效果不好，用户体验效果相对较差。

方案二：实现稍复杂，Gson在将对象转为JSON时会将一些字符进行转义从而导致某

些链接失效无法使用，但是用户体验相对较好，在不需要刷新页面的情况下，就可以产生局部刷新的效果，是一种交互式、快速动态网页应用。

### 3.2.6（售货员）功能6 - 修改售货员密码

#### 3.2.6.1方案一

对输入的原密码的正确性、新密码的位数进行判断，两次密码的输入内容是否一致进行比较。点击"保存"按钮后，系统提示密码修改成功。使用Ajax将前台的数据参数传到后台Controller的指定方法中，后台Controller方法接收参数id和password(修改的新密码)进行处理，在相应的Dao数据层接口中创建实现更改数据库用户密码操作的方法，里面的内容是对数据库进行更改密码的具体代码实现。

#### 3.2.6.2方案二

用JS检测两次输入的密码是否一致；**（1）**将售货员输入的的新密码获取到，并交给Ajax去处理，Ajax将密码参数传给后台；**（2）**添加注解@ResponseBody接收Ajax中json类型的数据参数；**（3）**在Controller的方法中使用@RequestParam接收前台传来的密码参数值；**（4）**生成6位随机数，把前台传来的密码进行MD5加密处理，再加盐值即为最终数据库需要存储的用户密码；**（5）**将最终经过加密后的密码加入到map中；**（6）**在Dao数据层接口中的实现类DaoImpl中进行实现，实现类里是对数据库用户密码修改的具体操作的代码实现；**（7）**在Service业务层接口中创建与Dao数据层接口中一致的方法，在Service业务层接口的实现类ServiceImpl中实现Service业务层接口中的此方法，方法体里调用的是Dao数据层接口中的方法；**（8）**在Controller中进行调用Service接口的此方法即可，在调用之前需要使用注解@Autowired 把Service接口装配进来；**（9）**保存用户新密码明码和随机生成的6位盐值存储数据库中。

#### 3.2.6.3优缺点对比

方案一：工作量相对较小，实现起来较容易可以满足需求。

方案二：工作量较大，可能无法实现全部功能，但可以防止数据库泄露，减少数据库泄露带来的损失，避免攻击者轻松获取用户名和密码，可以保护系统重要方面的密码。

### 3.2.7（售货员）功能7 - 添加商品

售货员在手动向有格式或长度要求的输入框中输入数据和最终全部信息填写完成提交系统时，在JavaScript事件处理函数中编写逻辑，实时验证输入合法性与正确性；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。

具体过程为：**（1）**在JSP页面中的Form表格中输入要添加的相应的信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面。

### 3.2.8（售货员）功能8 - 查看商品信息

#### 3.2.8.1方案一

完成JSP连接数据库并将数据库数据显示在浏览器页面。连接数据库分为四步**（1）**注册驱动；**（2）**获取连接；**（3）**获取Statement；**（4）**执行SQL语句并获得结果集。最终在页面上显示数据库数据。

#### 3.2.8.2方案二

使用Ajax，JSP页面编写Ajax脚本（jQuery API文档）访问Servlet层，Servlet获取Ajax传递的参数，传递给Service, Service层将参数传递给Dao层，利用JDBS连接数据库获取数据库信息，各级返回，利用Gson将bean对象转化为JSON对象，在JSP页面的Ajax中解析并显示到页面上。

#### 3.2.8.3优缺点对比

方案一：实现难度相对较小，但缺点是界面刷新效果不好，用户体验效果相对较差。

方案二：实现稍复杂，Gson在将对象转为JSON时会将一些字符进行转义从而导致某

些链接失效无法使用，但是用户体验相对较好，在不需要刷新页面的情况下，就可以产生局部刷新的效果，是一种交互式、快速动态网页应用。

### 3.2.9（售货员）功能9 - 搜索、编辑和删除商品信息

#### 3.2.9.1方案一

搜索：前端根据搜索条件以及搜索内容做请求操作到后端，然后从数据库查询，数据库链接为本地数据库，并返回前端展示结果；

编辑：若想修改某一商品信息，首先找到要修改的商品，其次跳转到修改页面（须先获取到原来商品的信息）；

删除：点击某商品操作中的删除按钮时获取要删除商品的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.9.2方案二

搜索：制作较小的Form，提交表格后控制器得到搜索的字条，传给服务层组成一个SELECT 语句，返回结果。具体过程为**（1）**输入要查询的信息，进行粗略和精确查询，将输入的内容通过Form表格传值给Servlet；**（2）**Servlet中查询search方法调用dao层的查询search方法，在数据库中进行模糊或者精确查询，将查询到的信息装在List中传给Servlet；**（3）**JSP返回最终查询的信息；

编辑：**（1）**显示全部的数据，点击要修改的一组数据，传值（此值为要修改的id）给Servlet；**（2）**Servlet中调用Dao层的update修改方法，update方法修改数据库中数据；**（3）**操作结果返回至Servlet中，JSP弹出操作结果；

删除：**（1）**选择要删除的商品，然后传值到Servlet中；**（2）**在Servlet中通过调用Dao层的查询方法，在数据库中查找该管理员信息；**（3）**售货员确定是否要删除，若要删除则继续传值给Servlet，调用Servlet中的删除del方法，该方法调用Dao层的del函数，实现数据库中的数据的删除；**（4）**操作结果返回至Servlet中，在Servlet中判断输出一个JSP结果页面。

#### 3.2.9.3方案三

搜索：全文检索框架，使用Lucene 和Solr来实现搜索系统；

编辑：选择Ajax异步或者同步提交表单数据到后台，后台进行业务逻辑处理后更新数据库里的表；

删除：点击某商品操作中的删除按钮时获取要删除信息的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.9.4优缺点对比

方案一：简单基础，易于实现，适用于直查数据库数据量少的情况，数据量大的时候会带来查询速度越来越慢的问题，因为涉及到所有数据的加载和检索，效率较低。可以添加细节，来完善程序，对输入框中输入的信息要进行判断，通过Form表格进行onsubmit的判断，判断输入的内容是否合理，是否为空。

方案二JSP文件为显示界面，包括增删改查四个界面，然后使用form或者href超链接的方法将网页获得的值传到Servlet中的方法里；而Servlet方法中调用Dao层里面的包括增删改查等方法的对象，以此实现对数据库里面的数据的增删改查，最后返回页面输出操作的结果。简单易于实现，适用于直查数据库数据量少的情况，当数据量增加甚至较大时查询效率会降低，必要时可以使用Ajax请求到后台进行查询。

方案三：搭建简单，可以直接导出数据库中表数据为文档，在文档中查询，效率极高，适用于数据量较大的系统查询。

### 3.2.10（售货员）功能10 - 商品类别管理

#### 3.2.10.1方案一

添加：售货员向输入框中输入商品类别信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要添加商品类别信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：若想修改某一商品类别信息，首先找到要修改的商品类别，其次跳转到修改页面（须先获取到原来商品类别的信息）；

删除：点击某商品类别操作中的删除按钮时获取要删除商品类别的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.10.2方案二

添加：售货员向输入框中输入商品类别信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要添加商品类别信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：**（1）**显示全部的商品类别信息，点击要修改的一组数据，传值（此值为要修改的id）给Servlet；**（2）**Servlet中调用Dao层的update修改方法，update方法修改数据库中数据；**（3）**操作结果返回至Servlet中，JSP弹出操作结果；

删除：**（1）**选择要删除的商品类别，然后传值到Servlet中；**（2）**在Servlet中通过调用Dao层的查询方法，在数据库中查找该类别信息；**（3）**售货员确定是否要删除，若要删除则继续传值给Servlet，调用Servlet中的删除del方法，该方法调用Dao层的del函数，实现数据库中的数据的删除；**（4）**操作结果返回至Servlet中，在Servlet中判断输出一个JSP结果页面。

#### 3.2.10.3方案三

添加：售货员向输入框中输入商品类别信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要添加商品类别信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：选择Ajax异步或者同步提交表单数据到后台，后台进行业务逻辑处理后更新数据库里的表；

删除：点击某商品类别操作中的删除按钮时获取要删除信息的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.10.4优缺点对比

方案一：简单基础，易于实现，可以添加细节，来完善程序。

方案二：简单易于实现，易于扩展功能。JSP文件为显示界面，然后使用form或者href超链接的方法将网页获得的值传到Servlet中的方法里；而Servlet方法中调用Dao层里面的包括增删改查等方法的对象，以此实现对数据库里面的数据的增删改查，最后返回页面输出操作的结果。

方案三：用户体验相对较好，在不需要刷新页面的情况下，就可以产生局部刷新的效果，是一种交互式、快速动态网页应用。

### 3.2.11（售货员）功能11 - 商品位置管理

#### 3.2.11.1方案一

添加：售货员向输入框中输入商品位置信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要添加商品位置信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：若想修改某一商品位置信息，首先找到要修改的商品位置，其次跳转到修改页面（须先获取到原来商品位置的信息）；

删除：点击某商品位置操作中的删除按钮时获取要删除商品位置的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.11.2方案二

添加：售货员向输入框中输入商品位置信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要添加商品位置信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：**（1）**显示全部的商品位置信息，点击要修改的一组数据，传值（此值为要修改的id）给Servlet；**（2）**Servlet中调用Dao层的update修改方法，update方法修改数据库中数据；**（3）**操作结果返回至Servlet中，JSP弹出操作结果；

删除：**（1）**选择要删除的商品位置，然后传值到Servlet中；**（2）**在Servlet中通过调用Dao层的查询方法，在数据库中查找该位置信息；**（3）**售货员确定是否要删除，若要删除则继续传值给Servlet，调用Servlet中的删除del方法，该方法调用Dao层的del函数，实现数据库中的数据的删除；**（4）**操作结果返回至Servlet中，在Servlet中判断输出一个JSP结果页面。

#### 3.2.11.3方案三

添加：售货员向输入框中输入商品位置信息并提交系统，在JavaScript中编写事件处理函数；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。具体过程为：**（1）**在JSP页面中输入要添加商品位置信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面；

编辑：选择Ajax异步或者同步提交表单数据到后台，后台进行业务逻辑处理后更新数据库里的表；

删除：点击某商品位置操作中的删除按钮时获取要删除信息的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法。

#### 3.2.11.4优缺点对比

方案一：简单基础，易于实现，可以添加细节，来完善程序。

方案二：简单易于实现，易于扩展功能。JSP文件为显示界面，然后使用form或者href超链接的方法将网页获得的值传到Servlet中的方法里；而Servlet方法中调用Dao层里面的包括增删改查等方法的对象，以此实现对数据库里面的数据的增删改查，最后返回页面输出操作的结果。

方案三：用户体验相对较好，在不需要刷新页面的情况下，就可以产生局部刷新的效果，是一种交互式、快速动态网页应用。

### 3.2.12（售货员）功能12 - 注册顾客账户

售货员在向有格式或长度要求的输入框中输入数据和最终全部信息填写完成提交系统时，在JavaScript事件处理函数中编写逻辑，实时验证输入合法性与正确性；使用CSS选择器功能美化添加账户界面；使用Ajax网页开发技术使得在无需重新加载整个网页的情况下，能够更新部分网页界面。

具体过程为：**（1）**在JSP页面中的Form表格中输入要添加的相应的信息，通过Form表格的提交按钮将输入框Input中的内容和值（该值确定调用Servlet中的add方法）传给Servlet；**（2）**在Servlet中调用Dao层中的add方法，Dao层中的add方法；**（3）**通过DB层获取与数据库的连接，实现数据库的添加；**（4）**返回操作结果给Servlet的add方法，由Servlet中的add方法判断输出添加成功的JSP信息页面。

### 3.2.13（售货员）功能13 - 编辑、删除顾客信息和查看购物记录

#### 3.2.13.1方案一

编辑：若想修改某一顾客信息，首先找到要修改的顾客，其次跳转到修改页面（须先获取到原来顾客的信息）；

删除：点击某顾客操作中的删除按钮时获取要删除顾客的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法；

查看：每个顾客的消费记录都存储在数据库表中，完成JSP连接数据库并将数据库该表中的数据显示在浏览器页面。连接数据库分为四步**（1）**注册驱动；**（2）**获取连接；**（3）**获取Statement；**（4）**执行SQL语句并获得结果集。最终在页面上显示数据库表中数据信息。

#### 3.2.13.2方案二

编辑：**（1）**显示全部的顾客信息，点击要修改的一组数据，传值（此值为要修改的id）给Servlet；**（2）**Servlet中调用Dao层的update修改方法，update方法修改数据库中数据；**（3）**操作结果返回至Servlet中，JSP弹出操作结果；

删除：**（1）**选择要删除的顾客，然后传值到Servlet中；**（2）**在Servlet中通过调用Dao层的查询方法，在数据库中查找顾客信息；**（3）**经理确定是否要删除，若要删除则继续传值给Servlet，调用Servlet中的删除del方法，该方法调用Dao层的del函数，实现数据库中的数据的删除；**（4）**操作结果返回至Servlet中，在Servlet中判断输出一个JSP结果页面；

查看：每个顾客的消费记录都存储在数据库表中，完成JSP连接数据库并将数据库该表中的数据显示在浏览器页面。连接数据库分为四步**（1）**注册驱动；**（2）**获取连接；**（3）**获取Statement；**（4）**执行SQL语句并获得结果集。最终在页面上显示数据库表中数据信息。

#### 3.2.13.3方案三

编辑：选择Ajax异步或者同步提交表单数据到后台，后台进行业务逻辑处理后更新数据库里的表；

删除：点击某顾客操作中的删除按钮时获取要删除顾客的id，把id传递到bean中（数据库操作类），交给数据访问层执行删除方法；

查看：使用Ajax，JSP页面编写Ajax脚本（jQuery API文档）访问Servlet层，Servlet获取Ajax传递的参数，传递给Service, Service层将参数传递给Dao层，利用JDBS连接数据库获取数据库中顾客的商品消费记录信息，各级返回，利用Gson将bean对象转化为JSON对象，在JSP页面的Ajax中解析并显示到页面上。

#### 3.2.13.4优缺点对比

方案一：简单基础，易于实现，可以添加细节，来完善程序。但缺点是界面刷新效果不好，用户体验效果相对较差。

方案二：简单易于实现，易于扩展功能。JSP文件为显示界面，然后使用form或者href超链接的方法将网页获得的值传到Servlet中的方法里；而Servlet方法中调用Dao层里面的包括增删改查等方法的对象，以此实现对数据库里面的数据的增删改查，最后返回页面输出操作的结果。

方案三：实现稍复杂，Gson在将对象转为JSON时会将一些字符进行转义从而导致某些链接失效无法使用，而且Ajax可能破坏浏览器的后退与加入收藏书签功能；但是用户体验相对较好，能在不更新整个页面的前提下维护数据，可以提高系统性能，优化用户界面。

### 3.2.14（售货员）功能14 - 记录消费

#### 3.2.14.1方案一

每个顾客的消费记录都存储在数据库表中，完成JSP连接数据库并将数据库该表中的数据显示在浏览器页面。连接数据库分为四步**（1）**注册驱动；**（2）**获取连接；**（3）**获取Statement；**（4）**执行SQL语句并获得结果集。最终在页面上显示数据库表中数据信息。

#### 3.2.14.2方案二

使用Ajax，JSP页面编写Ajax脚本（jQuery API文档）访问Servlet层，Servlet获取Ajax传递的参数，传递给Service, Service层将参数传递给Dao层，利用JDBS连接数据库获取数据库中顾客的书籍消费记录信息，各级返回，利用Gson将bean对象转化为JSON对象，在JSP页面的Ajax中解析并显示到页面上。

#### 3.2.14.3优缺点对比

方案一：简单基础，易于实现，可以添加细节，来完善程序。但缺点是界面刷新效果不好，用户体验效果相对较差。

方案二：实现稍复杂，Gson在将对象转为JSON时会将一些字符进行转义从而导致某些链接失效无法使用，而且Ajax可能破坏浏览器的后退与加入收藏书签功能；但是用户体验相对较好，能在不更新整个页面的前提下维护数据，可以提高系统性能，优化用户界面。

### 3.2.15（顾客）功能15 - 信息反馈

#### 3.2.15.1方案一

JS调取电脑本地客户端发送邮件——mailto，首先安装默认的E-mail软件，类似Microsoft Outlook、YoMail、Foxmail等并设置为默认启动。

基本语法结构：

<a href=mailto:sample@163.com>send email</a>

通过JS控制to、suject、cc、bcc、body等参数即可基本使用邮箱的功能。

#### 3.2.15.2方案二

web网页直接触发发送邮件。

#### 3.2.15.3方案三

前端静态网页无需后台发送邮件留言的方式（Formspree）——邮箱留言发送工具

使用第三方工具Formspree可以表单提交和Ajax提交，将表单连接Formspree的终端，Formspree会通过电子邮件提交编写的内容。在所有表单元素（即<input>，<select>和<textarea>）中包含name属性以接收提交数据。

#### 3.2.15.4方案四

使用node中nodemail。首先需要安装node的环境，然后安装nodemailer并配置config.js，调用sendMail即可。

#### 3.2.15.5方案五

使用jMail。jMail是一个第三方邮件操作组件，通常位于web服务器端，和站点程序紧密配合来接收及提交邮件到邮件服务器的控件，让网站拥有发送邮件和接收邮件的功能。

#### 3.2.15.6优缺点对比

方案一： 简单基础，易于实现，但主题和内容可能有时需要使用urlencode编码一下，用于格式解析，否则会出现乱码。

方案二：简单易实现。

方案三：无需PHP，JavaScript或注册，适合于静态网站；但是提交的邮箱有限制且不支持个别邮箱。

方案四：使用JS发送邮件 不会调用客户端，但需要安装与配置环境。

方案五：需要下载运行jMail.dll文件安装服务，同时浏览器存在兼容问题，需要开启POP/IMAP、host配置等，不推荐使用。

### 3.2.16（顾客）功能16 - 修改个人信息

#### 3.2.16.1方案一

顾客在个人信息界面选择修改信息功能，web中通过页面id查找信息明细并显示在JSP页面上，其次弹出个人信息修改窗口才可进行修改（须先获取到原来顾客的信息）。

#### 3.2.16.2方案二

传值（此值为要修改的id）给Servlet；Servlet中调用Dao层的update修改方法，update方法修改数据库中数据；操作结果返回至Servlet中，JSP弹出操作结果。

#### 3.2.16.3方案三

选择Ajax异步或者同步提交表单数据到后台，后台进行业务逻辑处理后更新数据库里的表。

#### 3.2.16.4优缺点对比

方案一：简单基础，易于实现。

方案二：JSP文件为显示界面，包括增删改查四个界面，然后使用form或者href超链接的方法将网页获得的值传到Servlet中的方法里；而Servlet方法中调用Dao层里面的包括增删改查等方法的对象，以此实现对数据库里面的数据的增删改查，最后返回页面输出操作的结果。简单易于实现。

方案三：页面无需刷新，在页面内与服务器通信，有非常好的用户体验，使用异步的方式与服务器通信，不需要中断操作。在动态更新页面的情况下，用户无法回到前一个页面状态，因为浏览器仅能记忆历史记录中的静态页面，一个被完整读入的页面与一个已经被动态修改过的页面之间的差别非常微妙，用户通常会希望单击后退按钮能够取消他们的前一次操作，但是在使用Ajax的应用程序中，这将无法实现。

### 3.2.17（顾客）功能17 - 修改顾客账号密码

#### 3.2.17.1方案一

对输入的原密码的正确性、新密码的位数进行判断，两次密码的输入内容是否一致进行比较。点击"保存"按钮后，系统提示密码修改成功。使用Ajax将前台的数据参数传到后台Controller的指定方法中，后台Controller方法接收参数id和password(修改的新密码)进行处理，在相应的Dao数据层接口中创建实现更改数据库用户密码操作的方法，里面的内容是对数据库进行更改密码的具体代码实现。

#### 3.2.17.2方案二

用JS检测两次输入的密码是否一致；**（1）**将超级管理员输入的新密码获取到，并交给Ajax去处理，Ajax将密码参数传给后台；**（2）**添加注解@ResponseBody接收Ajax中json类型的数据参数；**（3）**在Controller的方法中使用@RequestParam接收前台传来的密码参数值；**（4）**生成6位随机数，把前台传来的密码进行MD5加密处理，再加盐值即为最终数据库需要存储的用户密码；**（5）**将最终经过加密后的密码加入到map中；**（6）**在Dao数据层接口中的实现类DaoImpl中进行实现，实现类里是对数据库用户密码修改的具体操作的代码实现；**（7）**在Service业务层接口中创建与Dao数据层接口中一致的方法，在Service业务层接口的实现类ServiceImpl中实现Service业务层接口中的此方法，方法体里调用的是Dao数据层接口中的方法；**（8）**在Controller中进行调用Service接口的此方法即可，在调用之前需要使用注解@Autowired 把Service接口装配进来；**（9）**保存用户新密码明码和随机生成的6位盐值存储数据库中。

#### 3.2.17.3优缺点对比

方案一：工作量相对较小，实现起来较容易可以满足需求。

方案二：工作量较大，可能无法实现全部功能，但可以防止数据库泄露，减少数据库泄露带来的损失，避免攻击者轻松获取用户名和密码，可以保护系统重要方面的密码。

### 3.2.18（顾客）功能18 - 找回顾客账号密码

#### 3.2.18.1方案一

Servlet中进行数据处理，根据顾客输入的邮箱账号找到顾客，并生成发送邮件类的实例，再设置收件人和要发送的内容，最后发送邮件。

此处使用授权码，授权码是用于登录第三方邮件客户端的专用密码，适用于登录以下服务：POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV等服务，需要登录邮箱设置。

#### 3.2.18.2方案二

**（1）**生成密钥UUID；**（2）**设置过期时间outdata；**（3）**生成数字签名（由UUID、userid、outdata构成）；**（4）**将数字签名相关参数保存至数据库；**（5）**产生链接，发送用户邮箱；**（6）**用户点击链接验证身份后重置密码

#### 3.2.18.3优缺点对比

方案一：工作量相对较小，实现起来较容易可以满足需求。

方案二：与邮件服务器交互且需创建用于连接邮件服务器的参数配置，工作量较大，可能无法实现全部功能，但设置链接失效时间并实现顾客自主重置密码，可有效保护顾客账户和密码安全。

### 3.2.19（顾客）功能19 - 查看消费记录

#### 3.2.19.1方案一

顾客每购物消费一次则将该条消费记录插入一张数据库表中，在顾客查看个人消费记录时，完成JSP连接数据库并将数据库该表中的数据显示在浏览器页面。连接数据库分为四步**（1）**注册驱动；**（2）**获取连接；**（3）**获取Statement；**（4）**执行SQL语句并获得结果集。最终在页面上显示数据库表中数据信息。

#### 3.2.19.2方案二

使用Ajax，JSP页面编写Ajax脚本（jQuery API文档）访问Servlet层，Servlet获取Ajax传递的参数，传递给Service, Service层将参数传递给Dao层，利用JDBS连接数据库获取数据库中顾客购物消费的记录信息，各级返回，利用Gson将bean对象转化为JSON对象，在JSP页面的Ajax中解析并显示到页面上。

#### 3.2.19.3优缺点对比

方案一：实现难度相对较小，但缺点是界面刷新效果不好，用户体验效果相对较差。

方案二：实现稍复杂，Gson在将对象转为JSON时会将一些字符进行转义从而导致某

些链接失效无法使用，但是用户体验相对较好，能在不更新整个页面的前提下维护数据。这使得Web应用程序更为迅捷地回应用户动作，并避免了在网络上发送那些没有改变的信息。

## 3.3 选择最终方案的准则

1. 能够高效处理超市管理相关的各种信息，随时随地地管理和掌握商品信息及消费记录等信息，使超市的管理更加简洁、高效和方便。
2. 费用花费少、各项功能优先次序更满足用户需求、用户友好型优先。
3. 项目开发预算符合客户预期，符合现行的各项准则。

# 4 所建议的系统

## 4.1 **对所建议的系统的说明**

1. 开发出基于B/S设计模式的在线超市管理系统，帮助经理、售货员对商品和顾客进行管理，方便售货员对商品的添加、删除、修改、查找的操作和对顾客的购物消费的操作，方便顾客查看自己的消费记录；
2. 本项目开发的系统小巧而不失使用的功能，方便超市管理人员；多种图表统计，可统计每年，每月，每日的顾客消费和超市收入情况统计以及多种排行统计等；
3. 根据售货员管理商品信息和顾客消费信息的特点对数据库进行操作，使用先进的数据库技术与数据管理技术，使数据的准确性与安全性得到很大的提高，比如对数据的增删改查给予优化，并且在用户的并行操作与用户管理方面也有极大地改善。

**4.2 数据流程和处理流程**

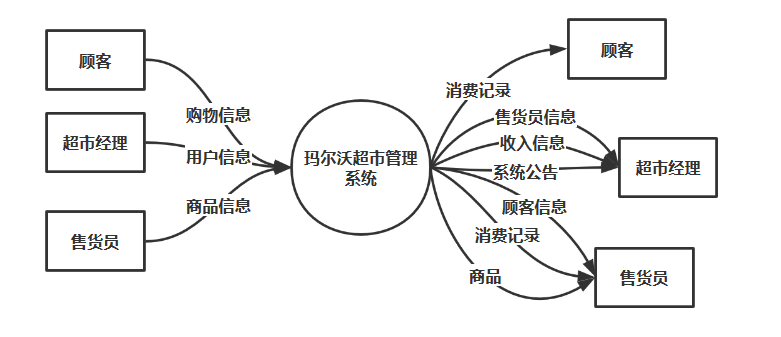


图1 玛尔沃超市管理系统数据流图

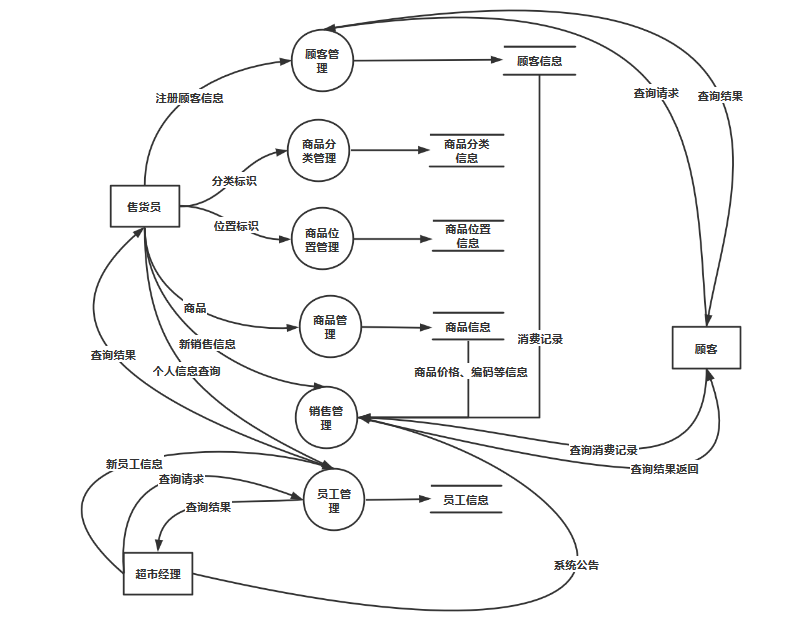


图2 玛尔沃超市管理系统细化数据流图

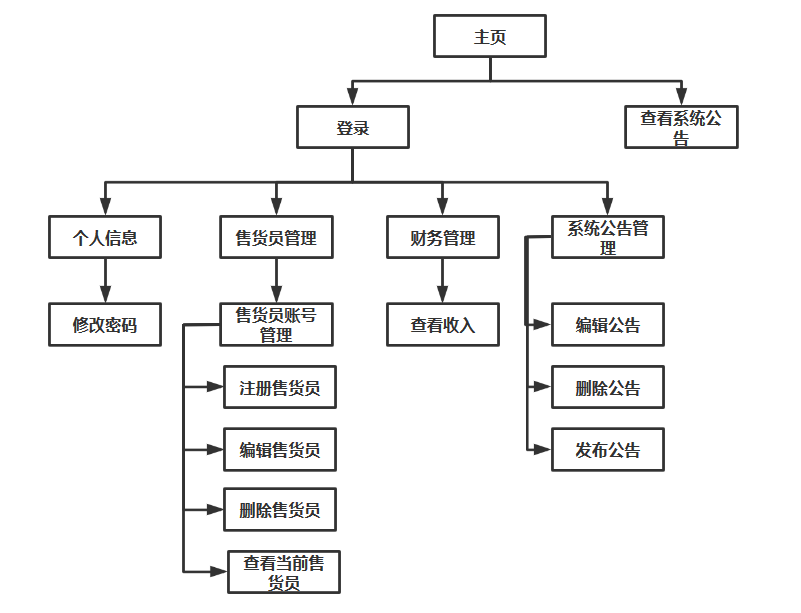


图3 经理业务处理流程

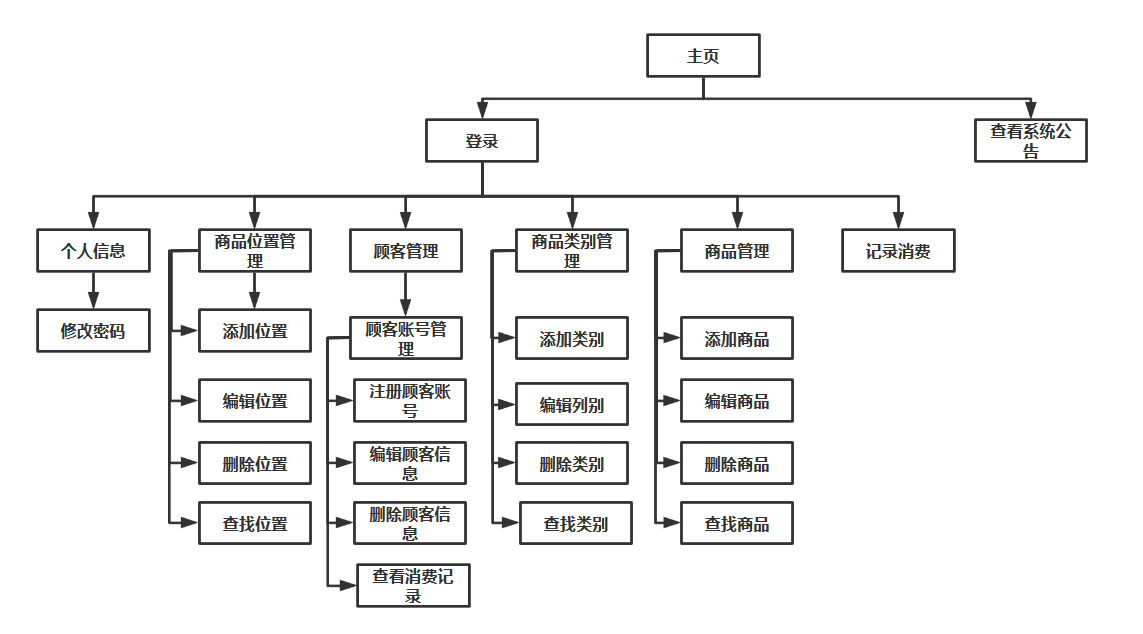


图4 售货员业务处理流程

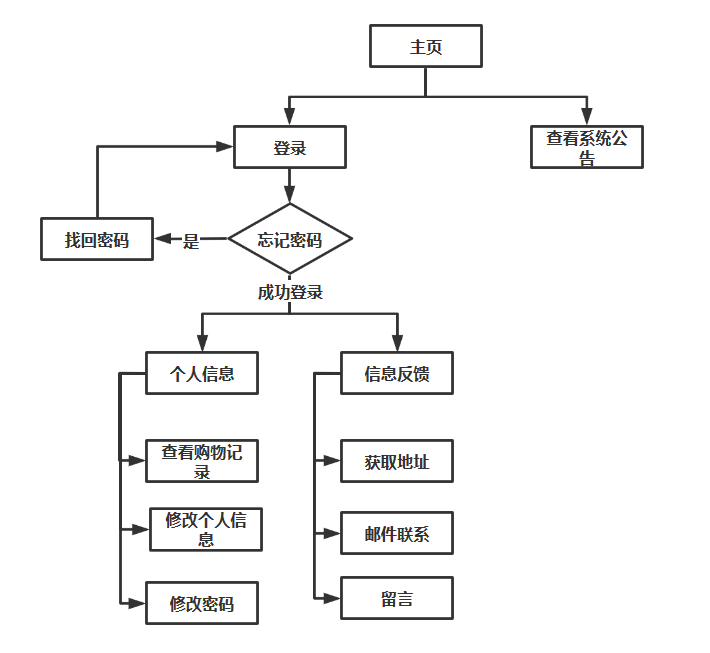


图5 顾客业务处理流程

# 5 可行性分析

超市管理工作十分重要又十分繁琐，在建立超市管理系统时需要投入一定的物力、人力和财力等，且开发的系统要具有一定的功能性和效率性。因此，必须从“经济可行性、技术可行性、用户使用可行性和法律可行性”这四个方面对玛尔沃超市管理系统进行可行性分析。

## 5.1 经济可行性

**5.1.1 投资成本**

系统的开发在设备和软件方面的支出是不可避免的但却是十分有限的，此超市管理系统并不是很复杂的系统，现在的计算机购买成本并不高，且设备也可使用原有的设备，开发周期较短，经济支出有限。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 任务 | 难度 | 重要性 | 人时 | 花费 |
| 项目设计 | ★★ | ★★★ | 15 | ￥2250 |
| 经理查看收入模块 | ★★★★ | ★★★★ | 8.5 | ￥1270.5 |
| 其他经理模块 | ★★★ | ★★★ | 20 | ￥3000 |
| 售货员管理顾客 | ★★★ | ★★★ | 8.5 | ￥1270.5 |
| 售货员管理商品 | ★★★★ | ★★★ | 10 | ￥1500 |
| 其他图书馆管理员模块 | ★★★ | ★★★ | 17 | ￥2550 |
| 顾客模块 | ★★★ | ★★★ | 23 | ￥3450 |
| 项目发布 | ★★ | ★ | 4 | ￥600 |
| 总计 | 106人时 | | | ￥15900 |

本项目计划开发预计106人时，每人时150元，总计开发费用为106×150=15900元。除了需要支付开发单位的有关费用之外，在本项目投入使用之后，每年还需要支付3500元的运行维护费用，此费用包括但不限于更新数据库信息、修复系统缺陷等费用。

### 5.1.2收益

玛尔沃超市管理系统可以减少超市的各种管理人员，减少了员工工资的开销；数据保存方便安全，可清楚地知道超市的商品去向和收入明细，及时接受市场的反馈信息。

玛尔沃超市管理系统对涉及各种商品管理的行业均有使用价值，通过超市管理系统可大大减轻工作人员的工作量，开发此系统对计算机硬件和软件配置都没有特别的需要，且开发周期较短。从系统开发到最后的实际运行，在很大程度上提高了计算机的利用率，为用户带来便利的同时，也为系统的推广创造了条件。经济效益远大于支出，更重要的是开发的软件可以给我们的管理带来更加科学的方法。故在经济上可行。

## 5.2 技术可行性

进行玛尔沃超市管理系统的开发可利用现在主流的Windows 10 操作系统，操作简捷方便。数据库系统使用对象-关系型数据库管理系统PostgreSQL，开发工具为Java 集成开发环境IntelliJ IDEA，IntelliJ IDEA是一个稳定的、成熟的平台，并且是被独立管理的。所用语言有Java、JavaScript、HTML和CSS等，开发玛尔沃超市管理系统所用的软件与语言在实际软件开发中已被大量应用，技术上都较为成熟。并且目前超市管理信息系统得到了广泛的应用，有许多的数据和信息得以处理，网络信息技术的发达保证了系统的可行性，因此系统在技术上是可行的。

**5.3 用户可行性**

如今计算机普及度十分广泛，大多数人都已掌握计算机的基本操作。曼德林图书馆管理系统凭借其简洁明了的界面和快捷的操作特性，并不要求用户对其特别的熟悉。系统操作简单，界面直观易懂，用户能很快地掌握系统使用方法。因此，能简单地对计算机进行操作的人员，对系统稍作了解便可在短时间内执行自己期望的操作。

使用Java Web过滤器实现系统超级管理员安全登录，可以防止非系统超级管理员人员误进入系统重要管理界面，确保超级管理员正常履行自己的管理职能，保障Web系统应用和数据安全。对于服务端的操作人员，由于软件设计提供给操作人员的接口仅仅会涉及到简单的文件新建、修改、复制和删除等操作，因此仅需操作人员熟悉简单的电脑操作即可，不需要进行专门培训。

该系统操作简单快捷，功能齐全，可以满足用户的基本需求且通俗易学，故使用上可行。

## 5.4 法律可行性

本软件自主开发研制，不抄袭，不盗用，不存在法律上的版权问题。软件主要功能为超市管理系统，帮助经理管理员工和查看超市营业额收入，并及时收到市场反馈信息；帮助售货员对商品和顾客进行管理，方便售货员进行商品的添加、删除、修改、查找的操作；方便顾客查看各自的购物消费记录，不存在法律政策上的问题。同时该系统的开发将不会侵犯任何个人、集体和国家的利益，也不会违反国家的政策，故具有法律可行性。