|  |  |  |
| --- | --- | --- |
| aaaaa  **本科生毕业论文（设计）**  xxxxx | | |
|  | 题　　目 | 基于Freemarker动态模板的研究与实现 |
|  |  |  |
|  | 姓　　名 | 孙宗衡 学号 2011416867 |
|  | 院　　系 | 信息科学与工程学院 |
|  | 专　　业 | 软件工程 |
|  | 指导教师 | 闫超 职称 讲师 |
| 年 月 日 | | |
| 曲阜师范大学教务处制 | | |

基于Freemarker动态模板的研究与实现

软件工程专业学生 孙宗衡

指导教师 闫超

摘要：随着互联网的飞速发展，各种网络技术也是层出不穷。对于开发人员，面对如此多的开发语言，以及各式各样的框架，在项目开发过程中又该如何选择呢？Java作为优秀的面向对象的高级语言，对服务器端程序的开发有着自己都独特的优势，许多优秀的web框架，如Spring,MyBatis,Hiberbate,Struts2等都是基于Java语言基础开发的。Java Web与Freemarker模板引擎的融合使得基于MVC思想的软件项目，开发变得更加容易。尤其是在企业宣传网站和电子商务网站中，颇受开发人员喜爱。本文对Spring框架中的MVC实现机制,结合Freemarker模板技术而进行扩展研究，设计一种新的MVC模式实现框架，论述其组成结构和运行机理。

关键词：Freermarker 电子商务 动态模板 AJAX Spring MVC

Based on Freemarker dynamic template

to research and implement

Student majoring in software engineering SunZongheng

Tutor YaoChao

**Abstract:** With the Internet developing rapidly, variety of network technologies emerge in an endless stream. For developers, there are so much of the development of language and all kinds of framework, in the project development process, we should how to choose it. As the advanced OOP language, Java has its own special advantages in server programming. Many excellent web framework such as Spring, MyBatis, Hiberbate, Struts2 and so on are based on java. Fuseing the java web and freemarker dynamic template can let the project become easier. Especially in Corporate website and e-commerce sites, it is popular with developer. In this paper, the framework of Spring implementation in MVC system fuse the Freemarker dynamic template to research, proposes a new MVC model framework. Then discuss the structure and operation mechanism.

**Key words:** Freermarker e-commerce dynamic template AJAX Spring MVC

[引言 4](#_Toc419036020)

[1背景及定义 5](#_Toc419036021)

[1.1背景及应用领域 5](#_Toc419036022)

[1.2 Freemarker定义 5](#_Toc419036023)

[1.3 FreeMarker 的特性 5](#_Toc419036024)

[1.3 Freemarker的用途 6](#_Toc419036025)

[1.4 论文结构 6](#_Toc419036026)

[2. 框架结构设计 6](#_Toc419036027)

[2.1 MVC设计模式 6](#_Toc419036028)

[2.2传统MVC框架 7](#_Toc419036029)

[2.2.1 传统MVC框架介绍 7](#_Toc419036030)

[2.2.2传统MVC框架存在的不足 8](#_Toc419036031)

[2.3 MVCc框架 8](#_Toc419036032)

[3. MVCc框架原理解析 8](#_Toc419036033)

[3.1 Freemarker运行原理 9](#_Toc419036034)

[3.2 动态数据交互 10](#_Toc419036035)

[4.模板配置 10](#_Toc419036036)

[4.1 软件架构 10](#_Toc419036037)

[4.1.1 设计描述 10](#_Toc419036038)

[4.2 Spring框架与Freemarker模板集成 10](#_Toc419036039)

[4.3 使用Freemarker生成静态HTML文件 13](#_Toc419036040)

[4.4 实现数据动态交互 16](#_Toc419036041)

[5. 模板应用实例 17](#_Toc419036042)

[5.1 需求描述 17](#_Toc419036043)

[3.2.1 系统管理端用例图 17](#_Toc419036044)

[3.2.2 卖家管理端用例图 18](#_Toc419036045)

[3.2.3 买家用例图 19](#_Toc419036046)

[5.2 功能设计 19](#_Toc419036047)

[5.2.1数据库介绍 19](#_Toc419036048)

[5.2.2 功能描述 19](#_Toc419036049)

[5.3引入MVCc框架 20](#_Toc419036050)

[5.4 功能实现 20](#_Toc419036051)

[6.系统测试 20](#_Toc419036052)

[6.1 测试计划 20](#_Toc419036053)

[6.1.1 目的 20](#_Toc419036054)

[6.1.2 测试环境 20](#_Toc419036055)

[6.1.3 测试工具 20](#_Toc419036056)

[6.1.4 测试对象 21](#_Toc419036057)

[6.2 测试执行 21](#_Toc419036058)

[6.2.7测试结果 21](#_Toc419036059)

[6.3测试结论与建议 21](#_Toc419036060)

[7.总结与展望 21](#_Toc419036061)

[谢辞 22](#_Toc419036062)

[参考文献 22](#_Toc419036063)

[附录A 23](#_Toc419036064)

[附录B 23](#_Toc419036065)

[4.3 运行设计 26](#_Toc419036066)

[4.3.1运行模块组合 26](#_Toc419036067)

[4.3.2运行控制 26](#_Toc419036068)

[4.3.3运行时间 26](#_Toc419036069)

[4.4 系统数据结构设计 27](#_Toc419036070)

[4.4.1逻辑结构设计要点 27](#_Toc419036071)

[4.4.2物理结构设计要点 27](#_Toc419036072)

[4.4.3数据结构与程序的关系 27](#_Toc419036073)

[2.3.1 数据删除及备份 27](#_Toc419036074)

# 引言

随着技术的不断发展, 越来越多的开发框架不断地进入。开发人员的视野中, 然而, 经过开发人员在实际应用中的检验后, 发现每一个框架都有着各自的优势和劣势, 那么是否 可以将众多的框架进行整合, 吸取各自的优势, 形成一套更加简洁适用的框架呢? 这正是文中内容研究的初衷。

Java作为优秀的面向对象的高级语言，对服务器端程序的开发有着自己都特的优势，许多优秀的web框架，如Spring,MyBatis,Hiberbate,Struts2等都是基于java语言基础开发的。MVC的思想受到广泛的认可，各大企业在开发项目的工程中也都采用MVC的思想进行分层。随着网络应用的快速增加,MVC模式对于Web应用的开发无疑是一种非常先进的设计思想，无论你选择哪种语言，无论应用多复杂,它都能为你理解分析应用模型时提供最基本的分析方法，为你构造产品提供清晰的设计框架，为你的软件工程提供规范的依据。

然而，Java Web在运用MVC分层的过程中，其表现层多采用jsp页面表现方式，这种方式难以摆脱tomcat的容器对象，使得软件在开发测试中变得复杂。因此，java Web与Freemarker模板引擎的融合使得基于MVC思想的软件项目，开发变得更加容易。尤其是在企业宣传网站和电子商务网站中，颇受开发人员喜爱。采用Freemarker动态模板加载技术，通过服务器端程序将静态模板文件获取，并将公共部分放入到Memeched中。当在线用户数大量增加，数据库服务器负载加大的时候，用户首次访问时，数据从数据库中取出，并将公共部分放入MemeChed,当其他用户再次访问时，首先读取Memeched的内容，通过Freemarker静态文件中的AJAX读取内容，从而减少页面渲染的时间，从物理结构上实现程序的复用。

# 1背景及定义

## 1.1背景及应用领域

目前流行的应用程序大多基于模型-视图-控制器 (Model-View-Controller, MVC)模式进行设计与实现，包括Web程序框架。轻量级Spring框架提供了对MVC模式的实现机制，它主要围绕分发器进行设计，包括可配置的处理器映射、视图解析、本地化、主题解析、支持文件上传等。该机制具有角色划分清晰、可重用业务代码等优点，但在具体应用与开发中，其实现难度大、配置复杂，且没有提供实用的数据持久化方法。  Spring框架提供了数据持久化机制，其实现方式分为 2种，一种以对象关系映射(Object Relational Mapping, ORM)工具为基础，另一种则对Java数据库连接(Java Data Base  Connectivity, JDBC)类进行封装和抽象。 前者能有效解决面向对象与关系数据库之间的不匹配问题，但必须依赖ORM工具。后者虽然简单易用，但不能有效解决面向对象与关系数据库之间的不匹配问题。鉴于此，本文对Spring框架中的MVC实现机制,结合FreeMarker模板技术而进行扩展研究，提出一种新的MVC模式实现框架，论述其组成结构和运行机理。

## 1.2 Freemarker定义

Freemarker是一个模版引擎,一个使用Java 编写，基于模版生成文本输出的通用工具。FreeMarker 被设计用来生成 HTMLWeb页面，特别是基于MVC模式的应用程序。FreeMarker具备一定的编成能力，但一般 java 程序准备要显示的数据，都由FreeMarker生成页面，并通过模版显示准备的数据。FreeMarker不是一个Web应用框架，而更适合作为一个Web应用框架的一个组件。

## 1.3 FreeMarker 的特性

1)通用性。能够按需要生成各种类型的文本，容易于嵌入到产品中,插件式模板载入器。

2)强大的模板语言。语法结构强大并且简单,支持常用的指令:if、elseif、else include等；在模板中创建和改变变量,可以在任何地方使用复杂表达式来指定值;支持宏运算,可加载参数，并支持宏嵌套。

3)通用数据模型。为了是开发者不受技术细节的打扰,你可以使用抽象的方式表示对象，告之模板开发者使用。FreeMarker 并不直接反射到 Java 对象中，Java 对象通常通过插件式对象封装,以变量方式在模板中显示。

4)智能的国际化和本地化。数字格式本地化敏感，多种不同语言的相同模板。

## 1.3 Freemarker的用途

在软件开发的实际应用过程中,Freemarker的用途大致如下：

1)作为模板输出不同的表现层。当后台数据对象和模板推送到 Freemarker 模板引擎后,将自动被解析输出相应的前端表现,无论模板是怎样的表现,Freemarker 都可以完美的将动态数据解析输出,如此一来,我们可以通过简单的参数控制 ,将同一套业务数据输出到不同的模板,从而形成个性化的展示。

2)输出特定规格的数据。在不同的应用系统之间如果需要进行数据交换 ,一般可以采用 XML 数据格式,而一旦数据接口 的规格确定，则必须按照约定的格式输出 ,此时 Freemarker 可以作为输出 XML 规范数据的工具。

3)用于代码自动生成。当前流行的Java 开发框架,有很多种,例如 Struts,Spring,Hibernate,Mybatis 等,基于这些框架开发,目的自然是为了提升开发效率,但不同的模板基本上都按相同的模式做重复的工作,为了进一步提升开发的效率,可以考虑将开发基于自动生成的代码再编辑,其效率将得到大幅提升。

## 1.4 论文结构

正文分为七个部分，下面是对各部分的简单介绍：

第一部分：绪论。该部分主要介绍了论文的选题背景、研究的意义以及目前国内外的发展状况。

第二部分：基于百度地图的停车服务平台的概述。该部分主要从用户、管理者的需求进行阐述。为该平台需要提供的服务，做了简单的分析，并为项目的开展确定了方向。

第三部分：可行性分析。该部分从市场可行性、技术可行性和项目存在的难点及解决的办法分析百度地图集成系统的开发与实现的可行性。

第四部分：需求分析。该部分从系统整体、停车场管理和用户三个方面做需求分析。

第五部分：概要设计。该部分先设计开发环境及技术储备，然后设计数据库和接口，之后运行设计，最后设计系统数据结构和系统出错处理。

第六部分：详细设计。该部分详细服务端功能、停车场管理端和Android客户端功能设计。

第七部分：总结与展望。该部分分为测试计划、测试执行和测试结论与建议三部分。

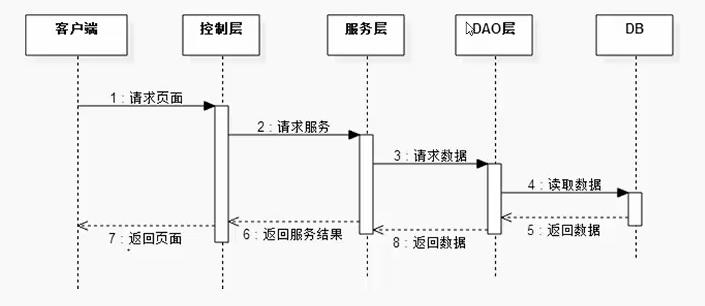
# 2. 框架结构设计

## 2.1 MVC设计模式

MVC是一种目前广泛流行的软件设计模式，早在70年代，IBM就推出了Sanfronscisico项目计划，其实就是MVC设计模式的研究。近来，随着J2EE的成熟，它正在成为在J2EE平台上推荐的一种设计模型，也是广大Java开发者非常感兴趣的设计模型。随着网络应用的快速增加,MVC模式对于Web应用的开发无疑是一种非常先进的设计思想，无论你选择哪种语言，无论应用多复杂,它都能为你理解分析应用模型时提供最基本的分析方法，为你构造产品提供清晰的设计框架，为你的软件工程提供规范的依据。

## 2.2传统MVC框架

### 2.2.1 传统MVC框架介绍



MVC即Model-View-Controller，即把一个应用的输入、处理、输出流程按照Model、View、Controller的方式进行分离，这样一个应用被分成三个层――模型层、视图层、控制层。

视图(View)代表用户交互界面，对于Web应用来说，可以概括为HTML界面，但有可能为XHTML、XML和Applet。随着应用的复杂性和规模性，界面的处理也变得具有挑战性。一个应用可能有很多不同的视图，MVC设计模式对于视图的处理仅限于视图上数据的采集和处理，以及用户的请求，而不包括在视图上的业务流程的处理。业务流程的处理交予模型(Model)处理。比如一个订单的视图只接受来自模型的数据并显示给用户，以及将用户界面的输入数据和请求传递给控制和模型。

模型(Model)就是业务流程/状态的处理以及业务规则的制定。业务流程的处理过程对其它层来说是黑箱操作，模型接受视图请求的数据，并返回最终的处理结果。业务模型的设计可以说是MVC最主要的核心。目前流行的EJB模型就是一个典型的应用例子，它从应用技术实现的角度对模型做了进一步的划分，以便充分利用现有的组件，但它不能作为应用设计模型的框架。它仅仅告诉你按这种模型设计就可以利用某些技术组件，从而减少了技术上的困难。对一个开发者来说，就可以专注于业务模型的设计。MVC设计模式告诉我们，把应用的模型按一定的规则抽取出来，抽取的层次很重要，这也是判断开发人员是否优秀的设计依据。抽象与具体不能隔得太远，也不能太近。MVC并没有提供模型的设计方法，而只告诉你应该组织管理这些模型，以便于模型的重构和提高重用性。我们可以用对象编程来做比喻，MVC定义了一个顶级类，告诉它的子类你只能做这些，但没法限制你能做这些。这点对编程的开发人员非常重要。业务模型还有一个很重要的模型那就是数据模型。数据模型主要指实体对象的数据保存（持续化）。比如将一张订单保存到数据库，从数据库获取订单。我们可以将这个模型单独列出，所有有关数据库的操作只限制在该模型中。

控制(Controller)可以理解为从用户接收请求, 将模型与视图匹配在一起，共同完成用户的请求。划分控制层的作用也很明显，它清楚地告诉你，它就是一个分发器，选择什么样的模型，选择什么样的视图，可以完成什么样的用户请求。控制层并不做任何的数据处理。例如，用户点击一个连接，控制层接受请求后, 并不处理业务信息，它只把用户的信息传递给模型，告诉模型做什么，选择符合要求的视图返回给用户。因此，一个模型可能对应多个视图，一个视图可能对应多个模型。

### 2.2.2传统MVC框架存在的不足

## 2.3 MVCc框架

2.3.1 MVCc框架组成

1）Spring MVC

Spring MVC是当前最优秀的MVC框架，自从Spring 2.5版本发布后，由于支持注解配置，易用性有了大幅度的提高。Spring 3.0更加完善，实现了对Struts 2的超越。现在越来越多的开发团队选择了Spring MVC。

Spring MVC的优点：

1、Spring MVC使用简单，学习成本低。

2、Spring MVC很容易就可以写出性能优秀的程序，Struts2要处处小心才可以写出性能优秀的程序。

3、Spring MVC的灵活是你无法想像的。

2) AJAX

AJAX即“Asynchronous Javascript And XML”（异步JavaScript和XML），是指一种创建交互式网页应用的网页开发技术。 AJAX = 异步 JavaScript和XML（标准通用标记语言的子集）。 AJAX 是一种用于创建快速动态网页的技术。 通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。 传统的网页（不使用 AJAX）如果需要更新内容，必须重载整个网页页面。

AJAX不是一种新的编程语言，而是一种用于创建更好更快以及交互性更强的Web应用程序的技术。 使用Javascript向服务器提出请求并处理响应而不阻塞用户！核心对象XMLHTTPRequest。通过这个对象，您的 JavaScript 可在不重载页面的情况与Web服务器交换数据。 AJAX 在浏览器与 Web 服务器之间使用异步数据传输（HTTP 请求），这样就可使网页从服务器请求少量的信息，而不是整个页面。 AJAX 可使因特网应用程序更小、更快，更友好。

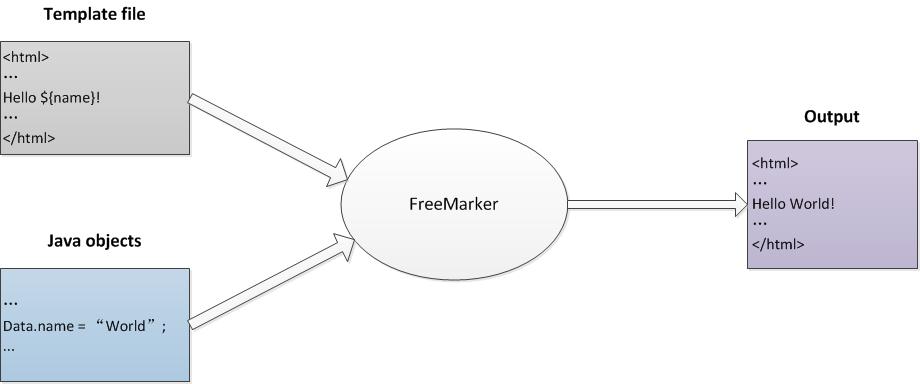
3) Freemarker

# 3. MVCc框架原理解析

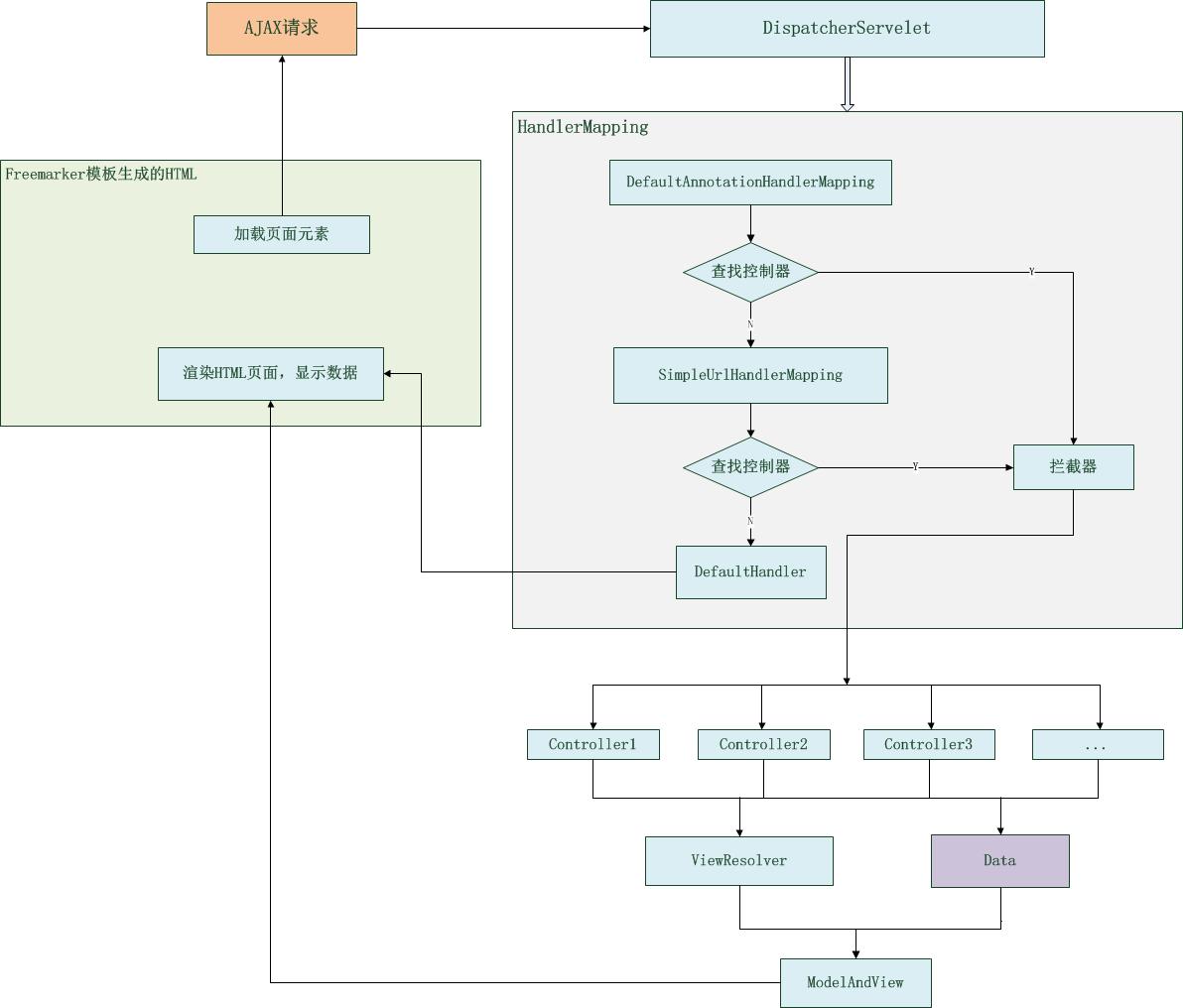
采用Freemarker动态模板加载技术，通过服务器端程序将静态模板文件获取，并将公共部分放入到Memeched中。当在线用户数大量增加，数据库服务器负载加大的时候，用户首次访问时，数据从数据库中取出，并将公共部分放入MemeChed,当其他用户再次访问时，首先读取Memeched的内容，通过Freemarker静态文件中的AJAX读取内容，从而减少页面渲染的时间，从物理结构上实现程序的复用。

## 3.1 Freemarker运行原理

很多采用 MVC 模式的软件架构都使用 FreeMarker 来实现模板功能,支持从任意位置加载模板;从类路径下,从数据库中等,易于定义特设的宏和函数。它的核心原理是: 数据模型 + 模版 = 输出。



## 3.2 动态数据交互



# 4.模板配置

## 4.1 软件架构

### 4.1.1 设计描述

## 4.2 Spring框架与Freemarker模板集成

1. 复制freemarker.jar到 Web 应用程序的 WEB-INF/lib 目录下。
2. Spring框架集成Freemarker配置文件：

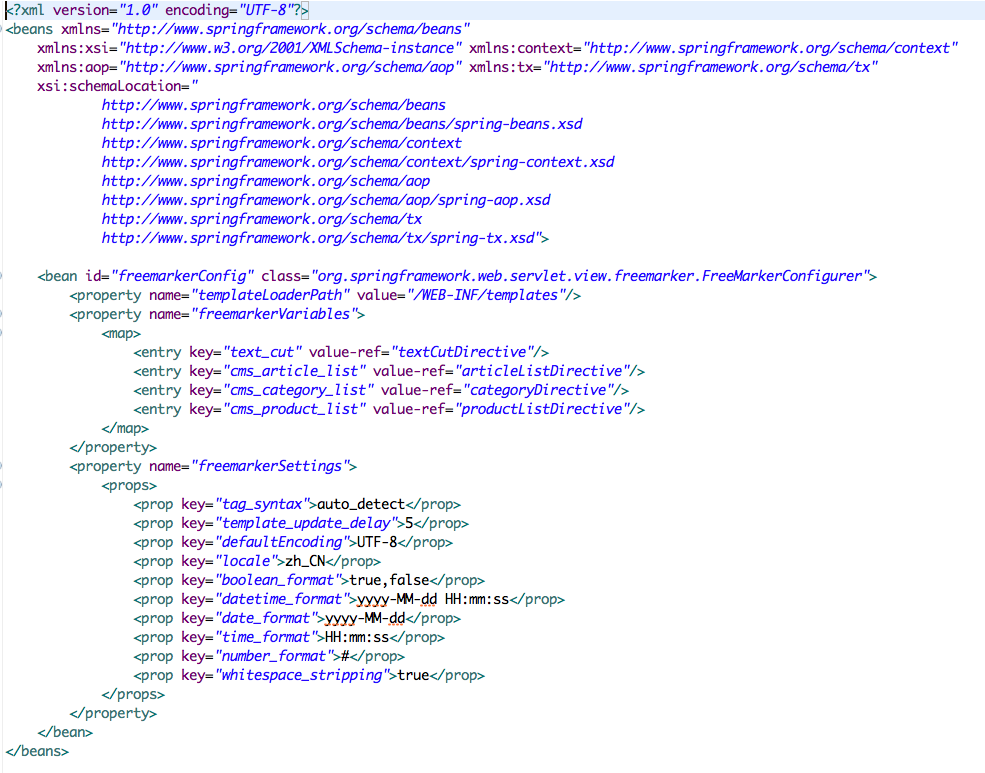
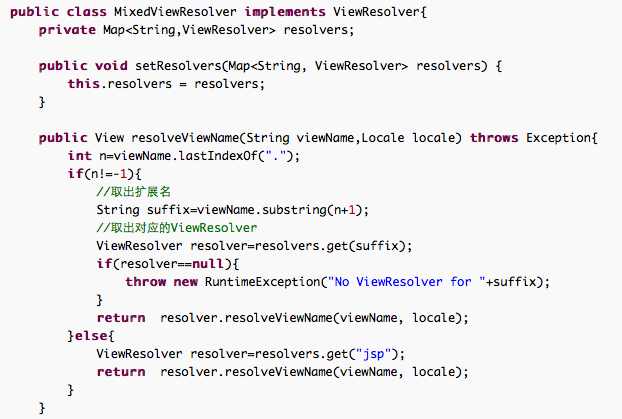


图5

1. Spring MVC 多视图配置：





## 4.3 使用Freemarker生成静态HTML文件

1） 获取模板文件生成HTML文件：



图6

2） 通过Freemarker的方式获取商品分类列表，并传入到HTML页面：



图8

3） 显示商品列表：

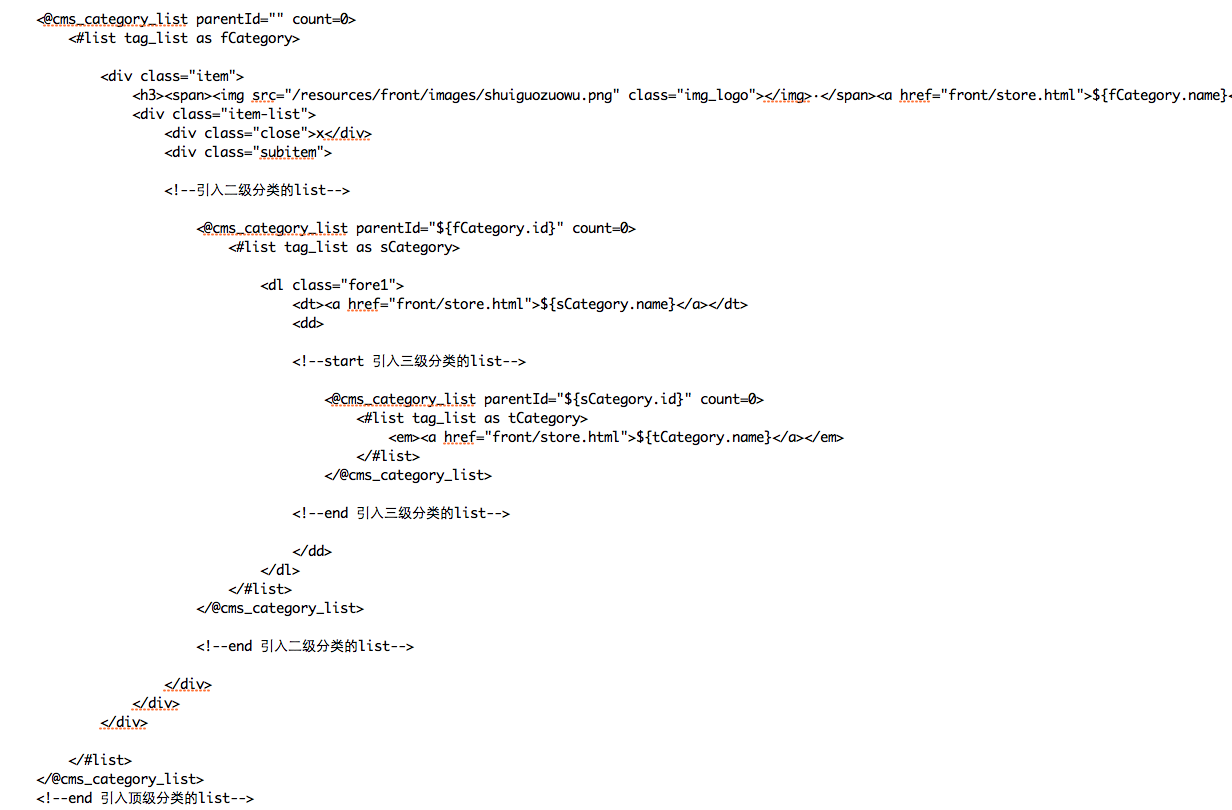


图11

## 4.4 实现数据动态交互

1） 引入下面两个jar包

jackson-core-asl-1.7.2.jar

jackson-mapper-asl-1.7.2.jar

2） Spring的配置文件中要有这一行，才能使用到spring内置支持的json转换。如果你手工把POJO转成json就可以不须要使用spring内置支持的json转换。

<mvc:annotation-driven />

1. 使用@ResponseBody注解
3. @RequestMapping("/ajax")
4. @ResponseBody
5. public Object ajax(HttpServletRequest request){
6. List<String> list=new ArrayList<String>();
7. list.add("电视");
8. nbsp;       list.add("洗衣机");
9. list.add("冰箱");
10. list.add("电脑");
11. list.add("汽车");
12. list.add("空调");
13. list.add("自行车");
14. list.add("饮水机");
15. list.add("热水器");
16. return list;
17. }

# 5. 模板应用实例

## 5.1 需求描述

### 3.2.1 系统管理端用例图

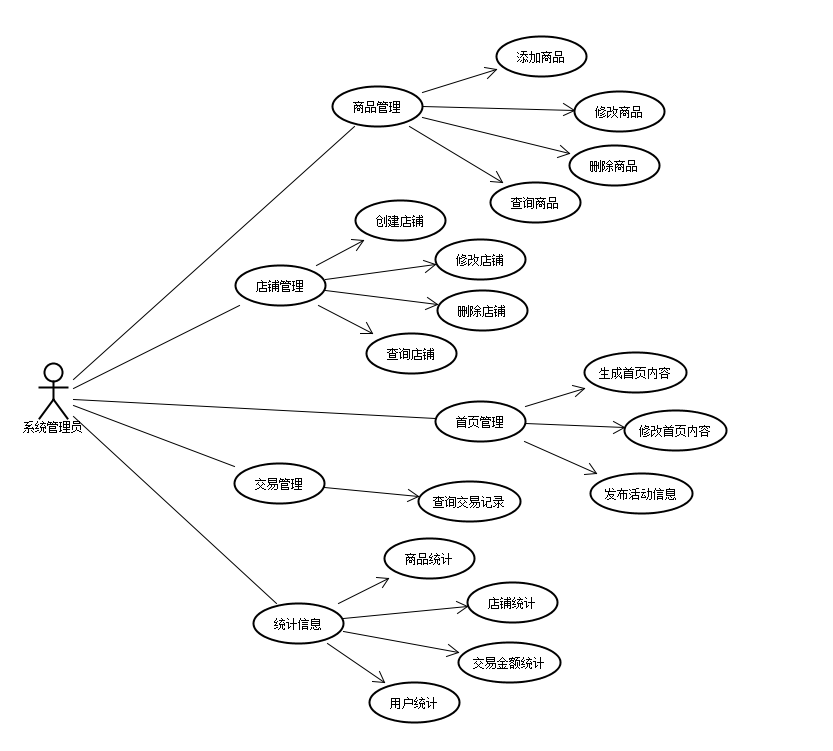


图2

### 3.2.2 卖家管理端用例图

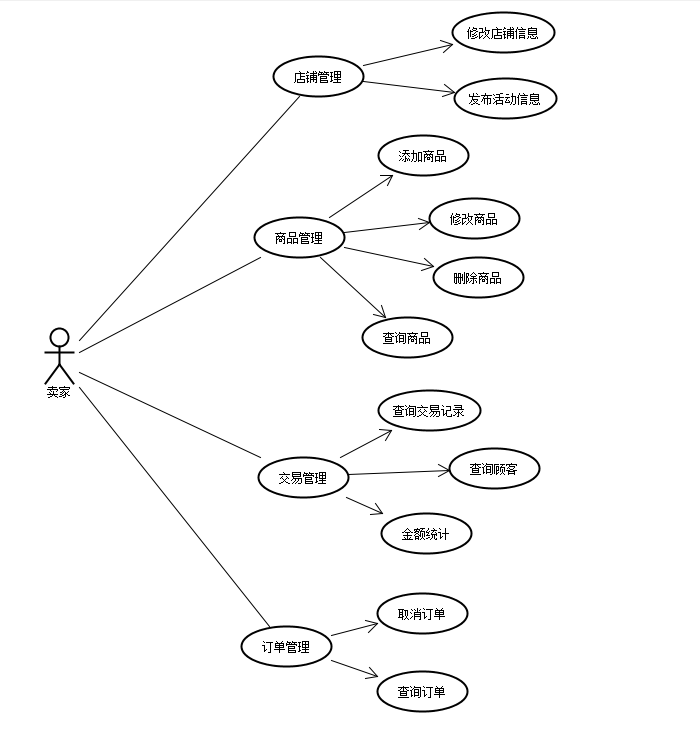


图3

### 3.2.3 买家用例图

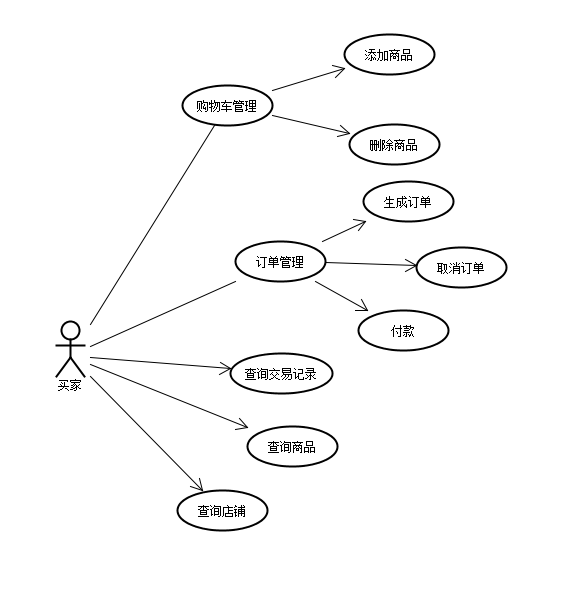


图4

## 5.2 功能设计

### 5.2.1数据库介绍

系统采用MySql数据库，因为MySql很适合中小型网站的数据库，加上它体积小、速度快、总体拥有成本低，所以系统采用 MySql数据库。它是一种关联数据库管理系统，将数据保存在不同的表中，而不是将所有数据都放在一个大仓库之中，增加速度并且提高灵活性。

### 5.2.2 功能描述

1）发布商品

2）修改商品

3）删除商品

4）查询商品

5）查询店铺

6) 创建店铺

7）修改店铺

8）删除店铺

9）查询历史记录

10）添加商品到购物车

11）生成订单

12）付款

13）取消订单

14）退款

15）查询订单

16）编辑购物车

## 5.3引入MVCc框架

## 5.4 功能实现

# 6.系统测试

## 6.1 测试计划

### 6.1.1 目的

确定基于电子商务网站系统测试环境，测试工具，测试范围

### 6.1.2 测试环境

|  |  |
| --- | --- |
| **环境名称** |  |
| 软件环境 | Windows 7 |
| 性能测试工具 | LoadRunner |
| 功能测试工具 | UFT |
| 缺陷管理工具 | TestLink |

表5

### 6.1.3 测试工具

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **用途** | **工具** | **生产厂商/自产** | **版本** | **备注** |
| 测试管理 | TestLink | 开源软件 |  | 中文版本 |
| 功能测试 | UFT | HP |  | 中文版本 |
| 性能测试 | LoadRunner | HP |  | 英文版本 |
| 被测系统 | 电子商务网站系统 |  | 9.2 | 中文版本 |
| 报告以及测试用例 | Excel/Word | Microsoft |  | 中文版本 |

表6

### 6.1.4 测试对象

本次测试对象为电子商务网站系统，语言为中文。

本次系统测试要求包含以下业务流程：

功能测试：

1）发布商品

2）修改商品

3）删除商品

4）查询商品

5）查询店铺

6) 创建店铺

7）修改店铺

8）删除店铺

9）查询历史记录

10）添加商品到购物车

11）生成订单

12）付款

13）取消订单

14）退款

15）查询订单

16）编辑购物车

界面测试：

1）界面规范，颜色、风格搭配；

2) 页面布局合理，人性化；

3) 界面文字信息准确；

4) 系统界面中的窗体与各种控件可正常显示和使用，易用性好；

5) Tab键、enter键、快捷键等可以正常使用

性能测试：

1) 系统支持的在线用户数不低于500。

2）查询商品相关操作的平均响应时间不超过3s。

## 6.2 测试执行

### 6.2.7测试结果

## 6.3测试结论与建议

# 7.总结与展望

本文主要介绍 FreeMarker模板技术, 同时结合具体例子阐述了它的具体应用过程和方法。在软件开发中碰到具有相同格式的输出对象的功能, 可以通过

FreeMarker模板技术来管理和设计, 这样可以提高程序的复用程度和开发效率, 同时程序也具有更好的维护性。不过在实际应用中需要把握一个度,程序也不能过于模板化管理, 模板化的程序易读性差, 模板管理主要是为了提高程序可复用性, 减少重复性的工作, 提高程序的灵活性。

采用Freemarker动态模板加载技术，通过服务器端程序将静态模板文件获取，并将公共部分放入到Memeched中。当在线用户数大量增加，数据库服务器负载加大的时候，用户首次访问时，数据从数据库中取出，并将公共部分放入MemeChed,当其他用户再次访问时，首先读取Memeched的内容，通过Freemarker静态文件中的AJAX读取内容，从而减少页面渲染的时间，从物理结构上实现了程序的复用，大大提升了系统性能，优化了用户体验。

# 谢辞

# 参考文献

1] 曲建明，陈杰. 模板引擎动态生成网页的应用研究[D]. 江西：计算机技术研究所，2014.

[2] 鞠凤娟，侯茜慧，肖守柏. FREEMARKER结合STRUTS2框架开发WEB研究[D].江西 南昌：蓝天学院，2014.

[3] 曾玉林. Freemarker在JSP项目中Word报表应用研究[D].四川：警察学院计算机科学与技术系，2014.

# 附录A

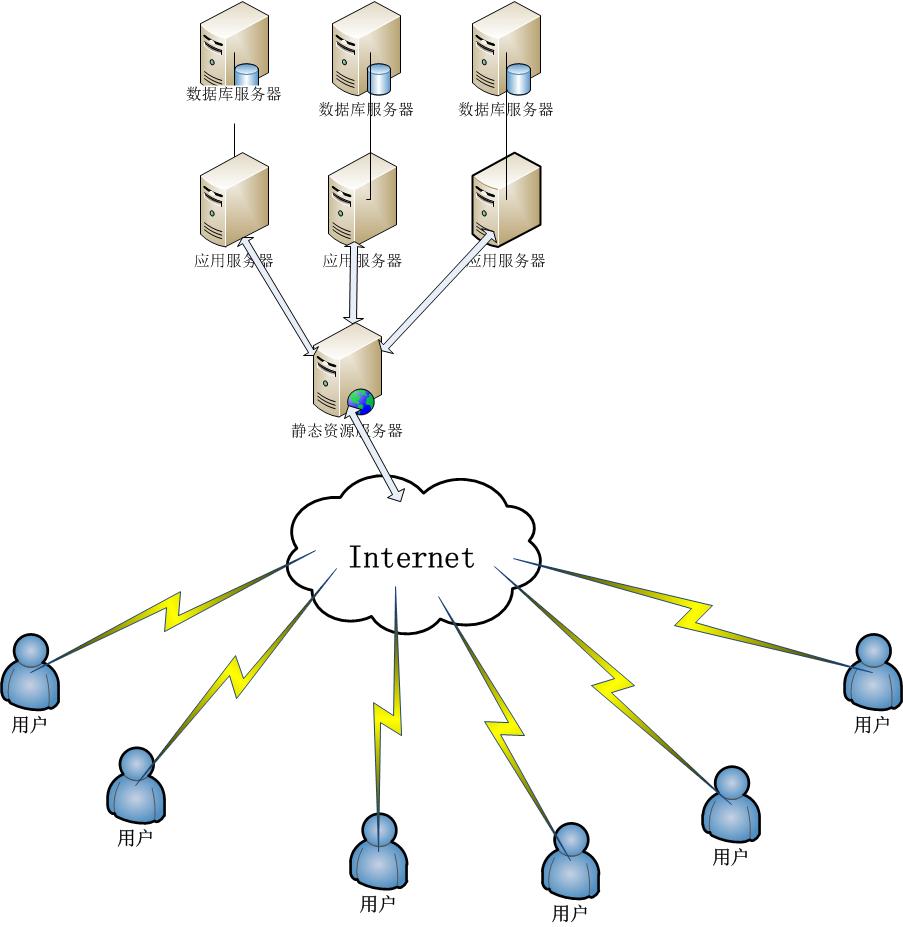


图1

# 附录B

商品表：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 列名 | | 数据类型 | 长度 | | | 备注 | 说明 | |
| id | Integer | | | 11 | 主键 | | | 自增ID | |
| name | varchar | | | 64 |  | | | 商品名称 | |
| no | varchar | | | 20 |  | | | 商品货号 | |
| category\_id | Integer | | |  |  | | | 商品分类id | |
| sell\_price | double | | |  |  | | | 销售价格 | |
| market\_price | double | | |  |  | | | 市场价格 | |
| cost\_price | double | | |  |  | | | 成本价格 | |
| up\_time | date | | |  |  | | | 上架时间 | |
| down\_time | date | | |  |  | | | 下架时间 | |
| create\_time | date | | |  |  | | | 创建时间 | |
| store\_nums | Integer | | |  |  | | | 库存数量 | |
| img | varchar | | | 255 |  | | | 原图 | |
| is\_del | tinyint | | | 1 |  | | | 删除 | |
| content | varchar | | | 256 |  | | | 商品描述 | |
| search\_words | text | | |  |  | | | 产品搜索词库 | |
| weight | double | | |  |  | | | 重量 | |
| store\_id | integer | | |  |  | | | 店铺ID | |
| visit | integer | | |  |  | | | 浏览次数 | |
| favorite | integer | | |  |  | | | 收藏次数 | |
| sort | integrer | | |  |  | | | 排序 | |
| comments | integer | | |  |  | | | 评论次数 | |
| sale | integer | | |  |  | | | 销量 | |
| grade | integer | | |  |  | | | 评分总数 | |

表1

购物车表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 备注 | 说明 |
| id | integer |  | 主键 |  |
| user\_id | integer |  |  | 用户ID |
| content | varchar | 256 |  | 购物内容 |
| create\_time | data |  |  | 创建时间 |

表2

订单表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 备注 | 列说明 |
| id | INTEGER |  |  | 自增ID |
| no | VARCHAR | 20 |  | 订单号 |
| user\_id | INTEGER |  |  | 用户ID |
| pay\_type | INTEGER |  |  | 支付方式ID,0表示货到付款 |
| distribution | INTEGER |  |  | 配送ID |
| status | Integer |  |  | 订单状态。1：生成订单；2：确认订单；3：取消订单；4：作废订单；5：完成订单 |
| pay\_status | Integer |  |  | 支付状态。0：未支付；1：已支付；2：退款 |
| distribution\_status | integer |  |  | 配送状态。0：未配送；1：已配送 |
| accept\_name | VARCHAR | 64 |  | 收货人姓名 |
| postcode | VARCHAR | 8 |  | 邮编 |
| telphone | VARCHAR | 16 |  | 联系电话 |
| country | INTEGER |  |  | 国ID |
| province | INTEGER |  |  | 省ID |
| city | INTEGER |  |  | 城市ID |
| area | INTEGER |  |  | 地区ID |
| address | VARCHAR | 250 |  | 收货地址 |
| mobile | VARCHAR | 16 |  | 手机 |
| payable\_amount | DOUBLE |  |  | 应付商品总金额 |
| real\_amount | DOUBLE |  |  | 实付商品总金额 |
| payable\_freight | DOUBLE |  |  | 总运费 |
| real\_freight | DOUBLE |  |  | 实际运费 |
| pay\_time | DATE |  |  | 付款时间 |
| send\_time | DATE |  |  | 发货时间 |
| create\_time | DATE |  |  | 下单时间 |
| completion\_time | DATE |  |  | 订单完成时间 |
| invoice | INTEGER |  |  | 发票。0：索要；1：不索要 |
| postscript | VARCHAR | 255 |  | 用户附言 |
| note | TEXT |  |  | 管理员备注 |
| if\_del | INTEGER |  |  | 是否删除。1为删除 |
| insured | DOUBLE |  |  | 报价 |
| if\_insured | INTEGER |  |  | 是否报价。0：保价；1：不包价 |
| pay\_fee | DOUBLE |  |  | 支付手续费 |
| invoice\_title | VARCHAR | 100 |  | 发票抬头 |
| taxes | DOUBLE |  |  | 税金 |
| order\_amount | DOUBLE |  |  | 订单总金额 |
| Print\_type | VARCHAR | 255 |  | 已打印的类型，类型用逗号隔开。Shop购物单，pick配货单，merge购物和配货，express快递单 |
| accept\_time | DATE |  |  | 用户收货时间 |
| trade\_no | VARCHAR | 255 |  | 支付平台交易号 |

表3

商品订单关联表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 备注 | 列说明 |
| id | integer |  | 主键 | 自增ID |
| order\_id | integer |  |  | 订单ID（外键） |
| product\_id | integer |  |  | 商品ID（外键） |
| img | varchar | 255 |  | 商品图片 |
| Product\_price | double |  |  | 商品价格 |
| real\_price | double |  |  | 实付金额 |
| nums | int |  |  | 商品数量 |
| weight | double |  |  | 重量 |
| array | text |  |  | 商品信息json数据格式 |

表4

## 4.3 运行设计

### 4.3.1运行模块组合

具体软件的运行模块组合为程序多窗口的运行环境，各模块在软件运行

过程中能较好的交换信息并处理数据。

### 4.3.2运行控制

软件运行时有友好接口，几本能够实现用户的数据处理要求。

### 4.3.3运行时间

大量用户同时进行数据的提交会使响应时间延长，响应时间会相对延长，但仍可在用户接受的等待时间内将页面呈现给用户。

## 4.4 系统数据结构设计

### 4.4.1逻辑结构设计要点

由所提供的数据结构描述，大体给出了所需数据的具体类型及相应层次

关系等基本信息，在此不多做论述。

### 4.4.2物理结构设计要点

由于所给出的数据结构和数据项太多，目前无法一一确定相应的访问方

式。存取单位已经标明。目前暂无相应策略，待编码阶段对设计考虑和保密

条件进行补充。

### 4.4.3数据结构与程序的关系

数据结构以类的方式进行组织。

访问数据结构中的数据项通过相应对象Get，Set等方法，进行良好的数

据封装和保护。

2.3 系统设计开发中存在的难点及解决办法

### 2.3.1 数据删除及备份

在用户添加商品到购物车或者购买商品之后，管理员或者卖家对商品信息（包括名称、图片和价格等）进行修改，甚至删除了该商品。此时，如果用户再次查询购物车和交易记录，会造成数据与之前不一致甚至丢失。

解决方案及各方案优缺点比较：

方案1：在管理者修改商品信息时，添加的一条新的商品，并且保留原始商品信息；在管理者删除商品时，采用逻辑删除的方式，将商品状态设置为不可用。

优点：该方案，只需在商品表中添加一个字段status,用来表示商品是否可用。

缺点：如果同一个商品被多次更改后，会生成多条新的记录，或造成商品库的存在大量的无效数据，从而对系统的性能造成严重的影响。

方案2：在买家添加商品到购物车或者购买商品的时候，将商品信息拷贝到一个新的表（user\_product）进行备份。用户的交易记录只与user\_product表进行关联。管理者对数据进行修改、删除时，只针对原始商品表。

优点：使商品库变得纯洁，减少了商品表中的冗余信息。

缺点：会在数据库中多添加一张表，此表中的会产生大量信息，该表大部分字段会与商品表中的信息一致，造成数据的冗余。

为了保持商品库的纯洁，在本系统中采用方案2，为了弥补方案2中的缺点，为管理员提供定期清除备份表信息的功能，限制用户只能查询一段时间内的交易记录。