



燕山大学
YANSHAN UNIVERSITY

课程设计报告

题目： “刺激精英”游戏信息管理系统

学院（系）： 信息科学与工程学院

年级专业： 18 级软件工程

小组名称： 青春有我

小组成员： 乔翱 白阜玉

杨捷 张璇

赵晓晴

校内指导教师： 梁顺攀

校内教师职称： 副教授

企业指导教师： 宋桥白

企业教师职称： 高级工程师

小组成绩					
项目 汇报效果	项目展示效果（45 分）		工具使 用情况	报告质量和 答辩效果	小组得分
	项目完成质 量	创新意识			
（10 分）	（40 分）	（5 分）	（10 分）	（5 分）	

学生信息		组内成绩		小组得分	个人得分
学号	姓名	组内贡献 （25 分）	合作意识 （5 分）		
201811040809	乔翱	25	5		
201811040783	白阜玉	24	5		
201811040786	张璇	23	5		
201811040800	杨捷	23	5		
201811040792	赵晓晴	22	5		

排序	学号	姓名	组内贡献和合作意识说明
1	201811040809	乔翱	负责任务的分配，代码、报告的修改整合，后台管理员模块的编写，很好的组织小组，合作意识良好
2	201811040783	白阜玉	负责注册登录功能、参加活动功能的编写和前端界面 UI 设计，报告以及 PPT 的部分编写。合作意识良好
3	201811040786	张璇	负责仓库模块的编写和前端界面的 UI 设计，报告以及 PPT 的部分编写。合作意识良好
4	201811040800	杨捷	负责商城模块的编写和 UI 设计，报告以及 PPT 的部分编写。合作意识良好
5	201811040792	赵晓晴	负责活动模块的部分编写，报告以及 PPT 的部分编写。合作意识良好

燕山大学课程设计（论文）任务书

院（系）： 信息科学与工程学院

基层教学单位：软件工程系

小组编号	09	学 生姓名		专业（班 级）	软件 18-6 班
设计 题 目	基于 Javaweb 的“刺激精英”游戏信息管理系统设计与开发				
设计 技 术 参 数	<p>要求实现下列功能：</p> <ol style="list-style-type: none"> 1. 用户功能(用户的注册、登录、管理员登录)。 2. 个人信息：用户登录后，能够查看个人信息，对个人信息进行修改。 3. 商品购买：在用户查看商品信息后，对于心仪的商品可以点击购买，购买后的商品可以在用户的个人仓库中查看。 4. 活动列表展示：当进入首页后，可以查看到所有活动列表内容， 5. 参与活动：用户点击活动可以参与活动，参与活动会收到金币奖励。 6. 管理功能：管理员可以对用户信息、商品信息、活动信息进行管理。 				
设计 要 求	<ol style="list-style-type: none"> 1. 和其他组员的子模块能组成一个完整的系统； 2. 整个网络课程只能使用一个数据库； 3. 文件夹的结构要合理、清晰； 4. 文档清晰、正确。 				
工 作 量	<ol style="list-style-type: none"> 1. 小组合作完成一个 Web 系统的设计、开发、部署、测试。 2. 每个组员独立设计与实现 Web 系统的一个较大功能模块/子系统。 3. 每组独立撰写课程设计说明书。 4. 保证每天 8 学时的出勤时间。 				
工 作 计 划	<p>第 1 天：熟悉开发环境；所需资料收集；需求分析；总体结构设计；学习环境设计；支持工具设计。</p> <p>第 2-3 天：系统详细设计；编写代码实现各种功能。</p> <p>第 4-5 天：调试，完善、优化系统；撰写课程设计说明书和总结；答辩。</p>				
参 考 资 料	<ol style="list-style-type: none"> 1. 详细书目参照课程设计指导书列举的参考书目。 2. 网上的精品课网站、网络课程网站。 				
指导教师签 字			基层教学单位 主任签字		

2020 年 6 月 24 日

燕山大学课程设计评审意见表

指导教师评语：

成绩：_____

指导教师：_____

年 月 日

答辩小组评语：

成绩：_____

评阅人：_____

年 月 日

课程设计总成绩：

答辩小组成员签字：

年 月 日

摘 要

本项目是一个基于 Javaweb 设计的“刺激精英”游戏信息管理系统。整体 web 应用的设计采取的是 MVC 的设计模式。项目的功能分为：用户注册、登录功能、购买商品、查看仓库商品、参加活动以及后台管理员对用户、活动、以及商品的管理。用户首先进行注册，完后登录，可以进入到项目的主界面，在主界面用户可以参加活动赚取游戏币，用户还可以在商城中购买商品，购买的商品可以在仓库中进行查看，管理员可以对用户、商品、活动的信息进行管理。

本 web 项目的前端界面采用 bootstrap 框架进行编写，后端采用 Servlet 以及 JSP 进行编写，数据库采用 MYSQL 数据库，整体的项目采取 MVC 的设计模式来进行设计，整个 web 项目结构简洁、清楚。

关键词：Javaweb；bootstrap；Servlet；JSP；游戏信息管理系统。

目 录

摘 要.....	i
1 绪 论.....	1
1.1 课程目标.....	2
1.2 课程内容.....	2
1.3 课程教学模式.....	2
1.4 课程的使用对象.....	2
2 需求分析.....	3
2.1 用户分析.....	3
2.1.1 普通用户.....	3
2.1.2 管理员.....	3
2.2 教学大纲分析.....	3
2.3 教学目标分析.....	3
2.3.1 课程总体目标.....	3
2.3.2 单元学习目标.....	3
2.4 教学内容分析.....	4
2.5 运行环境分析.....	4
2.5.1 硬件环境.....	4
2.5.2 软件环境.....	5
3 数据库设计.....	6
3.1 数据库的 ER 图.....	6
3.2 数据库中各个字段的设计.....	6
3.2.1 用户信息表 User.....	6
3.2.2 商品表 commodity.....	7
3.2.3 物品表 goods.....	7
3.2.4 活动表 activity.....	8
3.2.5 购买商品表 purchase_commodity.....	8
3.2.6 参加活动表 participate_activities.....	8
3.3 数据库中各个表的建立.....	9
3.3.1 user 表.....	9
3.3.2 goods 表.....	10
3.3.3 activity 表.....	10
3.3.4 commodity 表.....	11

3.3.5	participate_activities 表.....	12
3.3.6	purchase_commodity 表.....	12
4	用户模块设计.....	14
4.1	登录页面设计.....	14
4.2	注册页面设计.....	18
4.3	注册用户功能的实现.....	23
4.4	展示用户信息界面的设计.....	24
4.5	展示用户信息功能的实现.....	28
4.6	登录功能的实现.....	29
4.7	修改用户信息界面设计.....	31
4.8	修改用户信息功能的实现.....	34
5	管理员模块设计.....	36
5.1	管理员首页界面设计.....	36
5.2	用户信息管理界面设计.....	38
5.3	后台活动信息管理界面设计.....	39
5.4	后台商品信息管理界面设计.....	41
5.5	后台用户信息管理功能的实现.....	43
5.6	后台活动信息管理功能的实现.....	45
5.7	后台商品信息管理功能的实现.....	50
6	活动模块设计.....	56
6.1	活动界面设计.....	56
6.2	查询所有活动功能的实现.....	61
6.3	参加活动功能的实现.....	62
7	仓库模块设计.....	64
7.1	仓库界面设计.....	64
7.2	展示仓库所有物品功能的实现.....	68
7.3	丢弃仓库物品功能的实现.....	69
8	商城模块设计.....	71
8.1	商城界面设计.....	71
8.2	展示商城所有物品功能的实现.....	75
8.3	购买商品功能的实现.....	76
9	集成与测试.....	78
9.1	项目最终效果截图.....	78

总 结.....	82
参考文献资料.....	83

1 绪 论

如今生活快速，人们与好友休闲娱乐放松心情的时间越来越好，而刺激精英这个游戏媒体不仅让人们能够与生活中的好友又更多的交流，更能够认识新的朋友。而且刺激精英有利于锻炼人们的反应能力，团结能力，与他人沟通的能力，更甚之能够找到幸福。而对于游戏的一些活动以及游戏的武器等有时不能全部了解，而本项目能满足用户的以上需求。

本项目是基于 Javaweb 的“刺激精英”游戏信息管理系统，用户通过注册、登录，可以进入本网站，在本网站中用户可以查看自己的相关信息，包括用户的角色姓名、用户拥有的游戏币等信息，用户也可以修改相关信息，例如修改密码等，用户还可以在商城中购买商品，在首页参加活动赚取游戏币。

开发本项目的目的就是方便游戏用户对于自己的游戏信息进行管理，并且方便用户购买商品，参加活动。

本项目具有很大的应用前景，国内外目前也有相关平台具有同样的功能，但是本项目不同的是，功能比较全面。用户可以购买商品、参与活动等操作，本项目的特色也比较突出，前端界面也是精心设计的，提升了用户友好性。

在本次项目的完成过程中，组内成员团结合作进行项目的开发，由于开发时间有限，部分功能还有待完善，但是在这个过程中，每个人都竭尽全力，通过本次项目提高了自己的能力，每个人都收货颇丰。

本项目主要研究了利用 Javaweb 技术、采取 MVC 的设计模式进行小型的网站的开发，在前端还采用了 bootstrap 框架进行了前端界面的 UI 设计，通过前端后端的结合进行了一个小型系统的开发。

1.1 课程目标

应用 BootStrap 框架实现原型设计、并且应用 JDBC 实现数据库操作；以及应用 MVC 设计思想实现项目架构搭建，通过与 Java web 技术的思想结合完成项目的实现。

1.2 课程内容

时间	课程设计内容
第 1 周第 1 天 (6.24)	1. 确定项目方向 2. 确定项目功能 3. 进行需求分析
第 1 周第 2 天 (6.25)	1. 进行项目数据库的设计 2. 进行小组分工 3. 进行各个页面的 UI 设计
第 1 周第 3 天 (6.26)	1. 进行后端代码的编写 2. 各个组员分别负责一个模块后端代码的编写
第 1 周第 4 天 (6.27)	1. 进行前后端代码的整合 2. 进行功能的完善 3. 优化前端的界面
第 1 周第 5 天 (6.28)	1. 提交项目提交物 2. 项目评审答辩 3. 毕营

1.3 课程教学模式

同学之间进行自由结组，有利用组内部沟通，给出 MVC 框架，让同学利用框架进行项目的构建。在实际项目构建过程中通过对项目整体目标、分解目标及阶段里程碑的确定，提升参训人员的效率、提升参训人员的抗压力，不断提升参训人员灵活运用所学的各种专业知识及技术技能解决复杂综合实际问题的能力。

1.4 课程的使用对象

18 级软件工程学生

2 需求分析

2.1 用户分析

2.1.1 普通用户

普通用户即为查看者，在注册后，拥有账号后才可以进行登录。登录后，用户可以修改自己的个人信息，能够查看自己仓库内的物品，在商城中购买商品，并且可以在首页进行查看活动、参加活动，

2.1.2 管理员

管理者即为信息管理者，在登录之后，可以对用户信息、商城中的商品信息、活动信息进行管理，包括对用户的删除，对商品的新增、修改、删除，以及对活动的新增修改、删除。

2.2 教学大纲分析

同学之间进行自由结组，有利用组内部沟通，给出 MVC 框架，让同学利用框架进行项目的构建。在实际项目构建过程中通过对项目整体目标、分解目标及阶段里程碑的确定，提升参训人员的效率、提升参训人员的抗压力，不断提升参训人员灵活运用所学的各种专业知识及技术技能解决复杂综合实际问题的能力。

2.3 教学目标分析

应用 bootstrap 框架进行前端的搭建，运用 JDBC 技术实现对数据库的操作，并且可以熟练应用 servlet、Jsp 技术，应用 MVC 设计思想实现项目架构搭建，通过与 Java web 技术的思想结合完成项目的实现。

2.3.1 课程总体目标

基于 Javaweb 利用 eclipse+tomcat+MySQL 环境开发出一个功能完善的程序。

2.3.2 单元学习目标

第一天：

1. 讨论分析项目的实现的功能。
2. 数据库的设计以及实现。

第二天：

1. 应用 JDBC 实现数据交互。
2. 利用 Bootstrap 快速搭建“刺激精英”原型。

第三天：

1. 各个功能后端代码的编写。
2. 后端代码的整合。

第四天：

1. 前后端代码的整合。
2. 功能的完善，界面的美化。

第五天：

1. 提交项目提交物。
2. 项目评审答辩。
3. 毕营。

2.4 教学内容分析

在 WebStorm 中搭建好 bootstrap 和 jQuery，学会利用 bootstrap 和 jQuery 进行前端页面设计开发。

在 spring+tomcat+MySQL 环境下创建 MVC 项目，再利用 bootstrap 和 jQuery 进行前端页面设计开发完成一个完整的程序设计。

2.5 运行环境分析

2.5.1 硬件环境

电脑型号 戴尔 G7 7588 笔记本电脑（扫描时间：2020 年 06 月 26 日）

操作系统 Windows 10 64 位（ DirectX 12 ）

处理器 英特尔 Core i7-8750H @ 2.20GHz 六核

主板 戴尔 075F7T（ HM370 芯片组 ）

内存 8 GB（ 海力士 DDR4 2666MHz ）

主硬盘 英特尔 SSDSCKKF128G8 SATA 128GB（ 128 GB / 固态硬盘 ）

显卡 Nvidia GeForce GTX 1050 Ti（ 4 GB ）

显示器 LG LGD053F（ 15.5 英寸 ）

声卡 瑞昱 @ 英特尔 High Definition Audio 控制器

网卡 英特尔 Wireless-AC 9560

2.5.2 软件环境

Jdk 1.8.0_91 windows-x64

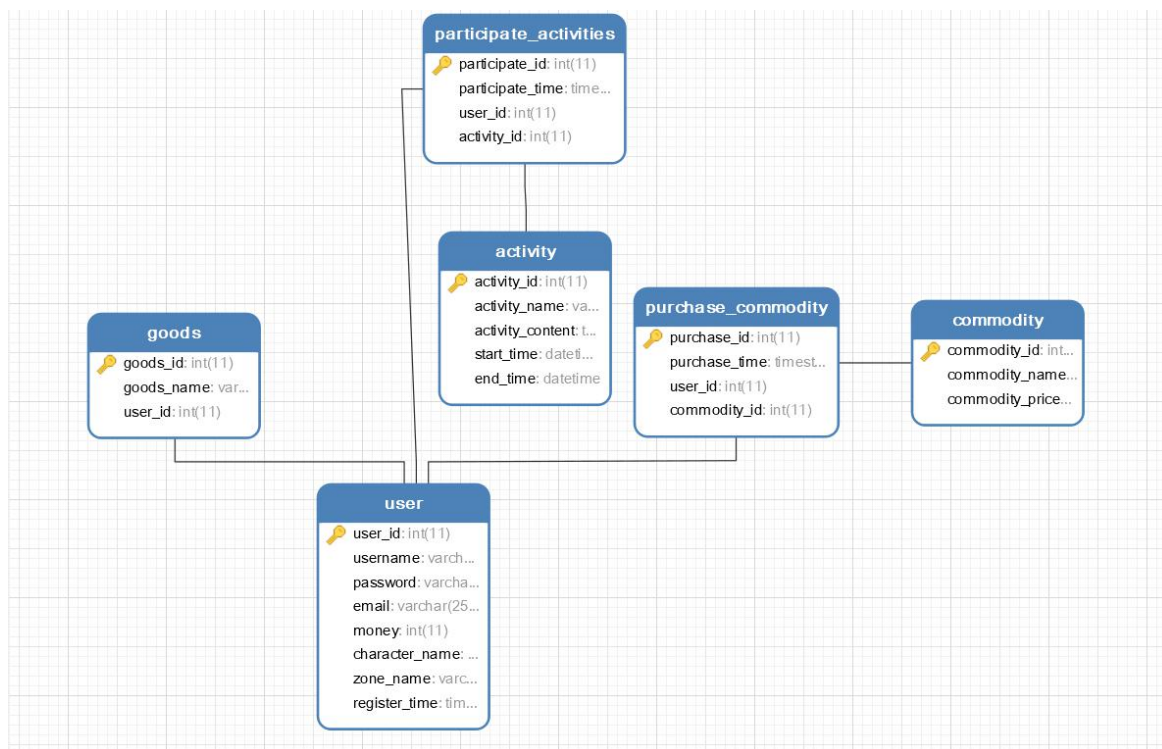
Eclipse 2019-12 windows-x64

Tomcat 9.0.17 windows-x64

Mysql 5.8 windows-x64

3 数据库设计

3.1 数据库的 ER 图



3.2 数据库中各个字段的设计

3.2.1 用户信息表 User

序号	列名称	列说明	类型	长度	属性	备注
1	user_id	用户编号	int	11	非空	主键 自增 1
2	username	用户姓名	varchar	255	非空	
3	password	密码	Varchar	255	非空	
4	email	用户邮箱	varchar	255	非空	

5	money	用户的游戏币	int	11	非空	
6	character_name	游戏角色名	varchar	255	非空	
7	zone_name	用户所在大区的名字	varchar	255	非空	
8	register_time	用户注册的时间	timestamp	0	非空	默认当前时间

3.2.2 商品表 commodity

序号	列名称	列说明	类型	长度	属性	备注
1	commodity_id	商品编号	int	11	非空	主键, 自增 1
2	commodity_name	商品名称	varchar	255	非空	
3	commodity_price	商品价格	int	11	非空	

3.2.3 物品表 goods

序号	列名称	列说明	类型	长度	属性	备注
1	goods_id	物品编号	int	11	非空	主键, 自增 1
2	goods_name	物品名称	varchar	255	非空	
3	user_id	物品拥有者编号	int	11	非空	

3.2.4 活动表 activity

序号	列名称	列说明	类型	长度	属性	备注
1	activity_id	活动编号	int	11	非空	主键， 自增 1
2	activity_name	活动名称	varchar	255	非空	
3	activity_content	活动内容	text	0	非空	
4	start_time	活动 开始时间	datetime	0	非空	
5	end_time	活动 结束时间	datetime	0	非空	

3.2.5 购买商品表 purchase_commodity

序号	列名称	列说明	类型	长度	属性	备注
1	purchase_id	购买编号	int	11	非空	主键， 自增 1
2	purchase_time	购买时间	timestamp	0	非空	默认当前时间
3	user_id	购买该商品用户编号	int	11	非空	外键
4	commodity_id	购买的商品编号	int	11	非空	外键

3.2.6 参加活动表 participate_activities

序号	列名称	列说明	类型	长度	属性	备注
1	participate_id	参与活动编号	int	11	非空	主键， 自增 1

2	participate_time	参与活动时间	timestamp	0	非空	默认当前时间
3	user_id	参与活动 用户编号	int	11	非空	外键
4	activity_id	被参与的 活动编号	int	11	非空	外键

3.3 数据库中各个表的建立

3.3.1 user 表

建表语句:

```

-----
-- Table structure for user
-----
DROP TABLE IF EXISTS `user`;
CREATE TABLE `user` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci
  NULL DEFAULT NULL,
  `password` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci
  NULL DEFAULT NULL,
  `email` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci NULL
  DEFAULT NULL,
  `money` double NULL DEFAULT NULL,
  `character_name` varchar(255) CHARACTER SET utf8 COLLATE
  utf8_general_ci NULL DEFAULT NULL,
  `zone_name` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci
  NULL DEFAULT NULL,
  `register_time` timestamp(0) NULL DEFAULT CURRENT_TIMESTAMP(0),
  PRIMARY KEY (`user_id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
SET FOREIGN_KEY_CHECKS = 1;

```

对应截图:

user_id	username	password	email	money	character_name	zone_name	register_time
6	笨笨	123	14840756	980	明明	2	2020-06-27 20:06

3.3.2 goods 表

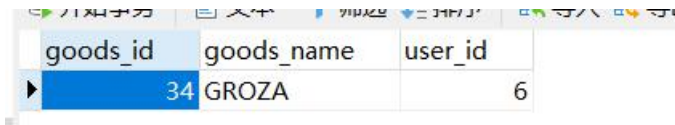
建表语句：

```

-----
-- Table structure for goods
-----
DROP TABLE IF EXISTS `goods`;
CREATE TABLE `goods` (
  `goods_id` int(11) NOT NULL AUTO_INCREMENT,
  `goods_name` varchar(255) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL,
  `user_id` int(11) NOT NULL,
  PRIMARY KEY (`goods_id`) USING BTREE,
  INDEX `user_id` (`user_id`) USING BTREE,
  CONSTRAINT `user_id` FOREIGN KEY (`user_id`) REFERENCES
`user` (`user_id`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;

```

对应截图：



goods_id	goods_name	user_id
34	GROZA	6

3.3.3 activity 表

建表语句：

```

-----
-- Table structure for activity
-----
DROP TABLE IF EXISTS `activity`;
CREATE TABLE `activity` (
  `activity_id` int(11) NOT NULL AUTO_INCREMENT,
  `activity_name` varchar(255) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL,
  `activity_content` text CHARACTER SET utf8 COLLATE utf8_general_ci
NOT NULL,
  `start_time` datetime(0) NOT NULL,
  `end_time` datetime(0) NOT NULL,

```

```
PRIMARY KEY (`activity_id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

对应截图：

activity_id	activity_name	activity_content	start_time	end_time
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

3.3.4 commodity 表

建表语句：

```
-- -----
-- Table structure for commodity
-- -----
DROP TABLE IF EXISTS `commodity`;
CREATE TABLE `commodity` (
  `commodity_id` int(11) NOT NULL AUTO_INCREMENT,
  `commodity_name` varchar(255) CHARACTER SET utf8 COLLATE
utf8_general_ci NULL DEFAULT NULL,
  `commodity_price` double NULL DEFAULT NULL,
  PRIMARY KEY (`commodity_id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

对应截图：

commodity_id	commodity_name	commodity_price
1	AKM	20
2	Kar98K	20
3	GROZA	20
4	M416	20
5	幻影战神	30
6	旷野奇兵	30
7	万圣娃娃	30
8	快乐主宰	30

3.3.5 participate_activities 表

建表语句：

```

-----
-- Table structure for participate_activities
-----
DROP TABLE IF EXISTS `participate_activities`;
CREATE TABLE `participate_activities` (
  `participate_id` int(11) NOT NULL AUTO_INCREMENT,
  `participate_time` datetime(0) NOT NULL,
  `user_id` int(11) NOT NULL,
  `activity_id` int(11) NOT NULL,
  PRIMARY KEY (`participate_id`) USING BTREE,
  INDEX `activity_id` (`activity_id`) USING BTREE,
  INDEX `user_id0` (`user_id`) USING BTREE,
  CONSTRAINT `activity_id` FOREIGN KEY (`activity_id`)
REFERENCES `activity` (`activity_id`) ON DELETE RESTRICT ON UPDATE
RESTRICT,
  CONSTRAINT `user_id0` FOREIGN KEY (`user_id`) REFERENCES
`user` (`user_id`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;

```

对应截图：

participate_id	participate_time	user_id	activity_id
(N/A)	(N/A)	(N/A)	(N/A)

3.3.6 purchase_commodity 表

建表语句：

```

-----
-- Table structure for purchase_commodity
-----
DROP TABLE IF EXISTS `purchase_commodity`;
CREATE TABLE `purchase_commodity` (
  `purchase_id` int(11) NOT NULL AUTO_INCREMENT,
  `purchase_time` datetime(0) NOT NULL,
  `user_id` int(11) NOT NULL,
  `commodity_id` int(11) NOT NULL,

```

```
PRIMARY KEY (`purchase_id`) USING BTREE,  
INDEX `commodity_id` (`commodity_id`) USING BTREE,  
INDEX `user_id1` (`user_id`) USING BTREE,  
CONSTRAINT `commodity_id` FOREIGN KEY (`commodity_id`)  
REFERENCES `commodity` (`commodity_id`) ON DELETE RESTRICT ON UPDATE  
RESTRICT,  
CONSTRAINT `user_id1` FOREIGN KEY (`user_id`) REFERENCES `user`  
(`user_id`) ON DELETE RESTRICT ON UPDATE RESTRICT  
) ENGINE = InnoDB AUTO_INCREMENT = 1 CHARACTER SET = utf8  
COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

对应截图：

participate_id	participate_time	user_id	activity_id
(N/A)	(N/A)	(N/A)	(N/A)

4 用户模块设计

4.1 登录页面设计

对于用户以及管理员的登录界面，采用 bootstrap 框架进行开发，其中主要元素是输入用户名和密码的文本框，对于文本框输入的内容都在前端利用 JavaScript 进行了校验，只有输入不为空并且符合输入规则才进行提交到后端。

对应的核心代码：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>普通用户登录界面</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
        .headForm{
            margin:20px ;
        }
        .active a{
            color: red;
        }
        section{
            background-image: url("imgs/bg.jpg");
            width: 100%;
            height: 812px;
        }
        .forMain{
            background: #fff;
            margin-top: 50px;
            border-radius: 10px;
            opacity:0.7;
        }
        h2{
            font-size: 18px;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="headForm">
                    <h2>普通用户登录</h2>
                    <div class="form-group">
                        <input type="text" class="form-control" placeholder="请输入用户名">
                    </div>
                    <div class="form-group">
                        <input type="password" class="form-control" placeholder="请输入密码">
                    </div>
                    <div class="form-group">
                        <input type="button" value="登录" class="btn btn-primary">
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
</html>
```



```

        footer{
            background-color:#cccccc;
            padding: 20px;
        }
    </style>
</head>
<body>
<div class="container">
    <header class="container">
        <nav class="nav navbar-inverse" style="padding-right: 20px">
            <div class="navbar-header">
                <button class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a href="#" class="navbar-brand"></a>
            </div>
            <div class="collapse navbar-collapse navbar-left">
                <ul class="nav navbar-nav">
                    <li><a href="index.jsp">首页</a></li>
                    <li><a href="register.jsp">游戏注册</a></li>
                    <li><a href="#" style="color: gold">游戏登录</a></li>
                </ul>
            </div>
            <form class="navbar-form navbar-right" role="search">
                <div class="form-group">
                    <input type="text" class="form-control"
placeholder="Search">
                </div>
                <button type="submit" class="btn btn-primary">搜索</button>
            </form>
        </nav>
    </header>
    <section class="container">
        <div class="row">
            <div class="col-md-4 col-md-offset-4 forMain">

```

```

<h2>普通用户</h2>
<form class="form-horizontal" action="LoginUserServlet"
method="post">
    <div class="form-group">
        <label for="username" class="col-md-3 control-label">
用户名</label>
        <div class="col-md-9">
            <input type="text" class="form-control"
id="username" name="username" placeholder="请输入用户名">
        </div>
    </div>
    <div class="form-group">
        <label for="password" class="col-md-3 control-label">
密码</label>
        <div class="col-md-9">
            <input type="password" class="form-control"
id="password" name="password" placeholder="请输入密码">
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-9 col-md-offset-3">
            <div class="checkbox">
                <label>
                    <input type="checkbox"> 记住密码
                </label>
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-4 col-md-offset-3">
            <button type="submit" class="btn btn-default">登
录</button>
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-4 col-md-offset-3">
            <p><a href="register.jsp">没有账号，注册
</a></p>
        </div>
    </div>

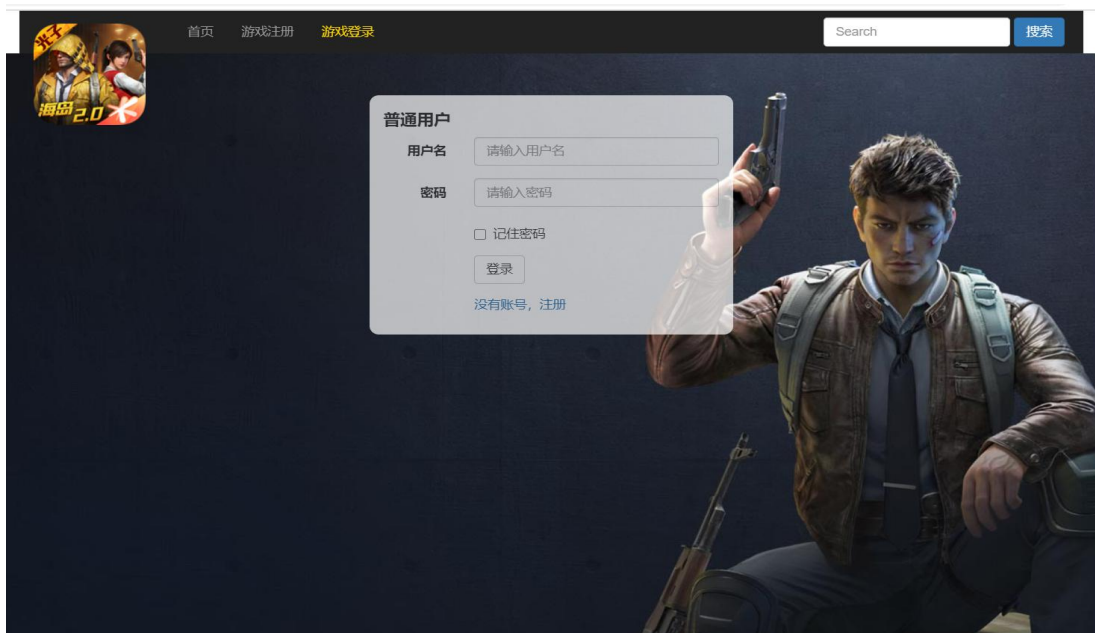
```

```

        </div>
    </form>
</div>
</div>
</section>
<footer class="container text-center">
    <h4>抵制不良游戏 拒绝垃圾游戏 注意自我保护 谨防受骗上当</h4>
    <h4>适度游戏益脑 沉迷游戏伤身 合理安排时间 享受健康生活</h4>
</footer>
</div>
<script src="javascript/jquery-1.12.4.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</body>
<script>
    var errori ='<%=request.getParameter("error")%>';
    if(errori=='yes'){
        alert("用户名或密码错误，请重试!");
    }
    var zci ='<%=request.getParameter("zc")%>';
    if(zci=='yes'){
        alert("注册成功，请登录");
    }
</script>
</html>

```

登录界面截图：



4.2 注册页面设计

对于注册页面采用 bootstrap 框架进行开发，并且注册的时候对于用户输入的信息进行校验，不符合输入规则的会有警示框进行警告，只有注册输入的所有信息都符合规则才能够进行表单的提交。

对应的核心代码：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>普通用户注册界面</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
        .headForm{
            margin:20px ;
        }
        .active a{
            color: red;
        }
        section{
            background-image: url("imgs/bg.jpg");
            width: 100%;
            height: 812px;
        }
        .forMain{
            background: #fff;
            margin-top: 50px;
            border-radius: 10px;
            opacity:0.7;
        }
        h2{
            font-size: 18px;
            font-weight: bold;
        }
        footer{
            background-color:#cccccc;
            padding: 20px;
```

```
    }  
    .main{  
        margin-top:3.5%;  
    }  
</style>  
</head>  
<script>  
    function oc1(){  
        var user1=document.getElementById("username");  
        if(user1.value.length==0){  
            return false;  
        }  
        else return true;  
    }  
    function oc2(){  
        var password=document.getElementById("password");  
        if(password.value.length==0){  
            return false;  
        }  
        else {  
            return true;  
        }  
    }  
    function oc3(){  
        var email=document.getElementById("email");  
        if(email.length==0){  
            return false;  
        }  
        else return true;  
    }  
    function oc4(){  
        var character_name=document.getElementById("character_name");  
        if(character_name.length==0){  
            return false ;  
        }  
        else return true;  
    }  
    function oc5(){  
        if(!oc1())  
        {  
            alert("用户名不能为空");  
        }  
    }  
}
```

```
        return false;
    }
    else if(!oc2())
    {
        alert("密码不能为空");
        return false;
    }
    else if(!oc3())
    {
        alert("邮箱不能为空");
        return false;
    }
    else if(!oc4())
    {
        alert("角色名不能为空");
        return false;
    }
    else return true;
}
</script>
<body>
<div class="container">
    <!-- 顶部导航条-->
    <header class="container">
        <nav class="nav navbar-inverse" style="padding-right: 20px">
            <div class="navbar-header">
                <button class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a href="#" class="navbar-brand"></a>
            </div>
            <div class="collapse navbar-collapse navbar-left">
                <ul class="nav navbar-nav">
                    <li><a href="#">首页</a></li>
                    <li><a href="#" style="color: gold">游戏注册</a></li>
```

```

        <li><a href="#">游戏登录</a></li>
    </ul>
</div>
<!--      在导航条中添加表单-->
<form class="navbar-form navbar-right" role="search">
    <div class="form-group">
        <input type="text" class="form-control"
placeholder="Search">
    </div>
    <button type="submit" class="btn btn-primary">搜索</button>
</form>
</nav>
</header>
<!--      中间注册开始-->
<section class="container">
    <div class="row">
        <div class="col-md-4 col-md-offset-4 forMain">
            <h2>普通用户</h2>
            <form class="form-horizontal" action="AddUserServlet"
method="post" onsubmit="return oc5()">
                <!--private int a=0,b=0,c=0,d=0; -->
                <div class="form-group">
                    <label for="username" class="col-md-3
control-label">用户名</label>
                    <div class="col-md-9">
                        <input type="text" class="form-control"
id="username" name="username" placeholder="请输入用户名" >
                    </div>
                </div>

                <div class="form-group">
                    <label class="col-md-3 control-label">密码</label>
                    <div class="col-md-9">
                        <input type="password" class="form-control"
id="password" name="password" placeholder="请输入密码" >
                    </div>
                </div>

                <div class="form-group">
                    <label class="col-md-3 control-label">邮箱</label>
                    <div class="col-md-9">

```

```

        <input type="email" class="form-control"
id="email" name="email" placeholder="请输入邮箱" >
    </div>
</div>
<div class="form-group">
    <label class="col-md-3 control-label">角色名
</label>

    <div class="col-md-9">
        <input type="text" class="form-control"
name="character_name" id="character_name" placeholder="请输入角色名" >
    </div>
</div>

<div class="form-group">
    <label class="col-md-3 control-label">大区名
</label>

    <div class="col-md-6 col-md-offset-1" >
        <select name="zone_name" class="form-group
main ">

            <option class="form-control" value="1"
name="zone_name" selected>1</option>
            <option class="form-control" value="2"
name="zone_name">2</option>
            <option class="form-control" value="3"
name="zone_name">3</option>

        </select>
    </div>
</div>
<div class="form-group">
    <div class="col-md-4 col-md-offset-3">
        <button type="submit" class="btn btn-default" >
注册</button>

    </div>
</div>
<div class="form-group">
    <div class="col-md-4 col-md-offset-3">
        <p><a href="Login.jsp">已有账号，登录</a></p>
    </div>
</div>
</form>
</div>

```

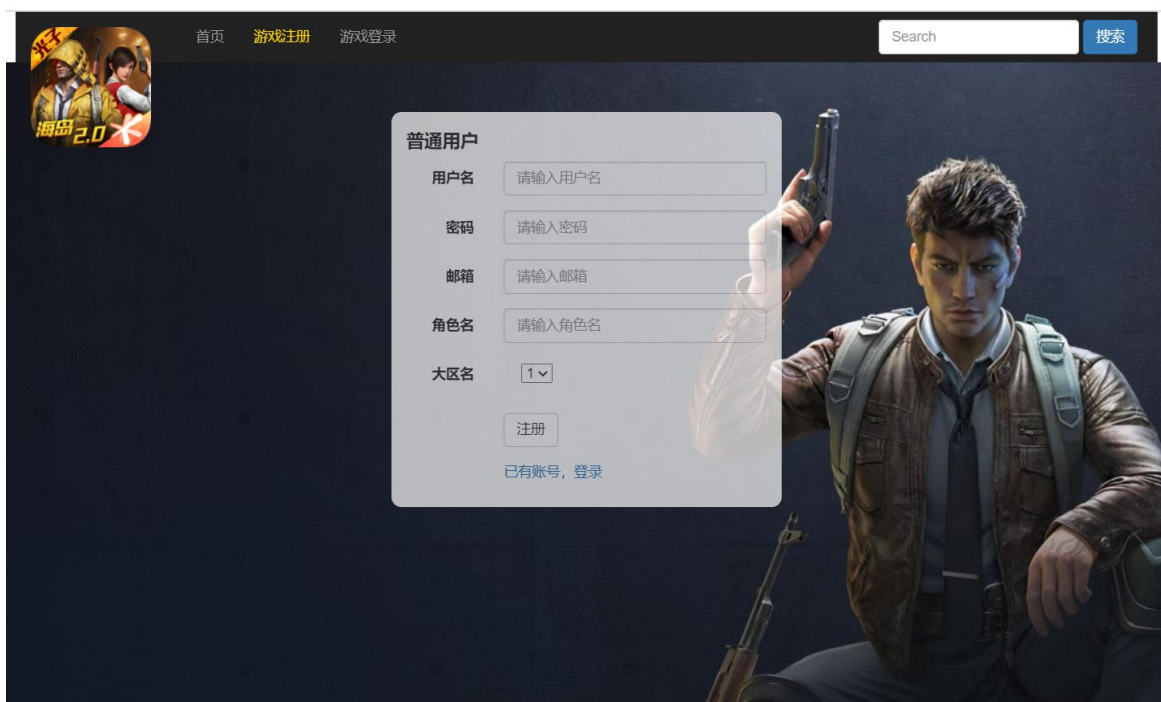


```

        </div>
    </section>
    <!-- 中间登录结束-->
    <footer class="container text-center">
        <h4>抵制不良游戏 拒绝垃圾游戏 注意自我保护 谨防受骗上当</h4>
        <h4>适度游戏益脑 沉迷游戏伤身 合理安排时间 享受健康生活</h4>
    </footer>
</div>
<script src="js/jquery-1.7.2.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</body>
</html>

```

注册页面截图：



4.3 注册用户功能的实现

对于用户注册的功能的实现，只需要调用业务逻辑层的添加用户的方法就可以实现，在 user 表里面添加一条信息，进行用户的新增，用户注册的功能就完成了。

对于的核心代码：

```

package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;

```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.User;
import com.etc.service.UserService;
import com.etc.service.impl.UserServiceimpl;
@WebServlet("/AddUserServlet")
public class AddUserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        String username=request.getParameter("username");
        String password=request.getParameter("password");
        String email=request.getParameter("email");
        String character_name=request.getParameter("character_name");
        String zone_name=request.getParameter("zone_name");
        try {
            User user = new User(username, password, email,
character_name,zone_name);
            UserService userService=new UserServiceimpl();
            boolean result = userService.addUser(user) ;
            if(!result) {
                System.out.println("error");
                response.sendRedirect("error.jsp");
            } else {
                response.sendRedirect("login.jsp?zc=yes");
            }
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

4.4 展示用户信息界面的设计

对于用户信息界面的编写采用 bootstrap 框架进行开发，最上方有导航栏可以实现各个功能的跳转，在主体部分显示用户的所有信息，并且有修改信息按钮和退出登录按钮，可以实现页面的跳转。

对应的核心代码:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
import="com.etc.entity.User"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>个人信息</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
        @media (max-width: 768px) {
            form {
                display: none;
            }
        }

        * {
            margin: 0;
            padding: 0;
        }

        html, body {
            width: 100%;
            height: 100%;
        }
        .column{
            opacity:0.65;
            background:#0f0f0f right;
            margin-left: 35%;
            margin-right:20%;
            border-radius:10%;
        }
        .tx{
            border-radius: 30%;
            border:solid 5px;
            margin-top: 10%;
        }
        .bg {
```

```

        display: table;
        width: 100%;
        height: 100%;
        text-align: center;
        background: url(imgs/bg.jpg) no-repeat bottom center;
        background-size: cover;
    }
    section{
        background: url(imgs/ufbg.jpeg) no-repeat bottom center;
        background-size: cover;
        height:590px;
    }
    footer{
        background-color:#fff;
        padding: 20px;
        font-weight: bolder;
        color: black;
    }
    li{
        list-style: none;
    }
</style>
</head>
<body>
<div class="container bg">
<header class="container">
    <nav class="nav navbar-inverse" style="padding-right: 20px">
        <div class="navbar-header">
            <button class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a href="#" class="navbar-brand"></a>
        </div>
        <div class="collapse navbar-collapse navbar-left">
            <ul class="nav navbar-nav">

```

```

        <li><a href="index.jsp">首页</a></li>
        <li><a style="color: gold" href="#">个人中心</a></li>
        <li><a href="FindAllCommoditiesSevert">游戏商城
</a></li>

        <li><a href="FindGoodsAllByUser_Id">个人仓库
</a></li>

        <li><a href="#">社交区</a></li>
    </ul>
</div>

    <form class="navbar-form navbar-right" role="search">
    <div class="form-group">
        <input type="text" class="form-control"
placeholder="Search">
    </div>
    <button type="submit" class="btn btn-primary">搜索</button>
    </form>
</nav>
</header>

<section class="container" >
    <div class="col-lg-10 col-md-10 text-center">
        <!--顶部头像和游戏名称-->
        <div class="column" style="margin-top:30px">
            <div>
                
            </div>
            <ul class="list-group">
                <h2 style="color: white"><li>编
号:${sessionScope.user_id}</li></h2>
                <h2 style="color: white"><li>昵
称:${sessionScope.character_name}</li></h2>
                <h2 style="color: white"><li>点券:
${sessionScope.money}</li></h2>
                <h2 style="color: white"><li>所在区:
${sessionScope.zone_name}</li></h2>
                <h2 style="color: white"><li>注册时间:
${sessionScope.register_time}</li></h2>
            </ul>
            <a href="modify.jsp" style="color:white">修改信息</a>
            <a href="login.jsp" style="color:white">退出登录</a>

```

```
</div>
</div>
</section>
<footer class="container text-center">
    <h4>抵制不良游戏 拒绝垃圾游戏 注意自我保护 谨防受骗上当</h4>
    <h4>适度游戏益脑 沉迷游戏伤身 合理安排时间 享受健康生活</h4>
</footer>

</div>
<script src="js/jquery-1.7.2.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</body>
</html>
```

个人信息页面对应的截图：



抵制不良游戏 拒绝垃圾游戏 注意自我保护 谨防受骗上当
适度游戏益脑 沉迷游戏伤身 合理安排时间 享受健康生活

4.5 展示用户信息功能的实现

此功能只需要根据用户 id 在数据库中的 user 表中进行，并且把查询到的结果存到 request 域中，在前端页面取出用户信息进行显示。

对应的核心代码：

```
package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.User;
import com.etc.service.UserService;
import com.etc.service.impl.UserServiceimpl;
@WebServlet("/QueryUserByIdServlet")
public class QueryUserByIdServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        int user_id = Integer.parseInt(request.getParameter("user_id"));
        System.out.println(user_id);
        UserService service = new UserServiceimpl();
        User user = service.findUserById(user_id);
        System.out.println(user);
        request.setAttribute("user", user);
        request.getRequestDispatcher("userinfo.jsp").forward(request, response);
    }
}
```

4.6 登录功能的实现

当用户在登录界面输入用户名和密码后，在 `servlet` 需要对其进行校验，此功能需要调用业务逻辑层的登录的函数，此函数是通过用户名和密码在 `user` 表中进行查询，如果查到了，则返回 `true` 可以进入首页，没查到表示用户名或者密码输入错误，返回 `false`，返回登录界面。

对应的核心代码：

```
package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
```

```
import com.etc.entity.User;
import com.etc.service.UserService;
import com.etc.service.impl.UserServiceimpl;
@WebServlet("/LoginUserServlet")
public class LoginUserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    request.setCharacterEncoding("utf-8");
    String username=request.getParameter("username");
    String password=request.getParameter("password");
    try {
        UserService userService = new UserServiceimpl();
        User user = userService.loginUser(username, password);
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        if(user==null) {
            response.sendRedirect("login.jsp?error=yes");
        }else {
            request.getSession().setAttribute("user", user);
            request.getSession().setAttribute("user_id", user.getUser_id());
            request.getSession().setAttribute("username",username);
            request.getSession().setAttribute("password", password);
            request.getSession().setAttribute("email", user.getEmail());
            request.getSession().setAttribute("money", user.getMoney());
            request.getSession().setAttribute("character_name",
user.getCharacter_name());
            request.getSession().setAttribute("zone_name", user.getZone_name());
            request.getSession().setAttribute("register_time",
user.getRegister_time());
            System.out.println(user);
            response.sendRedirect("index.jsp");
        }
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
}
```


4.7 修改用户信息界面设计

修改用户信息的界面采用 bootstrap 框架进行开发，显示出用户的信息。

对应的核心代码：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>个人信息</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
        .me {
            background-image: url("imgs/modifyBG.jpg");
            width: auto;
            height: 670px;
        }
        .main2
        {
            margin-top:8%;
        }
        .select{
            margin-top:2%;
        }
        .main1 {
            background: #fff;
        }
        .forMain{
            background: #fff;
            opacity:0.5;
            border-radius:10%
        }
    </style>
</head>
<body>
<div class="container me" style="text-align: center">
<div class="row">
<div class="col-md-7 col-md-offset-2    forMain">
```

```

<form class="form-horizontal" action="UpdateUserServlet" method="post">
  <div class="main1">
    <div class="main2">
      <div class="form-group">
        <label for="user_id" class="col-md-3 control-label"><h5>编号:</h5></label><br>
        <div class="col-md-6">
          <input type="text" id="user_id" class="form-control" name="user_id" value="{sessionScope.user_id}" disabled>
        </div>
      </div>
      <div class="form-group">
        <label for="username" class="col-md-3 control-label"><h5>修改用户名为:</h5></label><br>
        <div class="col-md-6">
          <input type="text" id="username" class="form-control" name="username" value="{sessionScope.username}">
        </div>
      </div>
      <div class="form-group">
        <label for="password" class="col-md-3 control-label"><h5>修改密码为:</h5></label><br>
        <div class="col-md-6">
          <input type="text" id="password" class="form-control" name="password" value="{sessionScope.password}">
        </div>
      </div>
      <div class="form-group">
        <label for="email" class="col-md-3 control-label"><h5>修改邮件为:</h5></label><br>
        <div class="col-md-6">
          <input type="text" id="email" class="form-control" name="email" value="{sessionScope.email}">
        </div>
      </div>
      <div class="form-group">
        <label for="character_name" class="col-md-3 control-label"><h5>修改昵称为:</h5></label><br>
        <div class="col-md-6">
          <input type="text" id="character_name" class="form-control" name="character_name" value="{sessionScope.character_name}">

```

```

        </div>
    </div>
    </div>
    <label for="user" class="col-md-4"><h5>修改区号为:</h5></label>
    <div class="col-md-1 select">
        <select name="zone_name" class="form-group">
            <option class="form-control" value="1" selected>1</option>
            <option class="form-control" value="2" >2</option>
            <option class="form-control" value="3">3</option>
        </select>
    </div>
    <br>
    <div class="col-md-9 col-md-offset-3 text">
    <button type="submit" class="btn btn-default">提交</button><br>
    <br>
    <a href="userinfo.jsp">返回个人中心</a></div></div></form></div></div>
</div>
<script src="js/jquery-1.7.2.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</body>
<script>
    var xgi ='<%=request.getParameter("xg")%>';
    if(xgi=='yes'){
        alert("修改成功");
    }
</script>
</html>

```

修改用户信息界面截图:



4.8 修改用户信息功能的实现

对于用户信息的修改，只需要获取前端传来的参数，并且进行封装，接着调用业务逻辑层的更新用户的方法就可以实现。

对应的核心代码：

```
package com.etc.servlet;
import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.User;
import com.etc.service.UserService;
import com.etc.service.impl.UserServiceImpl;
@WebServlet("/UpdateUserServlet")
public class UpdateUserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        int user_id=(int) request.getSession().getAttribute("user_id");
        String username=request.getParameter("username");
        String password=request.getParameter("password");
        String email=request.getParameter("email");
        String character_name=request.getParameter("character_name");
        String zone_name=request.getParameter("zone_name");
        request.getSession().setAttribute("username",username);
        request.getSession().setAttribute("password",password);
        request.getSession().setAttribute("email",email);
        request.getSession().setAttribute("character_name",character_name);
        request.getSession().setAttribute("zone_name",zone_name);
        int
money=Integer.parseInt(request.getSession().getAttribute("money").toString());
        try {
            User user = new User(user_id,username,
password,email,money,character_name,zone_name);
            UserService userService = new UserServiceimpl();
            boolean result =userService.updateUser(user);
```

```
        if(result) {
            response.sendRedirect("modify.jsp?xg=yes");
        }else {
            response.getWriter().println("修改失败！ ");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    doGet(request, response);
}
}
```

5 管理员模块设计

5.1 管理员首页界面设计

对于管理员，管理员可以实现用户信息、活动信息、商品信息的管理，所以需要在管理员界面的首页可以有这三个界面跳转的按钮，整体采用 bootstrap 框架进行开发，并且利用 bootstrap 中的小图标进行页面的修饰美化。

对应的核心代码：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>后台管理界面</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
        .bg {
            display: table;
            width: 100%;
            height: 754px;
            text-align: center;
            background: url(imgs/bg3.jpg) no-repeat bottom center;
            background-size: cover;
        }
    </style>
</head>
<body>
<div class="container bg">
<div class="container" style="margin-top:20px;opacity:0.9">
<h1 style="text-align: center;color: #8a6d3b;font-size:50px">欢迎进入管理员界面
</h1>
<h3 style="text-align: center;color: #8a6d3b;font-size:30px">请选择你要进行的操作</h3>
</div>
    <div class="container text-center" style="margin-top:150px ;opacity:0.7">
        <div class="row">
            <div class="col-md-5 col-md-offset-3 text-center" style="font-size:40px">
```

```

        <!--小图标使用空标签定义-->
        <span class="glyphicon glyphicon-user"></span>
        <a style="color: white" href="QueryAllUsersServlet">用户信息管理
    </a></td>
    </div>
</div>
<div class="row">
    <div class="col-md-5 col-md-offset-3 text-center" style="font-size:40px">
        <span class="glyphicon glyphicon-home"></span>
        <a style="color: white" href="QueryAllActivitiesServlet">活动信息管理
    </a>
    </div>
</div>
<div class="row">
    <div class="col-md-5 col-md-offset-3 text-center" style="font-size:40px">
        <span class="glyphicon glyphicon-th"></span>
        <a style="color: white" href="QueryAllCommoditiesServlet">商品信息管
    理</a>
    </div>
</div>
</div>
<script src="js/jquery1.7.2.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</body>
</html>

```

管理员首页对应截图：



5.2 用户信息管理界面设计

对于用户信息管理的界面，采用 bootstrap 框架开发，在此界面可以显示所有的用户的信息，并且可以对用户的信息进行操作。

对应的核心代码：

```
<%@page import="com.etc.entity.User"%>
<%@page import="java.util.List"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>用户信息后台管理</title>
<link href="bootstrap/css/bootstrap.css" rel="stylesheet">
<style type="text/css">
    .bg {
        display: table;
        width: 100%;
        height: 754px;
        text-align: center;
        background: url(imgs/bg4.jpg) no-repeat bottom center;
        background-size: cover;
    }
</style>
</head>
<body>
<div class="container bg">
<h2>用户信息后台管理</h2>
<div class="container" style="background-color:white;">
    <table class="table table-striped">
        <tr><th>用户 id</th><th>用户姓名</th><th>用户密码</th><th>用户邮箱
        </th><th>游戏币</th><th>角色名</th><th>大区名</th><th>注册时间</th><th
        colspan="2">操作</th></tr>
        <c:forEach items="${requestScope.users}" var="item">
            <tr>
                <td>${item.user_id}</td>
```



```

<td>${item.username}</td>
<td>${item.password}</td>
<td>${item.email}</td>
<td>${item.money}</td>
<td>${item.character_name}</td>
<td>${item.zone_name}</td>
<td>${item.register_time}</td>
<td><a
href="QueryUserByUsernameServlet?user_id=${item.user_id}">修改</a></td>
<td><a href="DeleteUserServlet?user_id=${item.user_id}">
删除</a></td>
</tr>
</c:forEach>
<tr><td><a href="manager.jsp">返回</a></td></tr>
</table>
</div>
</div>
<script src="js/jquery-1.7.2.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</body>
</html>

```

用户信息后台管理界面截图：



5.3 后台活动信息管理界面设计

对于管理员来说，可以对活动信息进行管理，包括发布活动、删除活动以及修改活动等。此界面采用 bootstrap 框架进行开发，显示所有活动的信息，并且可以对活动进行增删改查操作。

对应代码：

```

<%@page import="com.etc.entity.User"%>
<%@page import="java.util.List"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>用户信息后台管理</title>
<link href="bootstrap/css/bootstrap.css" rel="stylesheet">
<style type="text/css">
    .bg {
        display: table;
        width: 100%;
        height: 754px;
        text-align: center;
        background: url(imgs/bg4.jpg) no-repeat bottom center;
        background-size: cover;
    }
</style>
</head>
<body>
<div class="container bg">
<h2>活动信息后台管理</h2>
<div class="container">
    <table class="table table-striped">
        <tr><th>活动 id</th><th>活动名</th><th>活动内容</th><th>活动内容
</th><th>活动开始时间</th><th>活动结束时间</th><th colspan="2">操作</th></tr>
        <c:forEach items="${requestScope.activities}" var="item">
            <tr>
                <td>${item.activity_id}</td>
                <td>${item.activity_name}</td>
                <td>${item.activity_content}</td>
                <td>${item.start_time}</td>
                <td>${item.end_time}</td>
                <td><a href="">修改</a></td>
            </tr>
        </c:forEach>
    </table>
</div>
</div>

```

```

        <td><a
href="DeleteActivityServlet?activity_id=${item.activity_id}">删除</a></td>
        </tr>
    </c:forEach>
    <tr><td><a href="addActivity.jsp">新增商品</a></td><td><a
href="manager.jsp">返回</a></td></tr>
</table>
</div>
</div>
<script src="js/jquery-1.7.2.js"></script>
<script src="bootstrap/js/bootstrap.js"></script>
</body>
</html>

```

后台活动信息管理界面截图：



5.4 后台商品信息管理界面设计

对于管理员来说，可以对商品信息进行管理，包括发布商品、删除商品以及修改商品价格等。此界面采用 bootstrap 框架进行开发，显示所有商品的信息，并且可以对商品进行增删改查操作。

对应核心代码：

```

<%@page import="com.etc.entity.User"%>
<%@page import="java.util.List"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>

```

```

<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>用户信息后台管理</title>
<link href="bootstrap/css/bootstrap.css" rel="stylesheet">
<style type="text/css">
    .bg {
        display: table;
        width: 100%;
        height: 754px;
        text-align: center;
        background: url(imgs/bg5.jpg) no-repeat bottom center;
        background-size: cover;
    }
</style>
</head>
<body>
<div class="container bg">
<h2>商品信息后台管理</h2>
<div class="container">
    <table class="table table-striped">
        <tr><th>商品 id</th><th>商品名</th><th>商品价格</th><th colspan="2">操作</th></tr>
        <c:forEach items="${requestScope.commodities}" var="item">
            <tr>
                <td>${item.commodity_id}</td>
                <td>${item.commodity_name}</td>
                <td>${item.commodity_price}</td>
                <td><a href="user_id">修改</a></td>
                <td><a href="DeleteCommodityServlet?commodity_id=${item.commodity_id}">删除</a></td>
            </tr>
        </c:forEach>
        <tr><td><a href="addCommodity.jsp">新增商品</a></td><td><a href="manager.jsp">返回</a></td></tr>
    </table>
</div>
</div>
<script src="js/jquery-1.7.2.js"></script>

```

```
<script src="bootstrap/js/bootstrap.js"></script>
</body>
</html>
```

后台商品信息管理截图：



5.5 后台用户信息管理功能的实现

对于后台用户信息的管理的功能，只需要调用业务逻辑层编写的对于用户的增删改查对应的函数即可实现该功能，而业务逻辑层调用数据库层的相关函数对数据库的数据进行增删改查完成和数据库中的数据的数据的交互。

对应的核心代码：

```
package com.etc.servlet;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.User;
import com.etc.service.UserService;
import com.etc.service.impl.UserServiceImpl;
@WebServlet("/QueryAllUsersServlet")
public class QueryAllUsersServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
```

```

        UserService us=new UserServiceimpl();
        List<User> users = us.findAllUsers();
        System.out.println(users);
        request.setAttribute("users", users);
        request.getRequestDispatcher( "users.jsp").forward(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doGet(request, response);
    }
}

package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.service.UserService;
import com.etc.service.impl.UserServiceimpl;
@WebServlet("/DeleteUserServlet")
public class DeleteUserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        //接收前端传来的 学号
        int user_id =Integer.parseInt(request.getParameter("user_id"));
        UserService us=new UserServiceimpl();
        boolean result = us.deleteUser(user_id);
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        if(result) {
            response.sendRedirect("QueryAllUsersServlet");//重新查询 所有学生
        } else {
            response.getWriter().println("删除失败！");
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // TODO Auto-generated method stub

```

```

doGet(request, response);
    }
}

```

5.6 后台活动信息管理功能的实现

对于后台活动信息的管理的功能，只需要调用业务逻辑层编写的对于活动的增删改查对应的函数即可实现该功能，而业务逻辑层调用数据库层的相关函数对数据库的数据进行增删改查完成和数据库中的数据的数据的交互。

对应的核心代码：

```

package com.etc.servlet;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Activity;
import com.etc.service.ActivityService;
import com.etc.service.impl.ActivityServiceImpl;
@WebServlet("/QueryAllActivitiesServlet")
public class QueryAllActivitiesServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        ActivityService as=new ActivityServiceImpl();
        List<Activity> activities=as.list();
        System.out.println(activities);
        request.setAttribute("activities", activities);
        request.getRequestDispatcher("activity.jsp").forward(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}

package com.etc.servlet;

```

```
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Activity;
import com.etc.service.ActivityService;
import com.etc.service.impl.ActivityServiceImpl;
@WebServlet("/AddActivityServlet")
public class AddActivityServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        SimpleDateFormat simpleDateFormat=new
SimpleDateFormat("yyyy-MM-dd");
        String activity_name=request.getParameter("activity_name");
        String activity_content=request.getParameter("activity_content");

        Date start_time=null;
        try {
            start_time =
simpleDateFormat.parse((request.getParameter("start_time")));
        } catch (ParseException e) {
            // TODO 自动生成的 catch 块
            e.printStackTrace();
        }
        Date end_time=null;
        try {
            end_time =
simpleDateFormat.parse((request.getParameter("end_time")));
        } catch (ParseException e) {
            e.printStackTrace();
        }
        ActivityService as=new ActivityServiceImpl();
```



```

        Activity activity=new
Activity(activity_name,activity_content,start_time,end_time);
        System.out.println(activity);
        boolean result=as.addActivity(activity);
        if(result) {
            System.out.println("success");

        request.getRequestDispatcher("QueryAllActivitiesServlet").forward(request,resp
onse);
        }
        else {
            System.out.println("error");
        }
    }
}

```

```

package com.etc.servlet;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Activity;
import com.etc.service.ActivityService;
import com.etc.service.impl.ActivityServiceImpl;
@WebServlet("/UpdateActivityServlet")
public class UpdateActivityServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        SimpleDateFormat simpleDateFormat=new
SimpleDateFormat("yyyy-MM-dd");
        String activity_name=request.getParameter("activity_name");

```

```

        String activity_content=request.getParameter("activity_content");
        Date start_time=null;
        try {
            start_time =
simpleDateFormat.parse((request.getParameter("start_time")));
        } catch (ParseException e) {
            e.printStackTrace();
        }
        Date end_time=null;
        try {
            end_time =
simpleDateFormat.parse((request.getParameter("end_time")));
        } catch (ParseException e) {
            e.printStackTrace();
        }
        int activity_id=Integer.parseInt(request.getParameter("activity_id"));
        ActivityService as=new ActivityServiceImpl();
        Activity activity=new
Activity(activity_id,activity_name,activity_content,start_time,end_time);
        System.out.println(activity);
        boolean result=as.update(activity);
        if(result) {
            System.out.println("success");
            request.getRequestDispatcher("QueryAllActivitiesServlet").forward(request,response);
        }
        else {
            System.out.println("error");
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doGet(request, response);
    }
}

package com.etc.servlet;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;

```

```

import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.etc.service.ActivityService;
import com.etc.service.impl.ActivityServiceImpl;

/**
 * Servlet implementation class DeleteActivityServlet
 */
@WebServlet("/DeleteActivityServlet")
public class DeleteActivityServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public DeleteActivityServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        //接收前端传来的 学号
        int activity_id=Integer.parseInt(request.getParameter("activity_id"));

        ActivityService as=new ActivityServiceImpl();
        boolean result = as.deleteActivity(activity_id);
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        if(result) {
            response.sendRedirect("QueryAllActivitiesServlet");//重新查询 所有

```

学生

```

        }else {
            response.getWriter().println("删除失败！");
        }
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}

```

5.7 后台商品信息管理功能的实现

对于后台商品信息的管理的功能，只需要调用业务逻辑层编写的对于商品的增删改查对应的函数即可实现该功能，而业务逻辑层调用数据库层的相关函数对数据库的数据进行增删改查完成和数据库中的数据的数据的交互。

对应的核心代码：

```

package com.etc.servlet;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Commodity;
import com.etc.service.CommodityService;
import com.etc.service.impl.CommodityServiceImpl;
@WebServlet("/QueryAllCommoditiesServlet")
public class QueryAllCommoditiesServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
    }
}

```

```

        CommodityService comser=new CommodityServiceImpl();
        List<Commodity> commodities=comser.findAllCommodities();
        System.out.println(commodities);
        request.setAttribute("commodities", commodities);
        request.getRequestDispatcher("commodity.jsp").forward(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}

    ActivityService as=new ActivityServiceImpl();
    Activity activity=new
Activity(activity_name,activity_content,start_time,end_time);
    System.out.println(activity);
    boolean result=as.addActivity(activity);
    if(result) {
        System.out.println("success");
        request.getRequestDispatcher("QueryAllActivitiesServlet").forward(request,response);
    }
    else {
        System.out.println("error");
    }
}
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}

```

```

package com.etc.servlet;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletException;

```

```

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Activity;
import com.etc.service.ActivityService;
import com.etc.service.impl.ActivityServiceImpl;
@WebServlet("/AddActivityServlet")
public class AddActivityServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        SimpleDateFormat simpleDateFormat=new
SimpleDateFormat("yyyy-MM-dd");
        String activity_name=request.getParameter("activity_name");
        String activity_content=request.getParameter("activity_content");
        Date start_time=null;
        try {
            start_time =
simpleDateFormat.parse((request.getParameter("start_time")));
        } catch (ParseException e) {
            e.printStackTrace();
        }
        Date end_time=null;
        try {
            end_time = simpleDateFormat.parse((request.getParameter("end_time")));
        } catch (ParseException e) {
            e.printStackTrace();
        }
    }

package com.etc.servlet;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Commodity;
import com.etc.service.CommodityService;
import com.etc.service.impl.CommodityServiceImpl;
@WebServlet("/UpdateCommodityServlet")
public class UpdateCommodityServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        int
commodity_id=Integer.parseInt(request.getParameter("commodity_id"));
        String
commodity_name=(String)request.getParameter("commodity_name");
        int
commodity_price=Integer.parseInt(request.getParameter("commodity_price"));
        CommodityService cs=new CommodityServiceImpl();
        boolean result =cs.updateCommodity(new
Commodity(commodity_id,commodity_name,commodity_price));
        if(result) {
            response.sendRedirect("QueryAllCommoditiesServlet");//修
改完毕后，再次重新查询全部的学生 并显示
        }else {
            response.getWriter().println("修改失败！");
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}

```

```

package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;

```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.service.CommodityService;
import com.etc.service.impl.CommodityServiceImpl;
@WebServlet("/DeleteCommodityServlet")
public class DeleteCommodityServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        //接收前端传来的 学号
        int commodity_id
=Integer.parseInt(request.getParameter("commodity_id"));
        CommodityService cs=new CommodityServiceImpl();
        boolean result = cs.deleteCommodity(commodity_id);
        response.setContentType("text/html; charset=UTF-8");
        response.setCharacterEncoding("utf-8");
        if(result) {
            response.sendRedirect("QueryAllCommoditiesServlet");
        }else {
            response.getWriter().println("删除失败！ ");
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```


6 活动模块设计

6.1 活动界面设计

对于活动的界面，采用 bootstrap 框架进行开发，首先在此界面的最上面的部分有一个导航条，可以实现各个界面的跳转。

活动界面的中心部分设计了一个轮播图，用于对于活动信息的播报，轮播图显示了各个活动的信息。

活动页面的右边的部分有一个列表用于显示活动的信息，用户点击活动信息可以参与活动。

活动界面对应的核心代码：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述 3 个 meta 标签*必须*放在最前面，任何其他内容都*必须*跟随其后! -->
    <title>活动</title>
    <!-- Bootstrap -->
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
        @media (min-width: 768px) {
            .li-1{
                height: 100px;
                transition:all 0.2s linear;
            }
            .li-1:hover{
                transform-origin:center center;
                transform:scale(1.1,1.1);
            }
            #bg{
                position: absolute;
            }
            .minmin{
```

```
        display: none;
    }
}
@media (max-width: 768px){
    .div-1{
        display: none;
    }
    .div-3{
        display: none;
    }
    .minmin{
    }
}
.h3-1{
    color: #b9def0;
}
.h4-1{
    color: #b9def0;
    width: 400px; overflow: hidden;
    white-space: nowrap;
    text-overflow: ellipsis;
}
.h2-1:hover{
    color: orange;
    text-decoration: underline;
}
.h3-1:hover{
    color: white;
    text-decoration: underline;
}
.h4-1:hover{
    color: white;
    text-decoration: underline;
}
body{
    background-color: black;
    margin: 0px;
    padding: 0px;
}
.nav{
```

```

        background-color: black;
    }
    .button-1{
        top: 20px;
    }
    .me li{
        color: white;
    }
</style>
</head>
<body>
<!--导航栏-->
<div class="container">
    <nav class="nav navbar-inverse" style="padding-right:
20px"><!--inverse 白字黑底-->
        <div class="navbar-header header-1" >
            <button class="navbar-toggle button-1" data-toggle="collapse"
data-target=".navbar-collapse" >
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            
        </div>
        <div class="collapse navbar-collapse navbar-right">
            <ul class="nav navbar-nav ul-1"><!--navbar-nav 横向显示-->
                <li class="li-1"><a href="index.jsp"><h2 class="h2-1"
style="color: gold">首页</h2></a></li>
                <li class="li-1"><a href="userinfo.jsp"><h2 class="h2-1">
个人中心</h2></a></li>
                <li class="li-1"><a href="FindAllCommoditiesSevert"><h2
class="h2-1">游戏商城</h2></a></li>
                <li class="li-1"><a href="FindGoodsALLByUser_Id"><h2
class="h2-1">个人仓库</h2></a></li>
                <li class="li-1"><a href="#"><h2 class="h2-1">社交区
</h2></a></li>
            </ul>
        </div>
    </nav>
</div>

```

```

<!--背景图-->
<div class="div-1" style="position:relative;">
    
    <!--轮播图-->
    <div class="div-2" style="float: left; margin-top: 30%;margin-left:
3%">
        <div id="carousel-example-generic" class="carousel slide"
data-ride="carousel"><!--class 很重要-->
            <!--指示图标-->
            <ol class="carousel-indicators"><!--有序列表-->
                <li data-target="#carousel-example-generic"
data-slide-to="0" class="active"></li>
                <li data-target="#carousel-example-generic"
data-slide-to="1"></li>
                <li data-target="#carousel-example-generic"
data-slide-to="2"></li>
                <li data-target="#carousel-example-generic"
data-slide-to="3"></li>
                <li data-target="#carousel-example-generic"
data-slide-to="4"></li>
            </ol>
            <div class="carousel-inner" role="listbox">
                <div class="item active">
                    <!--alt 图像无法显示时候的文字-->
                </div>
                <div class="item">
                    
                </div>
                <div class="item">
                    
                </div>
                <div class="item">
                    
                </div>
                <div class="item">
                    
                </div>
            </div>
        </div>
    </div>

```



```

<!-- 加载 Bootstrap 的所有 JavaScript 插件。你也可以根据需要只加载单个插件。 -->
<script src="bootstrap/js/bootstrap.js"></script>
<script>
    $('.carousel').carousel({
        interval: 1000
    })
    function showdialog()
    {
        alert("参加成功,点券已发放至你的仓库");
    }
</script>
</body>
</html>

```

活动界面截图:



6.2 查询所有活动功能的实现

对于活动界面上活动的显示,首先需要先从数据库中查询出所有的活动,这一功能,通过在 servlet 中调用业务逻辑层中的查询所有活动的函数来实现,并且把查询到的活动存放到 session 中,方便在前端页面进行显示。

对应的核心代码:

```
package com.etc.servlet
```

```
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Activity;
import com.etc.service.ActivityService;
import com.etc.service.impl.ActivityServiceImpl;
@WebServlet("/QueryAllActivitiesServlet")
public class QueryAllActivitiesServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        ActivityService as=new ActivityServiceImpl();
        List<Activity> activities=as.list();
        System.out.println(activities);
        request.setAttribute("activities", activities);
        request.getRequestDispatcher("activity.jsp").forward(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```

6.3 参加活动功能的实现

对于用户参加活动，首先用户点击这个活动提示是否要参加活动，参加活动的
话会给用户的账户增加游戏币，同时这一次参加活动的记录存放到关系表中，以供
后期其他功能的使用。

对应的核心代码：

```
package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```



```

import com.etc.entity.Activity;
import com.etc.entity.Participate_activities;
import com.etc.entity.User;
import com.etc.service.ActivityService;
import com.etc.service.Participate_activitiesService;
import com.etc.service.UserService;
import com.etc.service.impl.Participate_activitiesServiceImpl;
import com.etc.service.impl.UserServiceimpl;
@WebServlet("/ParticipateActivityServlet")
public class ParticipateActivityServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        int user_id=Integer.parseInt(request.getSession().getAttribute("user_id").toString());
        // 修改剩余金币
        UserService userService = new UserServiceimpl();
        User user = userService.findUserById(user_id);
        int money =user.getMoney();
        money = money +50;
        int activity_id=Integer.parseInt(request.getParameter("activity_id"));
        Participate_activitiesService ps=new Participate_activitiesServiceImpl();
        ps.addParticipate_activities(new Participate_activities(user_id,activity_id));
        user.setMoney(money);
        userService.updateUser(user);
        System.out.println(user.getMoney());
        request.getSession().setAttribute("money",money);
        request.getRequestDispatcher("QueryActivitiesIndexServlet").forward(request,
response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}

```

7 仓库模块设计

7.1 仓库界面设计

对于仓库界面，采用 bootstrap 框架进行开发，首先最上面有一个导航栏用来实现各个界面的跳转。

仓库界面的核心部分显示仓库中具有的物品，并且有丢弃按钮可以丢弃物品，在仓库界面的右上方还显示用户具有的游戏币的数量。

仓库界面对应的核心代码：

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>仓库</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <script>
function panduan() {
    var txt;
    if (confirm("您确定要丢弃该物品吗，不可恢复哦！")) {
        txt = "物品丢弃成功";
        return true;
    } else {
        txt = "您取消了物品丢弃";
        return false;
    }
    document.getElementById("demo").innerHTML = txt;
}
</script>
<style type="text/css">
    @media (max-width: 768px) {
        form {
            display: none;
        }
    }
</style>
```

```

        * {
            margin: 0;
            padding: 0;
        }
        html, body {
            width: 100%;
            height: 100%;
        }
        .bg {
            display: table;
            width: 100%;
            height: 100%;
            opacity: 90%;
            text-align: center;
            background: url(imgs/bg.jpg) no-repeat bottom center;
            background-size: cover;
        }
        section{
            background-image: url("imgs/sebg.jpg");
            width: 100%;
            height: 1080px;
        }
        footer{
            background-color:#3e8f3e;
            padding: 20px;
            font-weight: bolder;
            color: white;
        }
    </style>
</head>
<div class="container bg">
    <header class="container">
        <nav class="nav navbar-inverse" style="padding-right: 20px">
            <div class="navbar-header">
                <button class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>

```

```

        </button>
        <a href="#" class="navbar-brand"></a>
    </div>
    <div class="collapse navbar-collapse navbar-left">
        <ul class="nav navbar-nav">
            <li><a href="index.jsp">首页</a></li>
            <li><a href="#">个人中心</a></li>
            <li><a href="FindAllCommoditiesSevert">游戏商城
</a></li>
            <li><a style="color: gold"
href="FindGoodsAllByUser_Id">个人仓库</a></li>
            <li><a href="#">社交区</a></li>
        </ul>
    </div>
    <!--      在导航条中添加表单-->
    <form class="navbar-form navbar-right" role="search" action=""
method="post">
        <div class="form-group">
            <input type="text" class="form-control"
placeholder="Search">
        </div>
        <button type="submit" class="btn btn-primary">搜索</button>
    </form>
</nav>
</header>
<section class="container">
    <div class="row" style="margin-top: 120px">
        <div class="col-md-6">
            <h2 style="color: white">衣服套装武器皮肤</h2>
        </div>
    </div>
    <div class="col-md-10 col-md-offset-1">
        <div class="col-md-3 col-sm-4 col-xs-6" style="text-align: center; height:
240px;">
            
            <p
style="color: white; size: font-size: large; font-weight: bolder">黑暗骑士</p>
        </div>
        <div class="col-md-3 col-sm-4 col-xs-6" style="text-align:

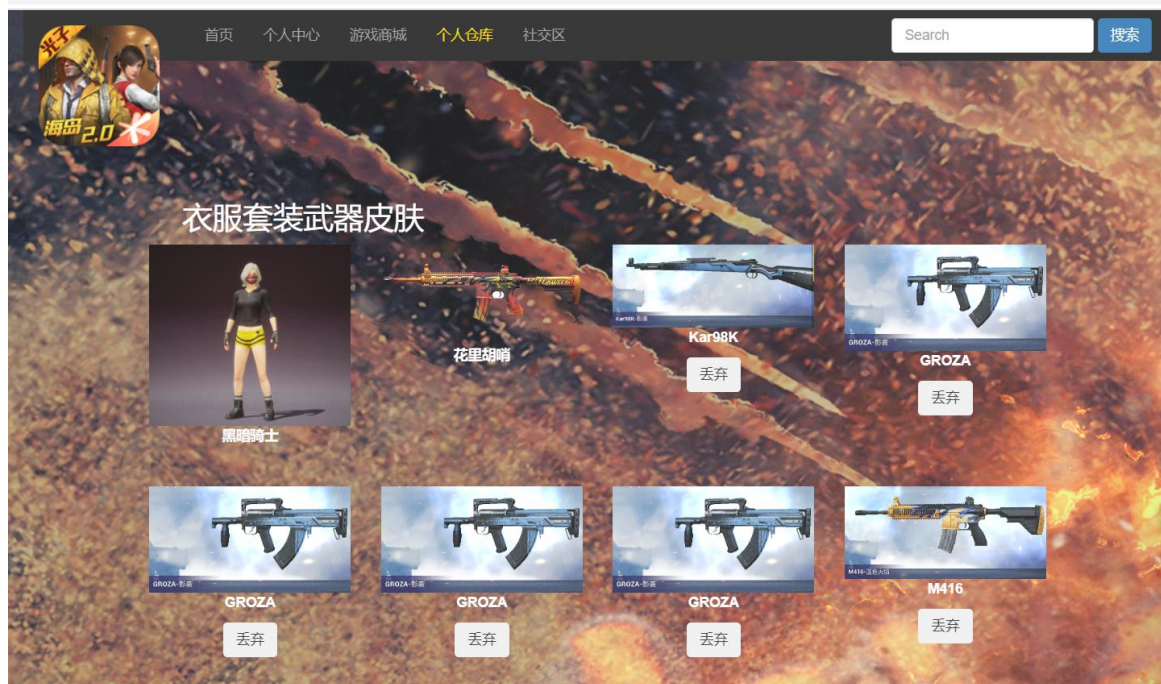
```

```

center;height: 240px;">
                                
                                <p
style="color:white;size:font-size:xx-large;font-weight:bolder">花里胡哨</p>
                                </div>
                                <c:forEach items="${requestScope.goods}" var="item">
                                    <div class="col-md-3 col-sm-4 col-xs-6" style="text-align:
center;height: 240px;">
                                        
                                        <p style="color:white;size:font-size:
xx-large;font-weight:bolder">${item.getGoods_name()}
                                        <form class="" action="DelGoods" method="post"
role="search" onsubmit="return panduan()">
                                            <button type="submit" class="btn">丢弃</button>
                                            <input type="hidden" name="goods_id"
value="${item.getGoods_id()}">
                                        </form>
                                    </div>
                                </c:forEach>
                            </div>
                        </section>
                        <footer class="container text-center">
                            <h4>抵制不良游戏 拒绝垃圾游戏 注意自我保护 谨防受骗上当</h4>
                            <h4>适度游戏益脑 沉迷游戏伤身 合理安排时间 享受健康生活</h4>
                        </footer>
                    </div>
                    <!--头部导航 end-->
                    <!-- jQuery (Bootstrap 的所有 JavaScript 插件都依赖 jQuery，所以必须放在前
边) -->
                    <script src="js/jquery-1.7.2.js"></script>
                    <!-- 加载 Bootstrap 的所有 JavaScript 插件。你也可以根据需要只加载单个插
件。 -->
                    <script src="bootstrap/js/bootstrap.js"></script>
                </body>
            </html>

```

仓库界面截图：



7.2 展示仓库所有物品功能的实现

对于仓库中所有商品的显示，需要从数据库中的 goods 表中根据用户 id 查询出该用户拥有的商品，在 servlet 中调用业务逻辑层编写好的函数，在业务逻辑层调用数据库层对应的函数进行用户中所拥有物品的查询，并且在 servlet 中把查询到的物品存到 session 中，在前端页面对 session 中的查询到的物品进行获取并且利用 jstl 进行循环显示。

对应的核心代码：

```
package com.etc.servlet;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Goods;
import com.etc.service.impl.GoodsServiceImpl;
@WebServlet("/FindGoodsAllByUser_Id")
public class FindGoodsAllByUser_Id extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        request.setCharacterEncoding("UTF-8");
```

```

        GoodsServiceImpl service = new GoodsServiceImpl();
        int
user_id=Integer.parseInt(request.getSession().getAttribute("user_id").toString());
        System.out.println(user_id);
        List<Goods> goods = service.findAllByUser_id(user_id); ;
        System.out.println(goods);
        request.setAttribute("goods", goods);
        request.getRequestDispatcher( "仓库.jsp").forward(request, response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        doGet(request, response);
    }
}

```

7.3 丢弃仓库物品功能的实现

对于用户仓库中的物品用户可以丢弃，对于此功能的实现，只需要调用删除物品表中的一条信息的函数就可以实现此操作。

对应的核心代码：

```

package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.service.GoodsService;
import com.etc.service.impl.GoodsServiceImpl;
@WebServlet("/DelGoods")
public class DelGoods extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        //删除
        request.setCharacterEncoding("utf-8");
        //接收前端传来物品 ID 和用户 ID
        int
user_id=Integer.parseInt(request.getSession().getAttribute("user_id").toString());
        //int user_id=1;
        int goods_id = Integer.parseInt(request.getParameter("goods_id"));
    }
}

```

```
GoodsService service = new GoodsServiceImpl();
boolean result = service.del(goods_id,user_id);
response.setContentType("text/html; charset=UTF-8");
if(result) {
    response.sendRedirect("FindGoodsAllByUser_Id");//重新查询
}
else {
    response.getWriter().println("丢弃失败！");
}

protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    doGet(request, response);
}
```

所有学生

8 商城模块设计

8.1 商城界面设计

对于商城界面的设计，采用 bootstrap 框架进行了开发，商城中的主体部分显示商品的信息，包括商品名、商品价格等，还可以在商城页面看到当前用户剩余的游戏币的数量，用户点击购买按钮可以购买商品。

商城界面对应核心代码：

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="zh-CN">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>商城</title>
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <script>
function panduan() {
    var txt;
    if (confirm("您确定要购买该商品吗？")) {
        txt = "商品购买成功";
        return true;
    } else {
        txt = "购买失败";
        return false;
    }
    document.getElementById("demo").innerHTML = txt;
}
</script>
<style type="text/css">
    @media (max-width: 768px) {
        form {
            display: none;
        }
    }
    * {
        margin: 0;
        padding: 0;
    }
</style>
```

```
}
html, body {
    width: 100%;
    height: 100%;
}
.bg {
    display: table;
    width: 100%;
    height: 100%;
    opacity: 90%;
    text-align: center;
    background: url(imgs/bg.jpg) no-repeat bottom center;
    background-size: cover;
}
section{
    background-image: url("imgs/sebg.jpg");
    width: 100%;
    height: 1080px;
}
footer{
    background-color:#3e8f3e;
    padding: 20px;
    font-weight: bolder;
    color: white;
}
</style>
</head>
<div class="container bg">
    <!-- 顶部导航条-->
    <header class="container">
        <nav class="nav navbar-inverse" style="padding-right: 20px">
            <div class="navbar-header">
                <button class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a href="#" class="navbar-brand"></a>
</div>
<div class="collapse navbar-collapse navbar-left">
  <ul class="nav navbar-nav">
    <li><a href="index.jsp">首页</a></li>
    <li><a href="userinfo.jsp">个人中心</a></li>
    <li><a style="color: gold"
href="FindAllCommoditiesSevert" >游戏商城</a></li>
    <li><a href="FindGoodsAllByUser_Id">个人仓库
</a></li>

    <li><a href="#">社交区</a></li>
  </ul>
</div>
<!--      在导航条中添加表单-->
<form class="navbar-form navbar-right" action="FindComById"
method="post" role="search">
  <div class="form-group">
    <input type="text" name="ComName" class="form-control"
placeholder="Search">
  </div>
  <button type="submit" class="btn btn-primary">搜索</button>
</form>
</nav>
</header>
<!-- 商品图片显示 -->
<section class="container" >
  <div style="text-align: right; ">
    <div class="col-md-4 col-md-offset-8" style="text-align:center;
border-style:solid; border-width:2px; border-color:black">
      <h2 style="color: white">剩余点券: ${sessionScope.money}
</h2>

    </div>
  </div>

  <div class="row" style="margin-top: 120px">
    <div class="col-md-2 ">
      <h2 style="color: white">武器皮肤</h2>
    </div>
  </div>
  <c:forEach items="${requestScope.commodities}" var="item"
begin="0" end="3">

```

```

<div class="col-md-3 col-sm-4 col-xs-6" style="text-align:
center;height: 240px;">
    
    <p
style="color:white">${item.getCommodity_name()}</p>
    <p
style="color:white">${item.getCommodity_price()}点券</p>
    <form class="" action="BuyCom" method="post"
role="search" onsubmit="return panduan()">
        <button type="submit" class="btn">购买</button>
        <input type="hidden"
name="commodity_name" value="${item.getCommodity_name()}">
        <input type="hidden"
name="commodity_price" value="${item.getCommodity_price()}">
        <input type="hidden" name="commodity_id"
value="${item.getCommodity_id() }">
    </form>
</div>
</c:forEach>
<div class="row" style="margin-top:200px;" >
    <div class="col-md-2 ">
        <h2 style="color: white">服装</h2>
    </div>
</div>
<c:forEach items="${requestScope.commodities}" var="item"
begin="4" end="7">

    <div class="col-md-3 col-sm-3 col-xs-6"
style="margin-bottom:100px; text-align: center;height: 240px;">
        
        <p
style="color:white">${item.getCommodity_name()}</p>
        <p
style="color:white">${item.getCommodity_price()}点券</p>
        <form class="" action="BuyCom" method="post"
role="search" onsubmit="return panduan()">
            <button type="submit" class="btn">购买</button>
            <input type="hidden"
name="commodity_name" value="${item.getCommodity_name()}">

```

```

                                <input type="hidden"
name="commodity_price" value="{item.getCommodity_price()}">
                                <input type="hidden" name="commodity_id"
value="{item.getCommodity_id()}">
                                </form>
                            </div>
                        </c:forEach>
                    </section>
                    <footer class="container text-center">
                        <h4>抵制不良游戏 拒绝垃圾游戏 注意自我保护 谨防受骗上当</h4>
                        <h4>适度游戏益脑 沉迷游戏伤身 合理安排时间 享受健康生活</h4>
                    </footer>
                </div>
                <script src="script/jquery-1.12.4.js"></script>
                <script src="bootstrap/js/bootstrap.js"></script>
            </body>
        </html>

```

8.2 展示商城所有物品功能的实现

首先需要从数据库中查询出所有的商品，编写数据库层和业务逻辑层，实现此功能，接着在 servlet 中调用业务逻辑层的查询方法进行查询，并且把查询结果存放到 session 中，在前端页面取出 session 中的数据进行展示。

对应的核心代码：

```

package com.etc.servlet;
import java.io.IOException;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Commodity;
import com.etc.service.CommodityService;
import com.etc.service.impl.CommodityServiceImpl;
@WebServlet("/FindAllCommoditiesSevert")
public class FindAllCommoditiesSevert extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");

```

```
CommodityService comser=new CommodityServiceImpl();
List<Commodity> commodities=comser.findAllCommodities();
System.out.println(commodities);
request.setAttribute("commodities", commodities);
request.getRequestDispatcher("商城.jsp").forward(request, response);
}
}
```

8.3 购买商品功能的实现

用户购买商品时首先根据此商品的 id 查询到此商品的信息，接着把此商品放到仓库中，也就是在物品表中新增一条记录，还需要在购买商品的关系表中新增一条购买记录，对于用户的游戏币也需要做响应的更新。

对应的核心代码：

```
package com.etc.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import com.etc.entity.Goods;
import com.etc.entity.Purchase_commodity;
import com.etc.entity.User;
import com.etc.service.GoodsService;
import com.etc.service.Purchase_commodityService;
import com.etc.service.UserService;
import com.etc.service.impl.GoodsServiceImpl;
import com.etc.service.impl.Purchase_commodityServiceImpl;
import com.etc.service.impl.UserServiceimpl;
@WebServlet("/BuyCom")
public class BuyCom extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        // 购买物品，加入仓库
        int
user_id=Integer.parseInt(request.getSession().getAttribute("user_id").toString());
        String commodity_name = request.getParameter("commodity_name").trim();
```

```
        GoodsService goodsService = new GoodsServiceImpl();
        Goods good = new Goods(commodity_name, user_id);
        goodsService.add(good);
        // 修改剩余金币
        int commodity_price =
Integer.parseInt(request.getParameter("commodity_price"));
        UserService userService = new UserServiceimpl();
        User user = userService.findUserById(user_id);
        int money =user.getMoney();
        money = money - commodity_price;
        user.setMoney(money);
        userService.updateUser(user);
        int
commodity_id=Integer.parseInt(request.getParameter("commodity_id"));
        Purchase_commodityService ps=new Purchase_commodityServiceImpl();
        ps.addPurchase_commodity(new
Purchase_commodity(user_id,commodity_id));
        System.out.println(user.getMoney());
        request.getSession().setAttribute("money",money);
        request.getRequestDispatcher("FindAllCommoditiesSevert").forward(request,
response);
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

9 集成与测试

9.1 项目最终效果截图

普通用户和管理员登录注册页面



普通用户注册页面



个人信息查看界面



抵制不良游戏 拒绝垃圾游戏 注意自我保护 谨防受骗上当
适度游戏益脑 沉迷游戏伤身 合理安排时间 享受健康生活

修改用户信息界面

编号:

修改用户名为:

修改密码为:

修改邮件为:

修改昵称为:

修改区号为:

提交

[返回个人中心](#)

管理员后台首页界面



后台用户信息管理界面

用户Id	用户姓名	用户密码	用户邮箱	游戏币	角色名	大区名	注册时间	操作
6	笨笨	123	1484075660@qq.com	980	明明	2	2020-06-27	修改 删除

[返回](#)

后台活动信息管理界面



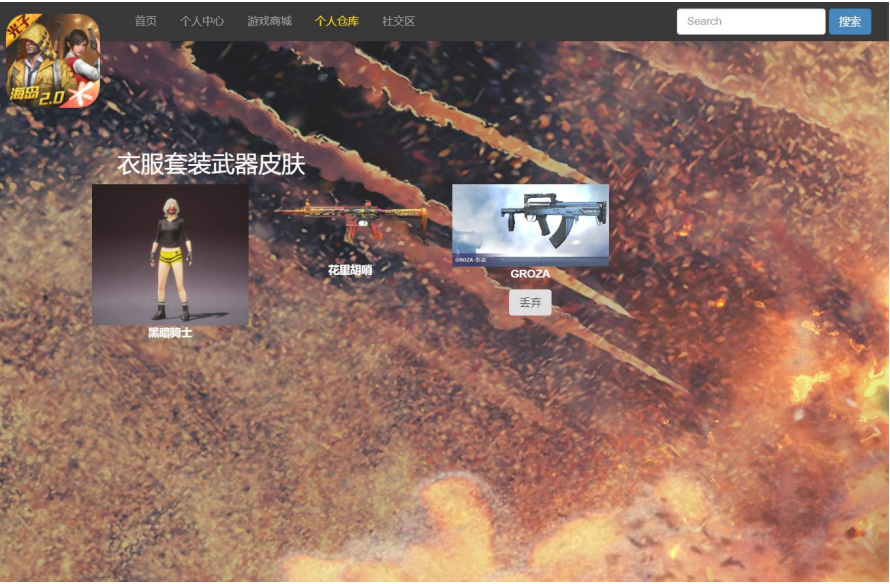
后台商品信息管理界面



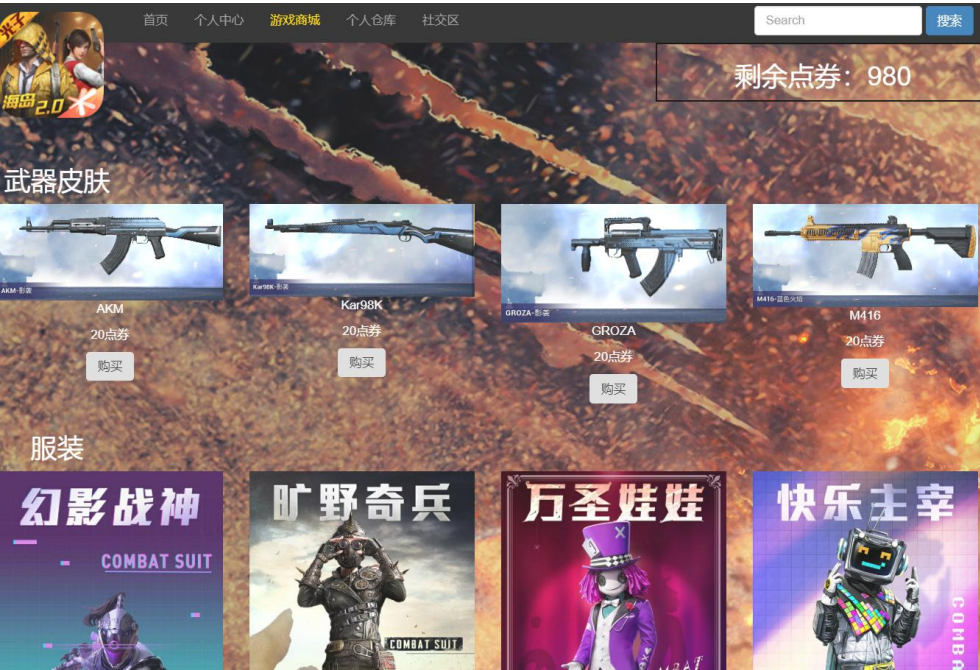
首页界面



仓库界面



商城界面



总 结

首先，一个团队开发一个小型的二级项目，时间较短，很多功能并未完全实现和得到完善，但基本的功能已经实现。项目在深入理解分析系统需求的基础上，运用软件工程、模块化设计思想等进行了数据库概念设计、数据库逻辑设计、系统架构设计，对系统的核心技术进行了分析，并对系统的设计及实现提出了解决方案，解决了部分技术难点。

做项目中途我们碰到了许许多多的问题，比如中文字符编码乱码，图片不能在窗口显示，修改信息后显示信息还是未修改的状态等等，针对这些问题，我们从源头入手，一步步分析，然后进行修改，亦或是咨询老师等，最终成功解决部分问题。

针对这次的项目，我们利用老师本来的框架在其基础上进行增删改查操作，设计项目，然后通过系统架构图，并结合需求，分析出了系统各个部分的工作情况以及数据的交互情况。每个部分的功能通过数据库服务器调用对应函数来实现。项目的创新点在于，从思想入手，由当下最为受欢迎的游戏为主题进行设计，其中我们利用了对象在后台与服务器交换传递数据，将图片名称和图片进行对应，显示在各个界面；添加各个界面内容之间的联系，牵一发而动全身，完美实现功能相关联。

对于此次的课程设计，由于时间问题，功能并没有完美很好的得到实现，比如活动未能点击实时查看内容，只是用户一个简单的参与活动；社交区功能舍弃等，但是基本的功能结构确已完成。若之后的情况允许，我们将进一步完善项目，加以社交区互动的功能，实现好友直接的交互，例如互送礼物等。

这次的课程设计，时间虽短，受益匪浅。既巩固了之前所学习的相关内容，又加深了对前端与后端之间关系的理解；利用了新学习的框架，完成项目功能的部分内容，更加知晓了框架学习使用在开发过程中的必要性。在课程设计的这几天每个人都铆足了劲儿投身项目之中，这种团队精神也值得用到以后的工作学习之中。

参考文献资料

- [1] QST 青软实训. Java8 基础应用与开发[M]. 北京:清华大学出版社, 2015. 6:50~57
- [2] QST 青软实训. Web 前端设计与开发[M]. 北京:清华大学出版社, 2011. 10:22~102
- [3] QST 青软实训. Java web 技术及应用[M]. 北京:清华大学出版社, 2016. 1:203~223