苏州工业园区服务外包职业学院

毕业设计（论文）

基于Java语言的人事管理系统的设计与实现

|  |  |
| --- | --- |
| 届 别 |  |
| 系　　别 |  |
| 专　　业 |  |
| 班　　级 |  |
| 姓　　名 |  |
| 学　　号 |  |
| 指导教师 |  |

**毕业设计（论文）独创性声明**

本人声明，所呈交的毕业设计（论文）系在指导老师的指导下本人独立完成的研究成果。论文中依法引用他人的成果，均已做出明确标注或得到许可。论文内容未包含法律意义上已属于他人的任何形式的研究成果，如论文涉及任何知识产权纠纷，本人将承担一切责任。

学生签名：\_\_\_\_\_\_\_\_\_\_日期：\_\_\_\_\_\_\_\_\_\_

目录

[毕业设计（论文）独创性声明 3](#_Toc22673632)

[第一章 概述 7](#_Toc22673633)

[1.1人事管理系统开发的目的 7](#_Toc22673634)

[1.2人事管理系统开发的可行性 7](#_Toc22673635)

[第二章 开发工具和开发平台 9](#_Toc22673636)

[2.1开发工具的选用及其技术介绍 9](#_Toc22673637)

[2.1.1 Idea 2019工具介绍 9](#_Toc22673638)

[2.1.2 Andriod Studio工具介绍 9](#_Toc22673639)

[2.1.3 Java技术介绍 9](#_Toc22673640)

[2.1.4 MySQL8.0数据库 9](#_Toc22673641)

[2.1.5 MVC架构 10](#_Toc22673642)

[2.1.6 Spring Boot框架 10](#_Toc22673643)

[2.1.7 Android技术介绍 10](#_Toc22673644)

[2.2软件运行环境 10](#_Toc22673645)

[2.3本章小结 10](#_Toc22673646)

[第三章 系统需求分析 11](#_Toc22673647)

[3.1功能需求 11](#_Toc22673648)

[3.1.1基本功能需求列表 11](#_Toc22673649)

[3.1.2系统软件设计需求 11](#_Toc22673650)

[3.2性能需求 12](#_Toc22673651)

[3.3安全需求 12](#_Toc22673652)

[3.4本章小结 12](#_Toc22673653)

[第四章 系统概要设计 13](#_Toc22673654)

[4.1系统模块 13](#_Toc22673655)

[4.2数据库结构设计 13](#_Toc22673656)

[4.2.1数据库概念结构设计 13](#_Toc22673657)

[4.2.2数据库逻辑结构设计 13](#_Toc22673658)

[4.2.3数据库结构设计表 13](#_Toc22673659)

[4.3本章小结 13](#_Toc22673660)

[第五章 系统详细设计与实现 14](#_Toc22673661)

[5.1系统界面设计 14](#_Toc22673662)

[5.1.2系统页面流程关系 14](#_Toc22673663)

[5.2主要功能实现 14](#_Toc22673664)

[5.2.1 系统界面的实现 14](#_Toc22673665)

[5.2.2 切换Fragment 14](#_Toc22673666)

[5.2.3 工资查询 15](#_Toc22673667)

[5.2.4 管理员注册 15](#_Toc22673668)

[5.2.5 考勤总查看 15](#_Toc22673669)

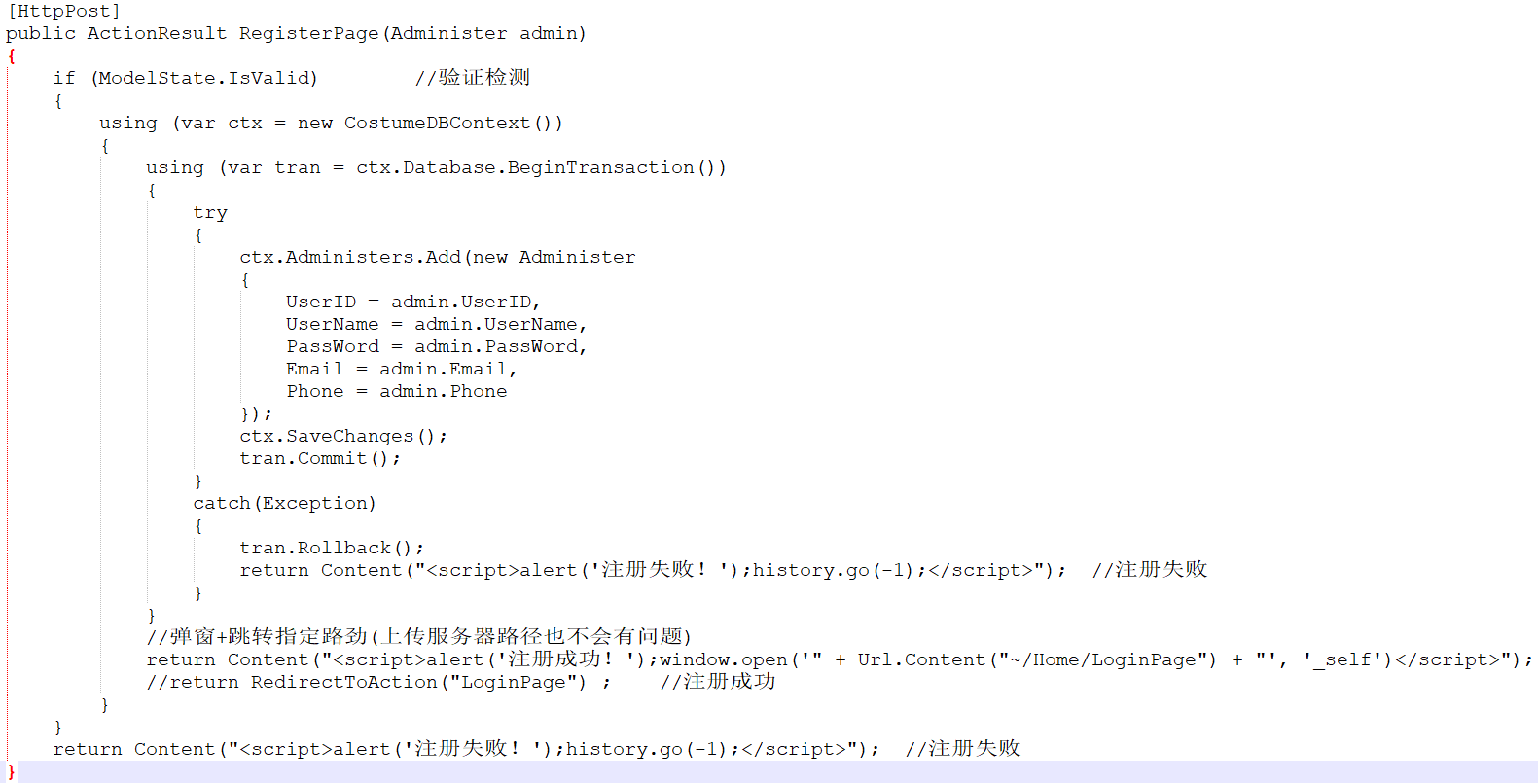
[5.2.5 个人考勤 16](#_Toc22673670)

[5.3本章小结 16](#_Toc22673671)

[第六章 总结与展望 18](#_Toc22673672)

[参考文献 19](#_Toc22673673)

[致 谢 20](#_Toc22673674)

[附 件（代 码） 21](#_Toc22673675)

摘要: 人事管理系统一直都是一个企业管理人员的利器，在现代企业管理中，人事管理系统需要具有高效的网络设计、安全设计以及扩展性设计，在保证系统运行速度的同时也应该保证系统的在高访问量下的稳定性，在本文的基于Java语言的人事管理系统中，使用了前后端分离技术，在Android技术流行的今天，使用该技术与Java后端技术结合起来，以形成一个较为现代化，安全稳定的人事管理系统。

关键词：人事；管理；Java;Android;

Absrtact: Personnel management system has been the most effective tools for a enterprise management personnel, in the modern enterprise management, personnel management system needs to be efficient network design, security design and extensibility, the guarantee system of speed at the same time should also guarantee the stability of the system under high traffic in this article the personnel management system based on Java language, using the front end separation technology, in today's popular Android technology, using the technology combined with Java back-end technology, in order to form a more modern, security and stability of the personnel management system.

Key words: The personnel;Management;Java;Android.

第一章 概述

## 1.1人事管理系统开发的目的

伴随着社会生产力的解放，企业中人力资源管理系统成为一个企业进行集中记录，监测以及劳动分析的利器，相比较旧时使用人工管理，电子化无纸化的人事管理系统成为现代企业人员管理利器，在管理人员调动、生产力分配、企业效益匹配以及提高员工效率中都承担着巨大的作用。

人事管理系统的目的在于需要使用现代化计算机进行高效、安全稳定的管理，减少人力成本的投入，提升企业生产力，将企业人员调配得到合理的安排与把控，将企业的收益得到可见的提升。

在当前的企业人事管理系统中，一般使用B/S架构，但是web端相比较C/S架构存在着个性化特点低、在现在三大主流浏览器内核中(webkit、Trident、Gecko等)需要进行大量的适配，尤其在速度与安全性上需要大量的设计成本，尤其在网络条件较差的环境下访问速度不尽如人意。

## 1.2人事管理系统开发的可行性

基于Java语言的人事管理系统中使用传统的B/S与C/S架构融合模式实现，相比较传统的B/S架构拥有更高的人性化设计以及更快的访问速度以及更高的安全性，而相比较C/S架构而言，本系统拥有更强的系统兼容性、系统拓展性以及更搞笑的部署。本系统运用了多种语言以及技术，具有刚好的灵活性以及兼容性。

技术可行性：本系统主要实现人员录入、人员查询、工资查询与生成、请假以及考勤等等基础功能，从技术层面上说，管理人员可以使用app将数据传输至服务器端，只需要符合服务器接口规范即可将数据插入数据库中，对于开发者而言，使用基于Java语言的Spring Boot框架可实现快速开发、调试以及部署正式环境。

操作可行性：本系统部署于阿里云轻量级云服务器上，使用Windows server2012，在本地测试机开发中，使用jetbrains Idea2019以及其衍生安卓开发工具Android studio 3.x进行开发，系统分别运行在windows以及Android操作系统之上，具有友好的开发界面，并且通过云服务器可以实现长时间不断电稳定运行。

经济可行性：本系统使用阿里云轻量级云服务器，价格相比较购买硬件主机并搭建内网，在硬件成本、电量成本以及运维成本等等具有很大的领先优势，用户可以通过Android管理App实现多端管理，在不使用内网穿透的情况下，外网安装上App即可实现多段互联，更加方便快捷，从经济角度来看，本系统比传统B/S架构系统具有更高的经济效应。

第二章 开发工具和开发平台

## 2.1开发工具的选用及其技术介绍

### 2.1.1 Idea 2019工具介绍

IDEA全称IntelliJ IDEA 是由JetBrains公司开发的一款Java语言集成开发环境。被称为最强Java IDE 相较于Eclipse它的代码提示补全功能十分强大，调试代码便捷，支持代码格式众多等优点。

### 2.1.2 Andriod Studio工具介绍

AndroidStudio开发工具是由Google基于IDEA推出的Android集成开发环境，继承了IDEA的所有优点。基于Gradle构建项目，有着强大的布局管理，实时预览的功能。相较于ADT而言AndroidStudio无论是从易用性上、功能完备性上、都远好于ADT，并且ADT早已停止更新。

### 2.1.3 Java技术介绍

Java是有Sun公司推出的一门面向对象的编程语言，是现今最流行的编程语言，Java具有健壮性、面对对象、可移植性等特性。Java在最初叫做Oak因商标被注册而后更名为Java。在1996年1月推出了JDK1.0。迭代至今已经到JDK13.0，但是目前业界使用最广泛的还是JDK7/8，并且也已经具有一套相当完备的企业开发架构

### 2.1.4 MySQL8.0数据库

MySQL是由MySQL AB公司开发的一个关系型数据管理系统，隶属于Oracle旗下，是世界上使用最广泛的关系型数据库。适用于中小型开发，为多种语言提供API，支持多线程，支持标准SQL语句。MySQL8.0中还具有以下新特性，安全性、优化器索引、通用表表达式、JSON增强、窗口函数、DDL等新特性。相较于Oracle数据库MySQL体积小速度快，并且开源。如不是大型项目MySQL是非常好的选择。

### 2.1.5 MVC架构

MVC是(model view controller)的缩写，是一种设计模式，将业务逻辑处理和用户交际分离开，这样设计可以让后端程序员专注于业务逻辑的，前端程序员专注于界面展示，还能够提高代码的复用性、耦合性底、便于维护。

### 2.1.6 Spring Boot框架

SpringBoot框架是基于Spring框架的一种Java开发框架，致力于简化配置，让开发者从传统的Spring框架的复杂臃肿的配置环境中脱离出来，致力于业务代码的编写。它对大量的技术进行了整合，如持久化技术、视图层技术、安全管理等等。

### 2.1.7 Android技术介绍

Android最初是有Andy Rubin基于Linux开发的一款移动端操作系统。于2005年被谷歌收购。Android在4.0时代之前在功能完备性和界面设计上都是很粗糙，自5.0开始Android在界面设计上首度超越了IOS。Android历经11年的发展，凭借其开源的特性， Android已然成为了全球最大的手机操作系统。拥有者庞大的用户群体。

## 2.2软件运行环境

在本服务器软件中运行平台为Windows Server2012视窗系统，系统开发环境为Idea2019，使用的数据库为Orcle公司的MySQL数据库，系统开发语言为Java语言。客户端软件开发环境为Android9系统，系统开发语言与服务器软件一致都是Java语言，软件开发环境为Android Studio 3.x。

## 2.3本章小结

本章节主要分析本系统使用的开发工具，运行平台以及使用的技术分析，首先介绍了Idea 2019以及Android Studio 3.x，介绍了MySQL数据库以及软件运行环境的介绍。下一章节将会着重介绍数据库表的设计与实现。

第三章 系统需求分析

本系统需求：管理员可以使用客户端软件，添加/删除员工，并且修改员工基本信息，例如员工基本工资、员工所在部门、职位等。并且可以查看当前考勤情况、工资等。

## 3.1功能需求

### 3.1.1基本功能需求列表

如下图3.1基本功能列表所示，本系统的中需要实现的功能分别为员工录入、员工信息修改、考勤查询以及工资查询，工资查询需要与考勤情况相关联，考勤查询中自带请假批准等等功能。

基础功能

员工录入

员工信息修改

考勤查询

工资查询

图3.1 基本功能列表

### 3.1.2系统软件设计需求

在本系统中的软件开发中需要manzu 以下四个要求。

1. 友好的UI(用户界面)：优秀的用户界面是对软件的人机交互、操作逻辑以及界面美观的基本要求。优秀的UI设计不仅仅需要个性、美观，还需要操作舒适、简单以及自由，完成用户与软件之间的双向交互，本文考虑到应用响应以及个性化操作，使用了Android App以及Web嵌套的方式，提高用户体验水平。
2. 优化数据存储服务：人事管理系统需要使用MySQL数据库存储人员私人信息，需要稳定、高效的数据存储服务，在本系统中使用了Spring Boot+MyBatis框架作为后端框架技术支撑，使用原始JDBC Temperate很容易造成SQL注入，使用现代化的ORM框架可杜绝SQL注入等安全问题，提高系统的数据安全性。
3. 软件的可扩展性：本文使用了原生Android App的原生技术与Web技术相结合的技术，小部分功能可以快速迭代、更新，也在运行速度与便捷性上做了平衡，在小版本的更新的过程中，可以使用更新web的方式快速修复bug以及增强软件的扩展性。
4. 软件使用简单高效：在本文的软件设计中，使用了图形化界面，安卓手机安装上App即可使用，管理员可清晰看得出软件中的主要功能分类以进行功能选择。

## 3.2安全需求

本系统中因为属于企业级软件，故在设计的过程中，需要满足以下设计需求。

1. 所有员工账号只有管理员创建，以及员工信息只有管理员可以进行更改。
2. 员工忘记密码可以使用忘记密码功能以实现密码的私有性，保证账号的安全性。
3. 所有的员工密码以及其他隐私信息都保存至数据库中，并且使用MD5加密操作等，以防止密码泄露等安全事故。

## 3.3本章小结

本章节主要在功能需求以及安全需求进行整个软件的需求分析，在此基础上剖析了该系统中的

第四章 系统概要设计

概要设计主要是对系统做一个概括性的介绍，下面对一部分功能模块设计内容进行相应阐述。

## 4.1系统模块

## 4.2数据库结构设计

### 4.2.1数据库概念结构设计

### 4.2.2数据库逻辑结构设计

### 4.2.3数据库结构设计表

表4-2 新闻信息表

|  |  |  |
| --- | --- | --- |
| **字段名** | **类型** | **约束条件** |
| ID | int | 主键，NOT NULL |
| UserID | int | 唯一键,用户id |
| UserName | nvarchar(MAX) | 用户姓名 |
| PassWord | nvarchar(MAX) | 用户密码 |
| Email | nvarchar(MAX) | 用户邮箱 |
| Phone | nvarchar(MAX) | 手机号码 |

## 4.3本章小结

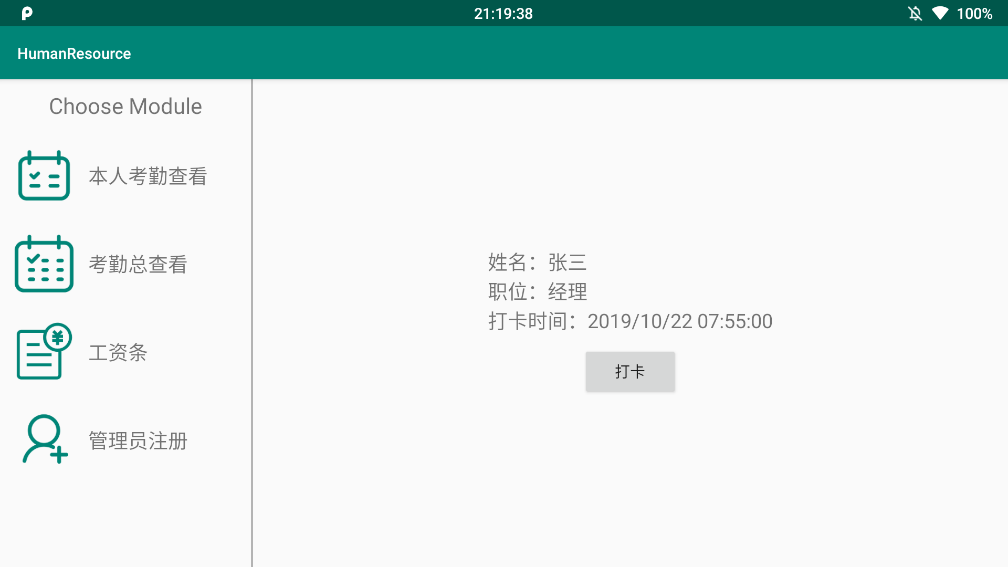
第五章 系统详细设计与实现

## 5.1系统界面设计

### 5.1.2系统页面流程关系

## 5.2主要功能实现

个人考勤查询



总考勤



管理注册



### 5.2.1 登陆界面的实现

1. **public** **void** userLogin() {
2. userName = ed\_username.getText().toString().trim();
3. userPassword = ed\_password.getText().toString().trim();
4. **if** (!TextUtils.isEmpty(userName) && !TextUtils.isEmpty(userPassword)) {
5. user = **new** User();
6. user.setUserName(userName);
7. user.setPassword(userPassword);
8. user.setRemember(isRemember);
9. sentJson.put("", "");
10. okHttpUtils.postJsonRequest(sentJson, "", **new** Callback() {}
11. }
12. }

### 5.2.2 切换Fragment

1. **public** **void** switchFragment(**int** showFragment) {
2. **if** (fragmentActivity != **null**){
3. FragmentTransaction transaction = fragmentActivity.getSupportFragmentManager().beginTransaction();
4. //要显示的Fragment
5. Fragment currentFragment = fragmentList.get(showFragment);
6. //要隐藏的Fragment
7. Fragment hideFragment = fragmentList.get(lastIndex);
8. //将当前的展示Fragment赋值给隐藏的索引下次执行此方法就可隐藏
9. lastIndex = showFragment;
10. transaction.hide(hideFragment);
11. **if** (!currentFragment.isAdded()) {
12. fragmentActivity.getSupportFragmentManager().beginTransaction().remove(currentFragment).commit();
13. transaction.add(R.id.fragmentLayout, currentFragment);
14. }
15. transaction.show(currentFragment);
16. transaction.commitAllowingStateLoss();
17. }
18. }

### 5.2.3 工资查询

1. **public** **void** onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
2. **super**.onViewCreated(view, savedInstanceState);
3. rv\_salary = view.findViewById(R.id.rv\_salary);
4. rv\_salary.setLayoutManager(**new** LinearLayoutManager(**this**.context));
5. salaryAdapter = **new** CommonAdapter(**this**.context,R.layout.salary\_slip,salarySlipList) {
6. @Override
7. **protected** **void** convert(ViewHolder holder, Object o, **int** position) {
8. rv\_salary\_item = holder.getView(R.id.rv\_salary\_grid);
9. RecyclerView.LayoutManager layoutManager = **new** LinearLayoutManager(context);
10. ((LinearLayoutManager) layoutManager).setOrientation(LinearLayoutManager.HORIZONTAL);
11. rv\_salary\_item.setLayoutManager(layoutManager);
12. }
13. };
14. }

### 5.2.4 管理员注册

1. **public** **void** registerAdministroter(){
2. btn\_register = view.findViewById(R.id.btn\_register);
3. btn\_register.setOnClickListener(v -> {
4. userName = ed\_register\_username.getText().toString().trim();
5. userPassword = ed\_register\_password.getText().toString().trim();
6. userEmail = ed\_register\_email.getText().toString().trim();
7. okHttpUtils.postJsonRequest("", "", **new** Callback() {
8. //网络访问成功注册
9. });
10. });
11. }

### 5.2.5 考勤总查看

1. **public** **void** allHumanWorkCheck(){
2. **public** **void** onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
3. **super**.onViewCreated(view, savedInstanceState);
4. rv\_all\_work\_check = view.findViewById(R.id.rv\_all\_work\_check);
5. rv\_all\_work\_check.setRefreshProgressStyle(ProgressStyle.SysProgress);
6. rv\_all\_work\_check.setLayoutManager(**new** GridLayoutManager(**this**.context,6));
7. checkItemList = **new** ArrayList<>();
8. //用于数据展示
9. checkItemList.add(**new** CheckItem("张三",**true**));
10. checkItemList.add(**new** CheckItem("李四",**true**));
11. checkItemList.add(**new** CheckItem("王某某",**false**));
12. checkItemList.add(**new** CheckItem("孙六",**false**));
13. allCheckAdapter = **new** CommonAdapter<CheckItem>(**this**.context,R.layout.check\_item,checkItemList) {
14. @Override
15. **protected** **void** convert(ViewHolder holder, CheckItem checkItem, **int** position) {
16. CheckBox checkBox = holder.getView(R.id.ck\_check\_work);
17. holder.setText(R.id.tv\_name,"姓名：" + checkItem.getCount());
18. checkBox.setChecked(checkItem.isClockIn());
19. }
20. };
21. rv\_all\_work\_check.setLoadingListener(**new** XRecyclerView.LoadingListener() {
22. });
23. rv\_all\_work\_check.setAdapter(allCheckAdapter);
24. }

### 5.2.5 个人考勤

1. **public** **void** humanWorkCheck(){
2. btn\_clock\_in = view.findViewById(R.id.btn\_clock\_in);
3. okHttpUtils = OkHttpUtils.getInstance();
4. okHttpUtils.postJsonRequest("", "", **new** Callback() {
5. @Override
6. **public** **void** onResponse(@NotNull Call call, @NotNull Response response) **throws** IOException {
7. //进行Json数据解析
8. tv\_check\_work.setText("姓名：张三\n职位：经理\n打卡时间：2019/10/22 07:55:00");    //此数据用于开发调试
9. }
10. });
11. }

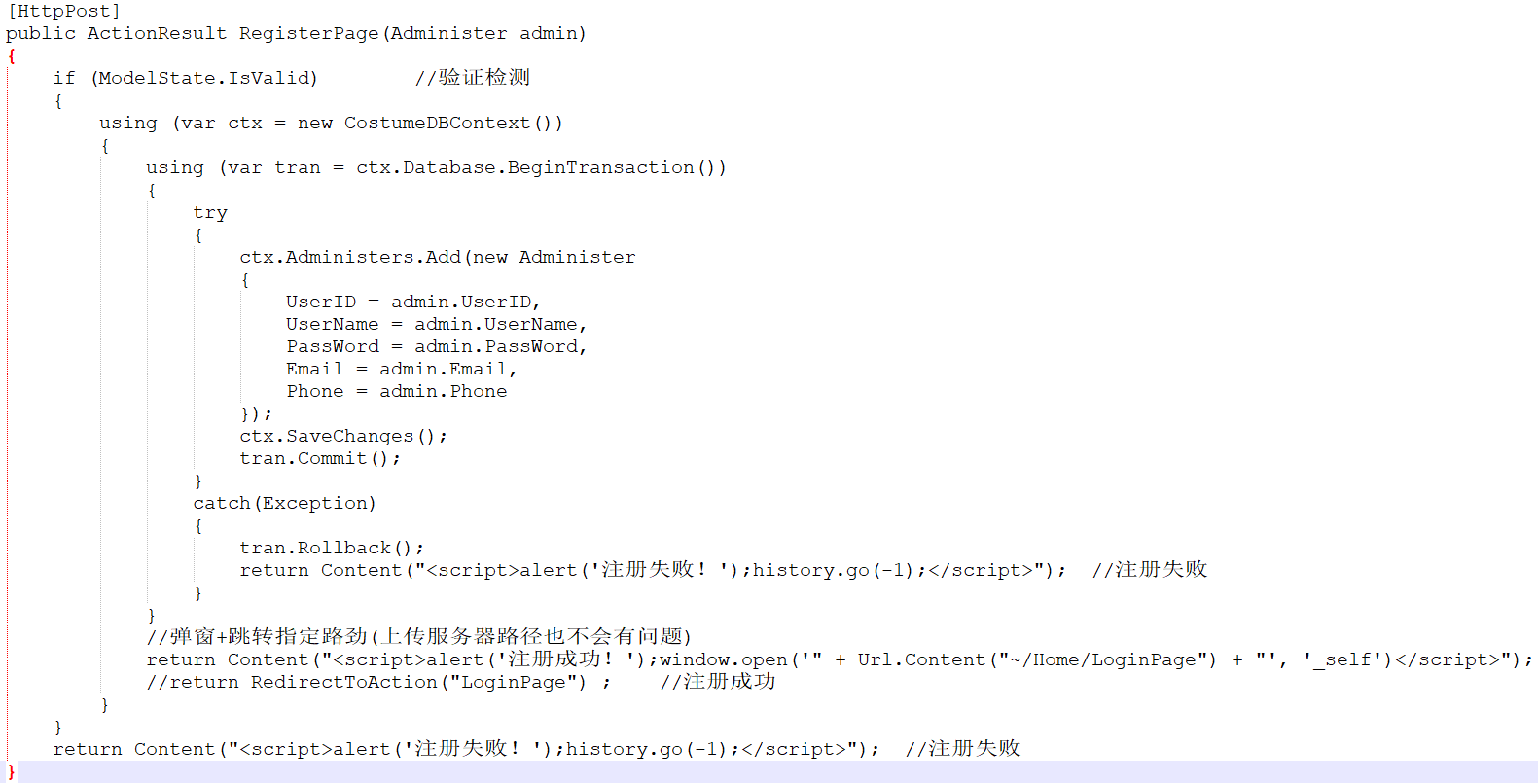
## 5.3本章小结

本章主要介绍了，此系统的Android App使用了FastJSON以及OkHttp类库进行与

第六章 总结与展望

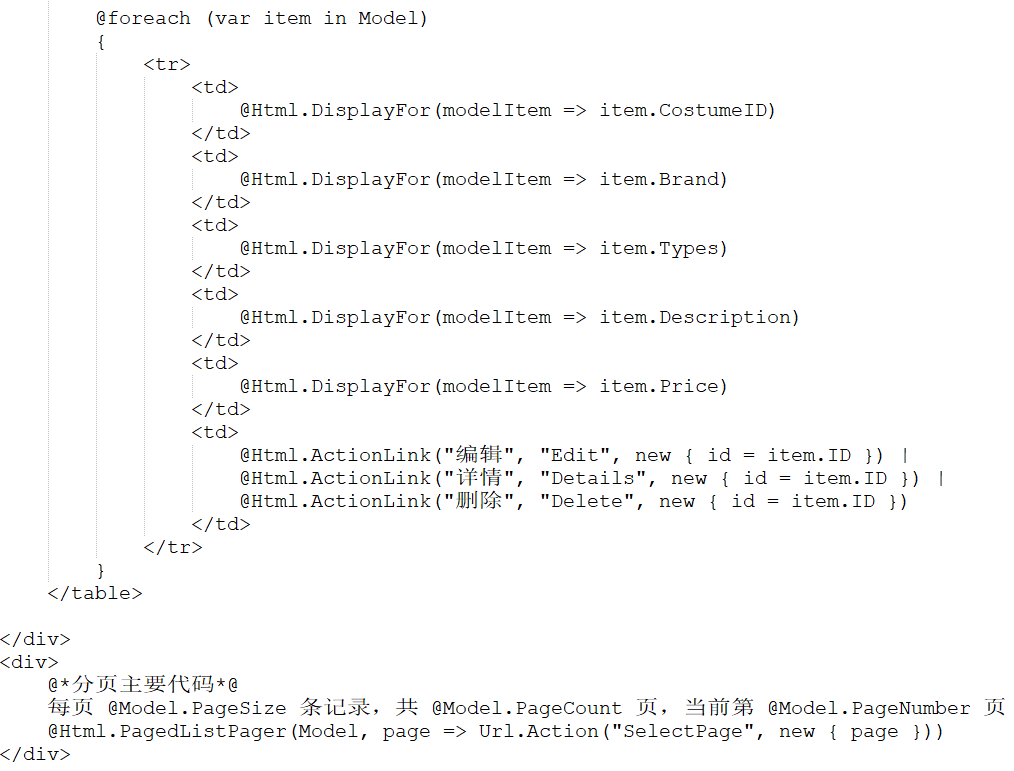
参考文献

# 致 谢

附 件（代 码）

View:





Model:

