

**<<超市购物管理系统>>**

**系统实现总结报告**



北京航空航天大学

2016-1

目 录

[1. 系统重述 1](#_Toc18824)

[1.1 系统重述 1](#_Toc4666)

[1.2 实现环境 1](#_Toc31507)

[2. 系统功能结构图（SC） 2](#_Toc9522)

[3. 基本表 2](#_Toc12421)

[3.1 基本表的定义 2](#_Toc2663)

[3.2 完整性约束 3](#_Toc32351)

[3.2.1 实体完整性 3](#_Toc32270)

[3.2.2 参照完整性 3](#_Toc2793)

[4. 安全性设计 3](#_Toc9359)

[5. 存储过程、触发器和函数 4](#_Toc11048)

[5.1 存储过程 4](#_Toc8239)

[5.1.1 说明 4](#_Toc23636)

[5.1.2 数据库端代码 4](#_Toc29988)

[5.1.3 服务器端代码 5](#_Toc9769)

[5.2 触发器 5](#_Toc14128)

[5.2.1 说明 5](#_Toc5252)

[5.2.2 触发器代码 5](#_Toc5435)

[5.3 函数 5](#_Toc11055)

[5.3.1 说明 5](#_Toc14972)

[5.3.2 数据库端代码 6](#_Toc3273)

[5.3.3 服务器端代码 6](#_Toc31285)

[6. 主要技术论述 6](#_Toc5674)

[7. 运行实例 7](#_Toc8720)

[8. 源程序简要说明 10](#_Toc31161)

[9. 收获和体会 11](#_Toc28830)

# 系统重述

## 系统重述

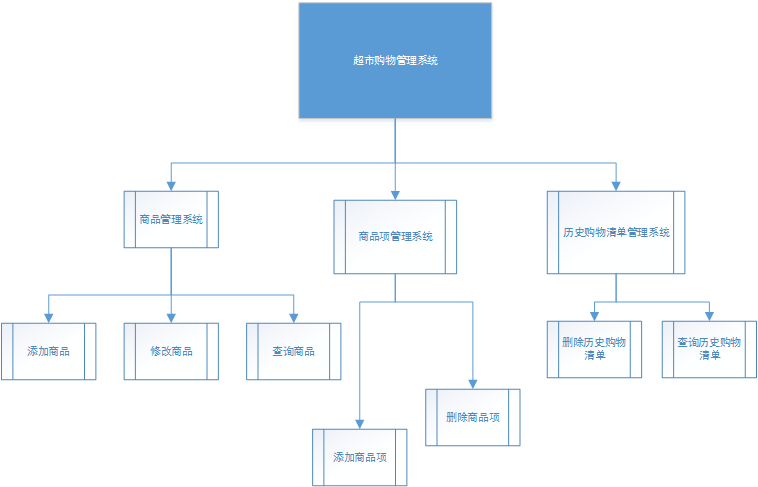
超市购物管理系统适用于购物、商品管理，用户可以查看商品信息、处理购物清单、下单等。管理员可以添加商品、修改商品信息、查看库存统计等。本系统采用MVC框架，力求模型、视图、控制器分离。

## 实现环境

**表1-1. 实现环境**

|  |  |
| --- | --- |
| 环境 | 声明 |
| 数据库管理系统 | SQL Server2014 Express |
| 应用的设计语言 | JSP |
| IDE | Eclipse J2EE |
| 服务器 | tomcat v7.0 |
| 操作系统 | windows10 |
| 版本控制系统 | github(github.com/CsterKuroi/dbProject) |
| 项目名称 | KuroiDBProject |
| 数据库名称 | jsp |
| 数据库账号 | sa |
| 数据库密码 | 123 |

# 系统功能结构图（SC）



**图2-1. 系统功能结构图**

# 基本表

items：id, name, city, price,number,picture

cartitem：itemid,cartid,number

cart：id,date,money

## 基本表的定义

CREATE TABLE [dbo].[items](

[id] [int] IDENTITY(1,1) NOT NULL,

[name] [varchar](50) NULL DEFAULT (NULL),

[city] [varchar](50) NULL DEFAULT (NULL),

[price] [int] NULL DEFAULT (NULL),

[number] [int] NULL DEFAULT (NULL),

[picture] [varchar](500) NULL DEFAULT (NULL));

CREATE TABLE [dbo].[carts](

[id] [int] IDENTITY(1,1) NOT NULL,

[date] [nchar](50) NULL,

[money] [nchar](10) NULL）;

CREATE TABLE [dbo].[cartitems](

[itemid] [int] NOT NULL,

[cartid] [int] NOT NULL,

[number] [int] NOT NULL

)

## 完整性约束

### 3.2.1 实体完整性

items主码为id

carts主码为id

cartitems主码为(itemid,cartid)

### 3.2.2 参照完整性

cartitems里有：

itemid references item(id)

cartid references cart(id)

# 安全性设计

**表4-1. 安全性设计**

|  |  |  |
| --- | --- | --- |
| 用户类型 | 外模式 | 权限 |
| 管理员 | adminindex.jsp  admindetails.jsp  cart.jsp  cartlist.jsp  count.jsp  creatitem.jsp  modifyitem.jsp | 增改查items  增删改查cartitems  增删查carts |
| 顾客 | index.jsp  details.jsp  cart.jsp  cartlist.jsp | 查items  增删改查cartitems  增删查carts |

# 存储过程、触发器和函数

## 存储过程

### 5.1.1 说明

删除某项历史清单的时候要同时完成删除清单和清单项

### 5.1.2 数据库端代码

CREAT proc [dbo].[deletecart]

@id int

as

begin

delete from cartitems where cartid=@id

delete from carts where id=@id

end

### 5.1.3 服务器端代码

conn = DBHelper.*getConnection*();

String sql = "{call deletecart(?)}";

out.write(sql);

stmt = conn.prepareCall(sql);

stmt.setInt(1,id);

stmt.execute();

## 触发器

### 5.2.1 说明

当下单以后，相应减少items表商品的库存(number)

### 5.2.2 触发器代码

CREAT trigger [dbo].[orderinsert1]

　　on [dbo].[cartitems]

　　after insert

　　as

　　update items set number=items.number-inserted.number

　　from items,inserted

　　where

　　items.id=inserted.itemid

## 函数

### 5.3.1 说明

标量值函数func()，查找id最大的cart，用来填入cartitem的cartid

### 5.3.2 数据库端代码

CREAT FUNCTION [dbo].[func]()

RETURNS int

AS

BEGIN

declare @avg1 int

select @avg1=max(id) fro m carts

return @avg1

END

### 5.3.3 服务器端代码

String sql = "insert into carts (date,money) values('"+date+"','"+money+"')";

out.write(sql);

stmt = conn.prepareStatement(sql);

stmt.execute();

sql = "select id as maxid from carts where id=dbo.func()";

# 主要技术论述

系统采用了Tomcat+JSP+SQL Server的架构，非常适合web应用的开发学习。技术要点有环境搭建、jar包引用，JSP脚本、HTML语言、JTSL标签、Jfreechart制图、数据库存取等。

# 运行实例



**图7-1. 管理员首页（商品列表）**

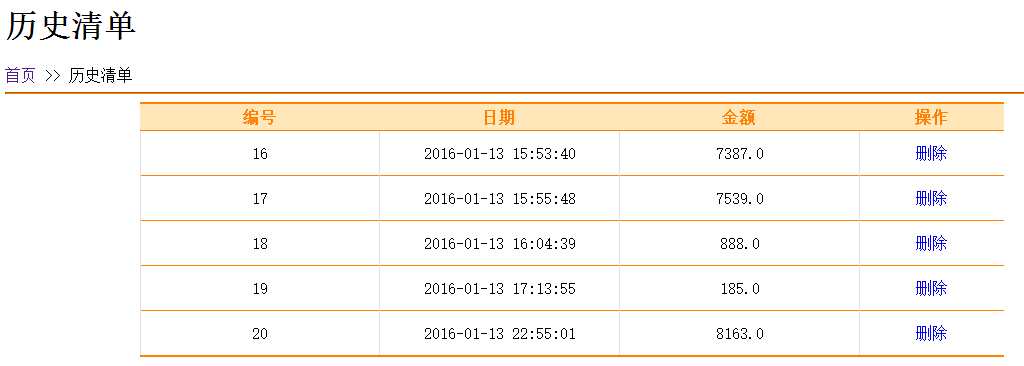
**图7-2. 顾客首页（商品列表）**



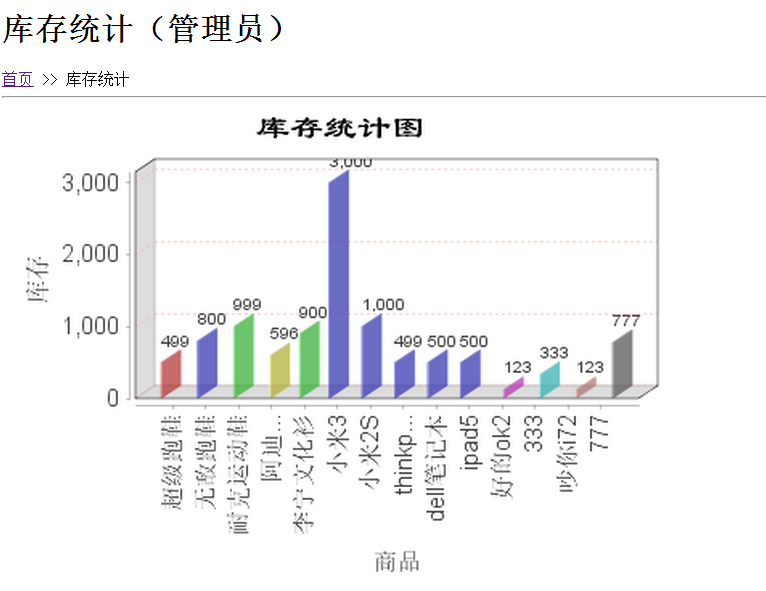
**图7-3. 添加到购物车**



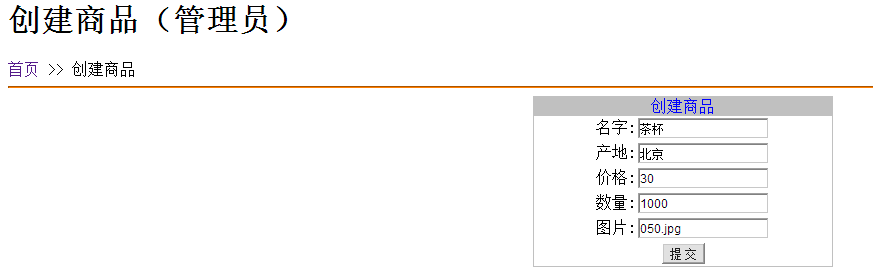
**图7-4. 购物清单**



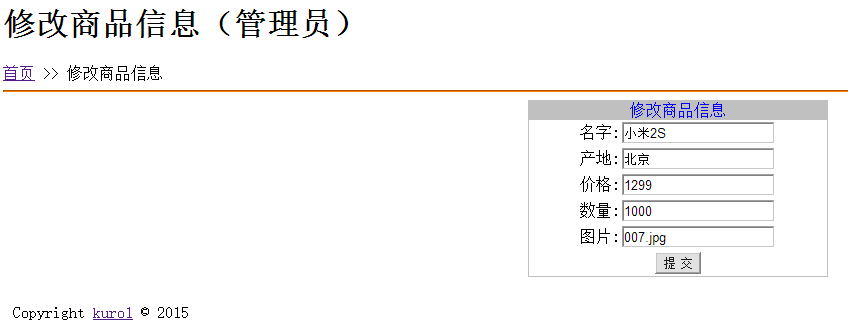
**图7-5. 历史清单**



**图7-6. 库存统计**

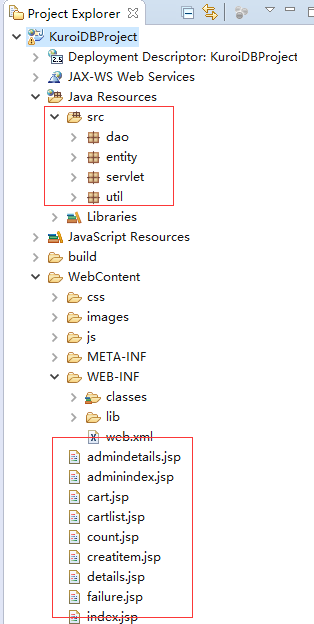


**图7-7. 创建商品**



**图7-7. 修改商品信息**

# 源程序简要说明



**图8-1. 源程序工程结构**

源程序是Eclipse的JSP应用结构，采用MVC架构，视图部分为.jsp，控制器部分为servlet，模型部分为model，util和dao提供数据库的连接和帮助。

# 收获和体会

这是我首次完成，并且是一个人完成的JSP应用程序，我从0基础开始学会了JSP、HTML，并且完成了相对完整了Tomcat+JSP+SQL Server的应用（虽然还比较简陋）。值得注意的是，我在开发过程中尽力保持MVC的视图、控制器、模型分离，虽然因此加大了工程量，但也是非常有意义的事。本学期学习完数据库后，我才真正体会到了应用、工程、项目的整体，不管是前端、后端，服务器、数据库，我都有了直观的感受以及操作、实现的经验，我相信这会是不错的经历。