**宠物医院管理系统的设计与实现**

|  |  |  |  |
| --- | --- | --- | --- |
| 学 院 | 计算机学院 | | |
| 专 业：  姓 名：  指导老师： | 软件工程 | | |
| 刘春和 | 学 号：  职 称： | 130202021019 |
| 章远 | 讲师 |

中国·珠海

二○一七 年 五 月

**诚信承诺书**

**本人郑重承诺：**本人郑重承诺呈交的毕业设计《宠物医院管理系统的设计与实现》是在指导教师的指导下，严格遵循学校的相关规定，独立完成研究下取得的成果，文中若引有用他人的观点与参考资料，均有在对应位置按先后罗列出其参考的文献，绝无抄袭别人成果及伪造数据等行为，设计中展示的数据完全真实。如设计中存在侵犯知识产权的行为，相应责任均由本人承担。

本人签名：

日期： 年 月 日

**宠物医院管理系统的设计与实现**

**摘 要**

在科技发展日新月异，人们生活水品逐渐提高的今天，社会各个阶层的人都非常青睐于饲养宠物，宠物生病时主人甚是担心，会带上宠物到宠物医院进行治疗，相应地，宠物医院的市场极具发展。再者，光顾宠物医院的人越来越多，医院工作的效率必须得到提高，传统的现场挂号排队，医生手写处方继而排队等待的流程无法满足现阶段的需求，亟需一个管理系统供宠物医院使用，使得医院的业务流程更加快速并且规范化。于是，本人开始设计并实现一个B/S架构的宠物医院管理系统。

本系统基于java SSM框架，数据库为Oracle，分为用户模块，医生模块，管理员模块以及报表中心模块。前台模块供用户浏览医院信息，挂号，订购套餐等流程使用，医生模块供医生发表文章，为宠物开处方等流程使用，管理员模块供管理员管理医院物品进出，员工入职离职等流程使用，报表中心模块医院各级员工查看和导出业绩报表使用。

**关键字**：java; Oracle; 宠物医院; 挂号 ; 开处方; 管理系统

**Design and implementation of online store system Based on J2EE**

**Abstract**

With technology development and popularization of the Internet, using information technology to replace the traditional manual management has spread very popular, and the use of electronic information management of traditional handmade paper lag and the waste of resources and other defects have great advantages, so development of classroom management system as needed to complete the task. This paper aims to design a B/S mode with the function of classroom management system. This system is used. NET architecture, a classroom management system on the network, it is based on the Windows7 system, to SQL Server 2008 database platform for the development, Internet Information Services IIS, ASP.NET (Active Server Page.NET) technology to develop classroom management system. It is by front desk part and background part composition, and which front desk part for user using, main including classroom using information, equipment borrowed information query, equipment maintenance records query, equipment statistics information, I of borrowed records, function; background part by administrator using, main including user registered, classroom using information management, equipment storage management, equipment out library management, equipment inventory information management, administrator management, function.

**Keywords:** c#; ASP.NET; WebForm; SQLserver2008 ; IIS online class system

目录

[1引言 1](#_Toc449544867)

[1.1教室管理系统的研究意义 1](#_Toc449544868)

[1.2教室管理系统的研究现状 1](#_Toc449544869)

[1.3系统设计目标 2](#_Toc449544870)

[1.4系统说明 2](#_Toc449544871)

[2开发环境和使用工具，及涉及技术 2](#_Toc449544872)

[3需求分析 3](#_Toc449544873)

[3.1可行性分析 3](#_Toc449544874)

[3.2 需求综述 3](#_Toc449544875)

[3.2.1用户需求 3](#_Toc449544876)

[3.2.2功能需求 3](#_Toc449544877)

[3.2.3非功能性需求 4](#_Toc449544878)

[3.3角色分析 4](#_Toc449544879)

[3.3详细需求分析 5](#_Toc449544880)

[3.3.1系统使用人员 5](#_Toc449544881)

[3.3.2系统用例 5](#_Toc449544882)

[4概要设计 9](#_Toc449544883)

[4.1 技术的选择 9](#_Toc449544884)

[4.2 构架设计 10](#_Toc449544885)

[4.3功能模块划分 10](#_Toc449544886)

[4.4系统数据模型设计 13](#_Toc449544887)

[4.4.1数据概念结构设计——E-R图 13](#_Toc449544888)

[4.4.2数据库设计 14](#_Toc449544889)

[5系统的设计与实现 19](#_Toc449544890)

[5.1 .NET分层结构的技术划分 19](#_Toc449544891)

[5.2系统用户注册的设计与实现 19](#_Toc449544892)

[5.3 登录的设计与实现 20](#_Toc449544893)

[5.4 教室管理的设计与实现 22](#_Toc449544894)

[5.5 设备管理的设计与实现 24](#_Toc449544896)

[6安装与测试 26](#_Toc449544897)

[6.1安装环境简介 26](#_Toc449544898)

[6.2系统配置 27](#_Toc449544899)

[6.3 系统测试 27](#_Toc449544900)

[6.5总结 29](#_Toc449544901)

[7结论 30](#_Toc449544902)

[参考文献 31](#_Toc449544903)

[致谢 32](#_Toc449544904)

[附录 33](#_Toc449544905)

# 1引言

## 1.1宠物医院管理系统的研究意义

经过自身的调查与观察，现阶段，无论是在社会的哪一个阶层，人们在满足了基本的衣食住行需求后，都开始提升自身的精神追求了，他们当中大部分人青睐于饲养宠物，日常经常会给宠物买价格不菲的食品和营养品，傍晚按照习惯会带上宠物在公园中散步，活动自己宠物的筋骨，可以说，宠物主人和宠物之间往往已经建立了很深厚的感情，所以当宠物生病的时候，主人一定迫切需要一家宠物医院能为宠物治疗，但是，全国的宠物医院数量还是很少，假如宠物医院还是使用传统的现场排队挂号方式来进行挂号业务，那么对于客户来说无疑很浪费时间，因为排队挂号以及等待治疗的时间可以用来处理很多其他的事情。而且，假如宠物主人因为其他种种的原因无法饲养宠物，需要把宠物寄养在其他宠物爱好者那里，也会因为没有相应的平台而烦恼。

现在社会信息化技术的发展突飞猛进，大多数传统企业，政府部门，医院都建立了属于自己的官方网站，人们需要办的业务很多都可以依赖互联网在线上完成，既节省了办理业务者的时间，又节省了受理业务单位的人力资源，物质资源。同样，宠物医院也应该有一个属于自己的管理系统，使得宠物医院不同的工作部门能通过合理的业务流程共享信息以及数据从而进行协同运作，保证业务流程各个环节不出差错，让工作人员进行无纸化办公，还能管理好医院与日俱增的数据，给医院领导随时监控随时监控各个部门相关环节的保证，同时节省了宠物医院客户的时间，带来更优质的用户体验。下面来简述一下本人设计并实现的宠物医院管理系统的优势：

1. 用户可以在医院官网上浏览到有关医院的信息，浏览关于饲养宠物的文章进行学习。
2. 用户可以随时获得医院的套餐信息，在线订购套餐，于线下体验套餐服务。
3. 用户可以拥有几个宠物，可以为其中一个宠物挂号，也可以对其发起寄养申请。
4. 用户可以在官网上发起针对某宠物的领养申请，通过通过上面提供的电话和发起寄养申请的主人进行沟通，等待发起寄养宠物的主人同意成功领养宠物。
5. 医生可以在系统中为宠物开处方，推进无纸化办公的发展，有发表文章的权限。
6. 管理员可以管理医院物品、部门、员工、订单、用户等信息，可以为不用级别的员工开查看相应的报表的权限。能查看以及导出所有的报表数据，给医院领导一个监控医院各个环节是否出错的保障，为日后医院的重大决策提供数据依据。
7. 系统有记录用户操作行为的功能。

## 1.2宠物医院管理系统的研究现状

宠物医院管理系统的设计与实现能帮助宠物医院的业务流程更加规范化，并且能大大节约纸张，钢笔等办公设备的开销，符合创新发展的理念。目前，一些小型的宠物医院依旧停留在手工管理的阶段，医生为宠物开的处方是通过纸张传递，客户带着宠物看病时，假如没有带上之前的看病时候的处方，并且客户也很难把之前的病历说清楚，那么医生将无法获知该宠物之前的病历。并且纸张作为传递媒介还有一个弊端就是容易损坏，已经宠物医院管理系统可以有效提升宠物医院的管理效率，节省办公资源，实现收益最优化。

由于本宠物医院系统是面向社会各种人士使用，所以系统用户部分的操作需要做得简单，界面中出现的文字要让没有计算机技术的用户也能看懂并且懂得操作。而管理员部分，医生部分以及报表部分由于是给内部人员使用，操作的数据量那是相当庞大的，员工入职后应该也会有相应的培训，所以，在开发这些模块时候，应该多考虑业务的规范化。

## 1.3系统设计目标

在本人调研的宠物医院中，普遍都满足客户在官网上在线挂号，然后客户可以凭着挂号单号带上宠物到宠物医院接受治疗，医生可以在系统中为宠物开处方，管理员能处理挂号订单，管理医院的物品器械的进出，员工的入职离职等，由于考虑到现在不少客户有寄养领养需求，并且现在很多宠物医院都推出了美容服务，不少宠物爱好者闲余时间都会带上宠物前去做下美容，一来，为了使本人设计的宠物医院管理系统将来能拥有更多的用户，二来，通过在官网推广自己医院的套餐让更多用户订购能使得宠物医院的利润最大化。所以，在用户模块，除了挂号外，我打算还继续添加寄养领养宠物和订购套餐两个功能。此外，在接收了用户的挂号，订购套餐等操作后，医院后台还应该有相应的处理，例如，医生为挂号订单开处方，管理员处理订单等。

当然，一个完整的宠物医院管理系统肯定不止以上所述的功能，还应该考虑医院内部信息数据如何管理的问题，例如，医院的药品器械信息数量的管理，员工的入职与离职管理等。在宠物医院管理系统数据与日俱增的情况下，还应该开发相应主题的报表，报表的权限由管理员管理，管理员可以为不同级别的员工开不同的报表权限。

## 1.4系统说明

本人设计并实现的宠物医院管理系统目的是为了用户发出业务办理申请到宠物医院受理用户业务请求以及宠物医院内部人员办公等一整套业务流程更加快速化，无纸化和规范化。整个系统分为以下几大模块：

1. 游客模块：游客只有浏览医院简介以及医院发表文章的权限，可以注册成为用户，成为用户后就可以进行挂号等业务操作。
2. 用户模块：用户经过登录后便可以进行为宠物预约挂号，订购套餐，寄养领养宠物等业务操作了。
3. 医生模块：医生经过登录后便可以发表科普类文章到官网上供别人阅读，还能为相应的宠物开处方。
4. 管理员模块：管理员能进行部门和员工管理，物品管理和物品分类管理，用户管理，订单管理，能查看和导出所有的报表，能维护报表系统的权限。
5. 报表模块：根据宠物医院管理系统与日俱增的数据开发生成的相应主题的报表。

# 2开发环境和使用工具，及涉及技术

操作系统：windows 7、 CentOS 6.4；

开发工具： IntelliJ IDEA；

数据库相关： Oracle Database 12c；

涉及技术：Java、Maven、Spring、SpringMVC、MyBatis、Tomcat、JavaScript、nginx、html5、FTP、css3、JQuery。

# 3需求分析

## 3.1可行性分析

经过本人的仔细研究和调查，无论是国内还是国外，现在宠物医院的数量虽然不多，但是还是处于上升趋势的，并且现阶段，无论是宠物医院，还是其他传统行业，都在慢慢往互联网这一大方向发展，所以说，开发一个宠物医院管理系统，是相当有卖点的，它能吸引那些正在运行着传统管理模式，但是又迫切希望医院往信息化方向改革的宠物医院。

再者，本人在宠物医院管理系统开发之前写好了用例，已经明确了系统中各个子功能的具体需求，主题采用的Java SSM框架是一套相当成熟的web开发框架，目前淘宝等大型的互联网企业的网站系统都有运用这个框架，可见这套框架的市场占有率还是挺高的，这么多大型企业在用，证明其稳定性会有一定的保障。而数据库方面我们打算采用由甲骨文公司生产的Oracle数据库，因为这款数据库软件毫无疑问是现在安全性级别最高的关系型数据库，目前银行，证券，政府部门的管理系统一般都在使用这种数据库管理软件，它的可移植性好，吞吐量高，且使用方便。

经过以上市场以及技术方面的可行性分析，可以得知本人这个宠物医院管理系统的设计与实现工作是可行的。

## 3.2 需求综述

### 3.2.1用户需求

本人设计与实现的系统是一款宠物医院管理系统，使用Java语言和Tomcat应用服务器以互联网为平台运行的web系统，用户的所有操作都在web页面上进行。本系统大致上分为游客，用户，医生，管理员，其他员工这五层权限。完成可以满足用户在线为宠物挂号，订购套餐，寄养领养，医生为宠物开处方，管理员管理医院物品以及员工信息，订单信息，管理医院数据报表以及管理各级员工查看各种报表的权限这一大致业务流程，确保该管理系统日后能安全，稳定，可靠地运行。能通过这款管理系统，为社会上所有的宠物爱好者带来更优质的体验，并且使宠物医院的业务流程更加规范化，收益最大化。

### 3.2.2功能需求

本系统主要分为：游客模块，用户模块，医生模块，管理员模块，报表模块这五个模块。

1. 游客模块：游客不用登录都可以浏览医院发表的文章，套餐信息以及待领养宠物信息等，可以通过注册成为系统用户。
2. 用户模块：用户经过登录后就有属于自己的宠物列表，一个用户可以拥有多个宠物，用户可以管理属于自己的宠物信息，可以对其中某个宠物进行挂号，发起寄养申请等业务，还能在网站上订购宠物医院推出的服务套餐或者发起领养宠物申请。
3. 医生模块：医生可以为具体挂号的宠物开处方。
4. 管理员模块：管理员可以管理医院的物品分类，物品信息，部门架构，员工信息，用户信息，员工信息，管理报表中心的报表以及报表中心用户的权限。
5. 报表模块：根据宠物医院与日俱增的信息数据生成相应的数据报表供医院内部员工查看。

### 3.2.3非功能性需求

开发该宠物医院管理系统时候，应该满足以下非功能需求：

1. 性能需求：系统界面设计人性化，前台部分界面出现的文字要通俗易懂，后台部分非计算机专业的员工经过简单培训后都能熟练操作系统。
2. 安全性需求：密码不能明文存储，要经过加密后存储。未登录用户访问需要登录才能访问的链接时候要有相应的拦截器拦截。
3. 可靠性需求：系统上线后尽量避免发生故障，可维护强。

## 3.3角色分析

根据所调研的宠物医院内部规章流程，系统分为五个角色：游客、普通用户、医生、管理员、报表中心用户。

下面为每个角色进行详细分析：

1. 游客
2. 游客无需登录，进入网站主页后可以查看医院文章列表。
3. 游客可以阅读宠物医院发表的文章。
4. 查看宠物医院相关介绍。
5. 查看待领养宠物列表。
6. 查看宠物医院推出的套餐信息。
7. 游客可以通过注册成为系统用户。
8. 普通用户
9. 用户进行业务操作时候必须要先登录。
10. 用户可以在自己的宠物列表中新增宠物。
11. 用户可以管理自己的宠物信息，对宠物进行增加，删除，修改。
12. 用户可以为自己宠物列表中某一个宠物进行挂号。
13. 用户可以对自己宠物列表中的某一个宠物发起寄养申请，并且可以同意其他用户对该宠物的领养申请。
14. 用户可以对待领养宠物发起领养申请。
15. 用户可以在线订购宠物医院推出的服务套餐。
16. 医生
17. 医生办公时需要进行登录。
18. 医生可以发表科普类文章到网站。
19. 医生可以查询与自己部门相关的挂号订单。
20. 医生可以为与自己部门相关并且还没被其他医生处理的订单上的宠物开处方。
21. 管理员
22. 管理员办公时需要登录后台系统。
23. 管理员可以管理宠物医院物品分类，对物品分类进行添加，删除以及修改。
24. 管理员可以管理宠物医院的物品库存，，价格，供应商等信息。
25. 管理员可以管理宠物医院部门架构，可以新增宠物医院部门，撤销宠物医院部门，设置哪个部门可以为可挂号部门以及对宠物医院部门进行重命名。
26. 管理员可以管理宠物医院前台网站的内容信息。
27. 管理员可以管理宠物医院普通用户的信息，设定其用户级别。
28. 管理员可以管理客户订购套餐的订单，修改订单的状态信息。
29. 管理员可以管理医院客户的挂号订单，核算其价格，修改订单的状态信息。
30. 管理员可以管理医院报表系统的报表，修改报表的状态信息。
31. 管理员可以管理医院报表系统的登录账户，可以添加，删除或者修改报表系统账户信息，设定该账户查看报表信息的权限。
32. 报表用户
33. 报表用户进行操作时需要登录系统。
34. 报表用户登录后能查看自己拥有查看权限的数据报表，可以导出该数据报表。

## 3.3详细需求分析

### 3.3.1系统使用人员

本人设计并实现的宠物医院管理系统的角色包括游客、普通用户、医生、管理员、报表用户五个。

### 3.3.2系统用例

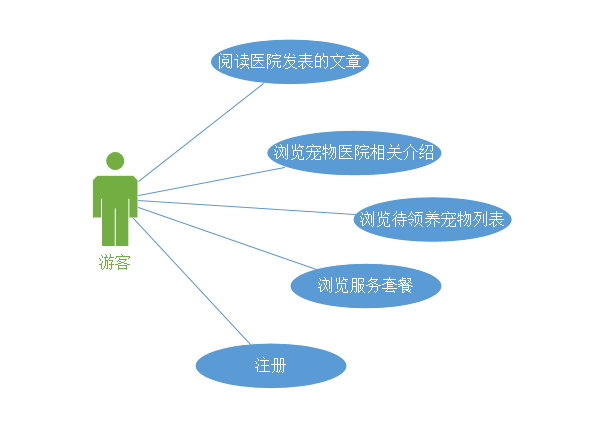


图3.3.2.1系统游客用例

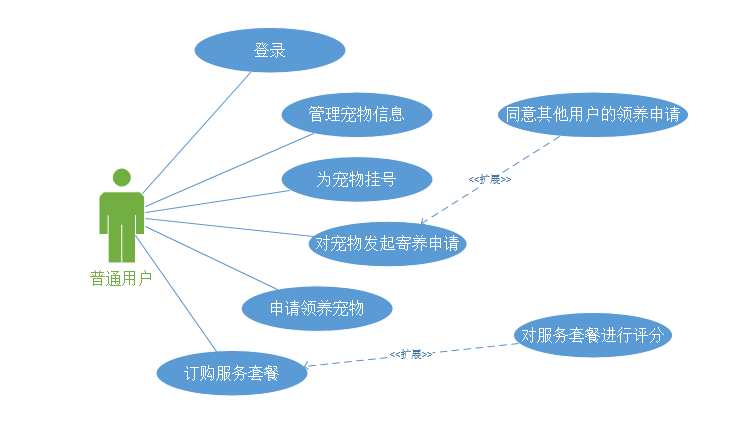


图3.3.2.2系统普通用户用例

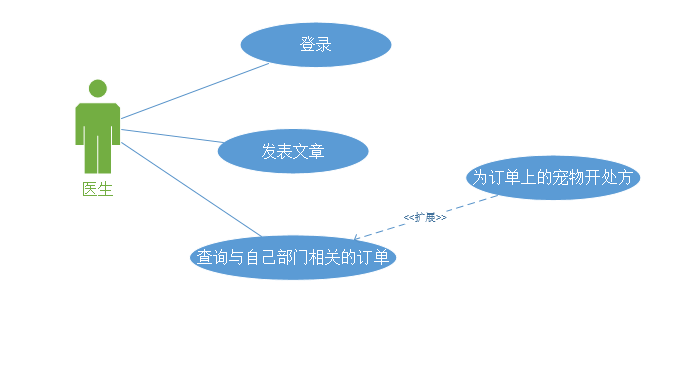


图3.3.2.3医生用例

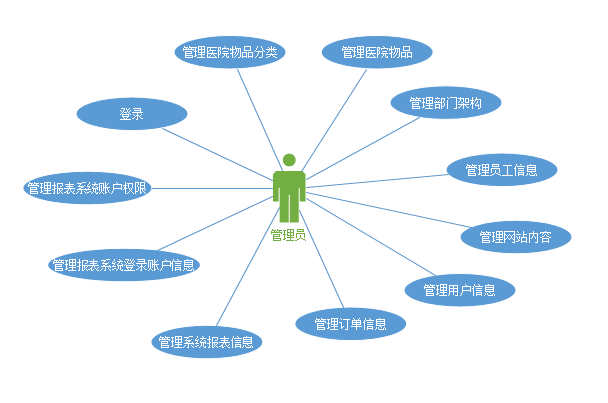


图3.3.2.4系统管理员用例

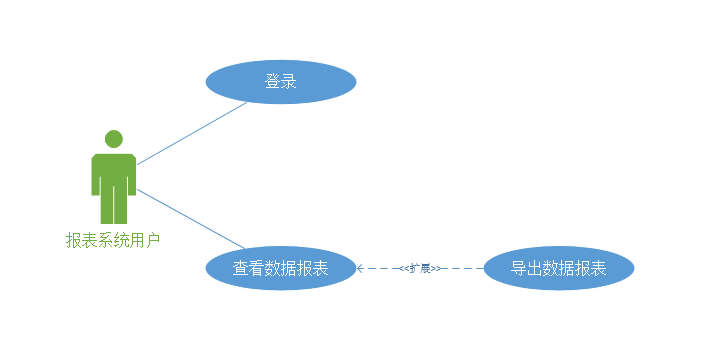


图3.3.2.5报表用户用例

**用例示范1：**游客注册

**范围**：本宠物医院管理系统

**角色**：游客

**前置条件：**游客进入宠物医院管理系统网站用户注册页面。

**后置条件：**注册成功。

**主成功场景：**

1. 游客点击宠物医院官方网站免费注册链接；
2. 游客进入注册页面；
3. 在注册页面按照要求正确填写个人信息，如用户名，密码等信息；
4. 点击注册按钮提交注册申请；
5. 网站出现注册成功提示信息。

扩展点**：**

1. 游客填写的信息不符合网站要求或者有信息栏没有填写

a1.登录名不是手机或者邮箱格式，系统提示请输入正确的手机号或者邮箱进行注册。

a2.登录名在系统中已经存在，系统提示请重新输入登录名。

a3.密码安全性级别不够，不符合网站要求，系统提示重新输入密码。

a4.两次输入的密码不一致，系统提示请检查两次密码是否一致。

**用例示范2：**普通用户登录

**范围**：本宠物医院管理系统

**角色**：普通用户

**前置条件：**登录账号已经在本宠物医院管理系统注册，并且正确填写用户名和密码信息。

**后置条件：**登录成功。

**主成功场景：**

1. 用户点击宠物医院官方网站登录链接。
2. 进入登录页面；
3. 用户在登录页面按照要求正确输入用户名和密码。
4. 用户点击登录按钮；
5. 登陆成功，跳转到宠物医院官方网站首页，在首页右上角显示用户昵称以及欢迎信息；

扩展点**：**

1. 用户输入的密码错误，网站提示账号或者密码错误，登录失败。

**用例示范3：**新增员工

**范围**：本宠物医院管理系统

**角色**：管理员

**前置条件：**管理员输入正确的管理员账号和密码成功登录，进入管理员管理系统首页。

**后置条件：**成功新增员工

**主成功场景：**

1. 管理员点击新增员工按钮进入新增员工界面；
2. 管理员点击选择部门按钮，系统弹出部门选择窗口；
3. 管理员展开部门架构图；
4. 管理员点击选择新增员工所在的部门；
5. 部门选择窗口关闭，管理员选择的部门信息显示在选择部门按钮旁。
6. 管理员继续填写员工的姓名，年龄，身份证等信息。
7. 管理员点击上传照片按钮，弹出选择图片窗口。
8. 管理员点击添加图片按钮，从本地选择新增员工的照片。
9. 管理员点击开始上传按钮。
10. 系统提示上传图片成功。
11. 管理员点击全部插入按钮。
12. 选择图片窗口关闭，插入的图片显示在新增员工界面上传照片下面。
13. 管理员点击提交按钮，系统提示新增员工成功。

扩展点**：**

1. 管理员在表单没有填写完成的情况下就点击提交按钮，系统弹出一个框提示用户表单还没填写完成，并在表单还没填写的文本框隔壁提示此输入项为必输项。

**用例示范4：**撤销部门

**范围**：本宠物医院管理系统

**角色**：管理员

**前置条件：**管理员在后台管理系统登录页面输入正确的管理员账号和密码并成功登录，进入管理员管理系统首页。

**后置条件：**成功撤销部门

**主成功场景：**

1. 管理员点击部门管理按钮进入部门管理页面；
2. 管理员展开部门架构图，直到找到需要撤销的部门；
3. 点击要撤销的部门，然后点击鼠标右键；
4. 找到删除按钮，点击一下；
5. 系统弹出窗口提示撤销部门成功；
6. 管理员在窗口点击确定按钮，提示窗口关闭。
7. 撤销的部门在部门架构图中自动移除。

扩展点**：**

1. 需要删除的部门下面还有子部门或者还有在职的员工没有处理，系统提示必须处理完该部门下面的子部门或者在职员工才能撤销该部门。

**用例示范5：**新增宠物

**范围**：本宠物医院管理系统

**角色**：普通用户

**前置条件：**使用自己在宠物医院管理系统注册的账号成功登录网站，进入网站首页。

**后置条件：**成功新增宠物

**主成功场景：**

1. 用户在首页点击我的宠物按钮，进入自己的宠物列表的第一页；
2. 用户点击新增按钮，进入新增宠物页面；
3. 用户按照要求填写宠物的类型，年龄，性别等信息；
4. 用户点击上传照片按钮，系统弹出图片选择窗口；
5. 用户点击添加图片按钮，从本地选择新增宠物的照片；
6. 用户点击开始上传按钮，开始上传宠物照片；
7. 系统提示上传照片成功；
8. 用户点击全部插入按钮；
9. 选择图片窗口关闭，插入的图片显示在新增宠物界面上传照片下面。
10. 用户点击新增按钮，网站跳转至宠物列表第一页，新添加的宠物出现在宠物列表的第一位。

扩展点：

1. 用户填写的宠物信息不符合网站要求或者表单没有填写完成

a1.宠物年龄超出限制，系统提示宠物年龄必须在1-30之间；

a2.宠物昵称字符长度超出，系统提示宠物昵称限制在30个字符内；

a3.宠物性别输入为男或者女时，系统提示宠物性别只能为公或者母；

a4.表单没有填写完成时，用户点击新增按钮，系统提示填写信息或者表单没有填写完成，请仔细检查。

# 4概要设计

经过前面的需求分析，本人已经明确了系统需要实现哪些功能，满足怎样的效果，现在就开始本系统的概要设计了。

所谓概要设计，就是系统开发前期构思并设计如何实现该系统的问题，分为两个重要的阶段，它们分别是系统设计以及结构设计的两大阶段。系统设计阶段主要解决的问题是根据提出的需求来确定这个系统应该采用什么什么开发语言进行开发，采用什么开发框架从而加快开发效率，采用什么数据库进行存储和管理数据能保证数据的准确性和安全性。结构设计阶段主要解决的问题是确定系统的整体架构是怎样的，包括哪些模块，模块与模块之间怎样联系在一起，会产生什么样的特殊情况，应该怎样解决这些将会出现的问题。

## 4.1 技术的选择

本系统为一个B\S架构宠物医院管理系统，虽然目前出现了很多新的网站开发技术，很多企业也逐渐在项目中引入这些新技术，以缩短开发周期，使自己的产品能更快地上线。哪怕就像京东，淘宝这些大企业，在刚刚开始成立的时候，也并没有采用Java技术进行网站开发，淘宝刚成立的时候用的是PHP技术，而京东刚成立的时候用的是C#技术进行网站开发。但是，在他们的企业开始迅速发展，业务量急剧增加后，才发现，他们所采用的网站开发技术无法支撑得起公司大量的业务量。所以，后来，他们全都改为了使用Java 作为网站开发的首要技术选择。很感谢我的学校，开设的课程一直都能适应社会的需求，在大三下学期的软件工程综合实训课程上，我掌握了Java EE技术。毕业设计是检验大学四年的学习成果的一个很重要的环节，所以我采用Java SSM框架技术来进行系统开发。而由于使用Java开发的过程中，都需要引用第三方的开源类库，这时候问题就出现了，随着第三方类库依赖逐渐增多，不但每次的引进的操作太过繁琐，而且可能还会出现版本冲突这些比较严重的问题，这时候，我们就很有必要在项目开发中引进依赖管理工具maven。数据库部分，由于医院管理系统应该更加注重数据准确性，安全性，所以本人采用了目前为止安全性级别最高的关系型数据库Oracle。而作为一个管理系统，使用者难免会有上传图片的需求，以前软件工程综合实训课做的项目是直接把图片上传到web应用中，但是现在我打算在开发过程中搭建一台ftp服务器，把图片上传到ftp服务器，由Nginx作处理静态资源服务器，因为Nginx具有高并发，响应快的特点，它对图片等静态资源的处理的能力更更优。

## 4.2 构架设计

本人参照实习单位的开发规范图搭建的宠物医院管理系统工程结构图如下：

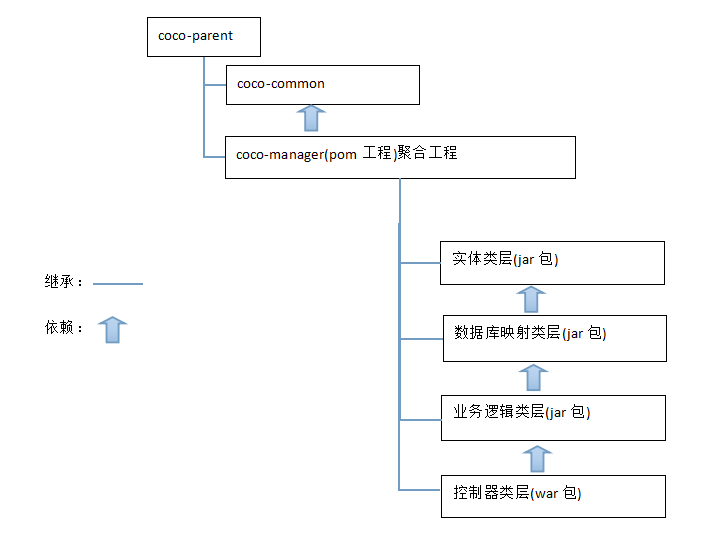


图4.2工程结构图

说明：

1. parent -- 管理依赖jar包的版本，这是公司级别工程的聚合工程。
2. common -- 管理通用组件、工具类。
3. coco-manage -- 宠物医院管理系统工程。
4. 由于实体类层，数据库映射类层，业务逻辑类层都为jar包，所以里面不能放配置文件，所有的配置文件都放在控制类层。

## 4.3功能模块划分

本宠物医院管理系统将根据游客，普通用户，医生，管理员，报表用户五个角色来进行模块结构的分解，可以得到功能模块如下图所示：

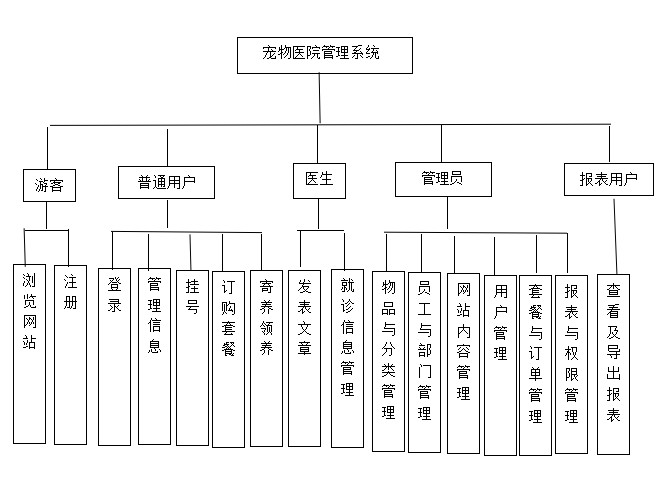


图4.3.1 宠物医院管理系统的功能模块

如下为各个模块功能简介：

1. 游客浏览网站：为了使更多的网友了解本宠物医院，吸引更多的网友注册，本宠物系统有关医院介绍，发表的文章，推出的套餐，待领养宠物的列表等内容网友无需注册都可以查看，有业务办理需求时才需要注册。
2. 游客注册：本宠物医院管理系统普通用户注册功能由游客自主在宠物医院官网完成，游客只需要按照要求正确填写个人信息就可以完成注册。
3. 用户登录：用户进行网站首页点击登录按钮，或者处于未登录状态时点击了订购套餐，我要领养等需要登录才能进行业务的按钮时都会跳转到登录页面，正确填写账号密码后点击登录按钮便可以完成登录。
4. 管理信息：用户可以进入个人信息页面修改自己的个人信息，也可以进入我的宠物页面管理与自己有关的宠物列表中所有宠物的信息。
5. 挂号：用户可以进入我的宠物列表中找到需要挂号的宠物那一栏点击挂号，然后在挂号页面为宠物选择挂号的部门，再选择预约时间即可完成挂号。
6. 订购套餐：用户可以浏览医院推出的服务套餐，选择需要订购的服务套餐点击我要订购按钮进行订购。
7. 寄养领养：寄养领养功能是本宠物医院管理系统提供给广大宠物爱好者的一个平台，管理员并不会介入。一次完整的寄养领养操作必须得有两个用户参与才能成功。用户A进入到我的宠物页面，找到需要寄养的宠物那一栏点击寄养，然后在寄养申请页面填写寄养相关信息，点击寄养按钮后该寄养信息就能发布到网站的的我要领养页面，用户B在看到用户A的寄养信息后，会通过寄养信息上面的联系方式与对方沟通好，随后发起寄养申请，用户A在自己的宠物列表中可以查看到用户B的领养申请，在点击同意后该宠物就会添加到用户B的宠物列表中了。
8. 发表文章：医生登录后，可以发表文章列表中发表科普类的文章，网站首页文章列表有一栏是专门展示科普类文章的，相应地，医生发表的文章都会在该列表显示。
9. 就诊信息管理：因为用户挂号是选择医院部门进行挂号的，一个部门里面有很多个医生，所以医生点开待就诊列表，出现的是与该医生所在部门相关的待就诊列表，医生可以为该列表上面的挂号单号开处方。
10. 物品与分类管理：管理员可以管理医院物品分类，物品数量，价格，供应商等信息。
11. 员工与部门管理：管理员可以管理宠物医院部门的新增，撤销以及重命名，可以设置特定的部门为可挂号部门，可以管理宠物医院员工的姓名，身份证，工资等信息。
12. 网站内容管理：管理员可以管理宠物医院网站前台文章，医院简介等内容。
13. 用户管理：管理员可以管理用户信息，设定用户级别。
14. 套餐与订单管理：管理员可以发布新的套餐，修改套餐信息或者下架套餐，可以管理套餐订单，修改该套餐订单的状态，还可以核算挂号订单价格，修改其状态。
15. 报表与权限管理：管理员可以管理宠物医院所有报表信息，设定哪些报表为可用报表，还能管理报表系统的登录账户，设定这些账户有查看哪些报表的权限。
16. 查看及导出报表：报表用户登录后，系统会根据这个登录账户能查看哪些报表在首页显示哪些报表，用户点击特定的报表按钮，就能查看该报表，用户还能点击导出按钮导出该报表。

## 4.4系统数据模型设计

### 4.4.1数据概念结构设计——E-R图

E-R模型，就是实体-联系模型，实体就像现实世界中出现的学生，教师，课程，联系就像选修，实体学生选修了实体课程，这只是一个简单的例子。对于一个复杂的系统，实体与实体之间怎样联系起来往往没有以上所举的例子那么简单，在我们对我们所在的真实世界的事务进行抽象化时，思维可能会很乱，所以我们就要借助E-R图来描述实体与实体之间的联系，这样不但能很方便地构造系统的信息数据结构，开发人员在程序编码阶段也能少走很多弯路，这是理解清楚系统中实体之间对应关系的一种有效方式，它主要由如下三个部分组成：

1. 矩形框：实体用矩形框表示，实体的名称记录在矩形框内。
2. 棱形框：实体之间的联系用棱形框表示，联系的名称记录在棱形框内。
3. 椭圆形框：无论是实体还是联系，他们的属性都用椭圆形框表示，属性名称记录在椭圆形框内，如果属性名称下面多了一条划线，那么就代表该属性为主属性。
4. 连线：作为一条桥梁，架设在实体与属性或者实体与联系之间，在实体与联系之间，往往会在连线上标明连接类型。

此系统的E-R图设计如图4.4.1.1所示：

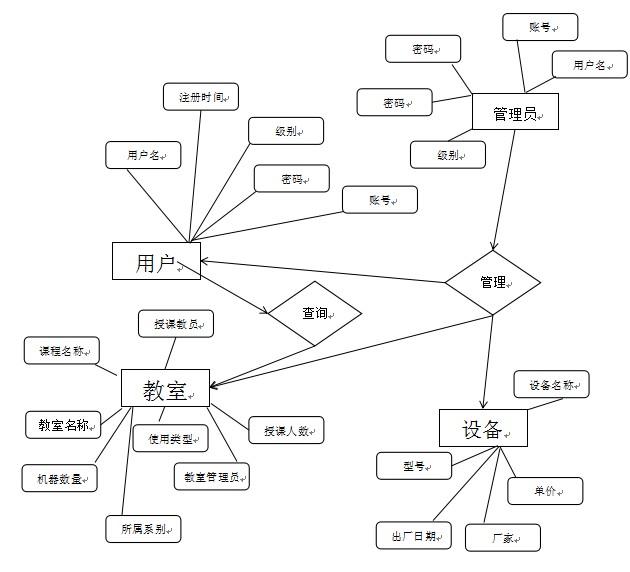


图4.4.1.1 系统E-R图

### 4.4.2数据库设计

在去年实习的那几个月中，我深深体会到了数据库设计的重要性，尽管SQL语句并不复杂，但是数据库的设计需要很多经验的积累，技术服务于业务，而公司的业务很大程度上就是依赖于数据库的设计，数据库设计是否合理，关系到日后开发工作是否能顺利进行以及产品上线后各项性能是否在可接受范围内。在我们的宠物医院管理系统的设计与实现过程中，数据库设计可以说是一个相当重要的环节。

根据前面所述的需求分析，为了今后宠物医院的业务能正常进行，本人设计的数据库一共设计了10张数据表，其中包括医院物品分类表，下面是本人设计的数据库中详细的数据表结构：

表4.4.2.1医院物品分类实体表（tb\_item\_cat）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 字段描述 | 主键 |
| id | number | 20 | 否 | 物品分类id | 是 |
| Parent\_id | number | 20 | 是 | 父类目id |  |
| name | varchar | 50 | 是 | 类目名称 |  |
| status | number | 1 | 否 | 类目状态 |  |
| sort\_order | number | 4 | 是 | 页面排列序号 |  |
| is\_parent | number | 1 | 否 | 是否为父类目 |  |
| created | date |  | 否 | 创建时间 |  |
| updated | date |  | 否 | 更新时间 |  |

表4.4.2.2 医院物品实体表（tb\_item）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 字段描述 | 主键 |
| id | number | 20 | 否 | 物品id | 是 |
| title | varchar | 100 | 否 | 物品名称 |  |
| supplier | varchar | 500 | 否 | 供应商 |  |
| price | number | 20 | 否 | 物品价格 |  |
| num | number | 10 | 否 | 库存数量 |  |
| barcode | varchar | 30 | 是 | 物品条形码 |  |
| image | varchar | 500 | 是 | 物品图片 |  |
| cid | number | 10 | 否 | 物品所属类目id |  |
| status | number | 4 | 否 | 物品状态 |  |
| created | date |  | 否 | 创建时间 |  |
| updated | date |  | 否 | 更新时间 |  |

表4.4.2.3 医院部门实体表（tb\_position\_cat）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 字段描述 | 主键 |
| id | number | 20 | 否 | 部门id | 是 |
| parent\_id | number | 20 | 是 | 该部门所属部门id |  |
| name | varchar | 50 | 否 | 部门名称 |  |
| status | number | 1 | 否 | 状态：是否可挂号 |  |
| sort\_order | number | 4 | 是 | 页面排列序号 |  |
| is\_parent | number | 1 | 否 | 是否为父部门 |  |
| created | date |  | 否 | 创建时间 |  |
| updated | date |  | 否 | 更新时间 |  |

表4.4.2.4 医院员工实体表（SheBei）

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 |
| SheBeiId | Int | 4 | 否 |
| SheBeiName | Varchar | 50 | 是 |
| XingHao | Varchar | 50 | 是 |
| GuiGe | Varchar | 50 | 是 |
| DanJia | Varchar | 50 | 是 |
| ChangJia | Varchar | 50 | 是 |
| ChuChanRiQi | Varchar | 50 | 是 |

表4.4.2.5 设备借还记录实体表（SheBeiLend）

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 |
| SheBeiLendId | int | 4 | 否 |
| LenTime | Varchar | 50 | 是 |
| SheBeiId | int | 4 | 是 |
| SheBeiName | Varchar | 50 | 是 |
| LenUserId | Int | 4 | 是 |
| LendUserName | Varchar | 50 | 是 |
| YuanYin | Varchar | 500 | 是 |
| AudiUser | varchar | 50 | 是 |
| Content | Varchar | 500 | 是 |
| ReturnTime | Varchar | 50 | 是 |

表4.4.2.6 设备出库记录实体表（SheBeiOut）

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 |
| SheBeiOutId | Int | 4 | 否 |
| SheBeiId | int | 4 | 是 |
| SheBeiName | Varchar | 50 | 是 |
| Varchar | Varchar | 50 | 是 |
| JiFangId | int | 4 | 是 |
| JiFangName | Varchar | 50 | 是 |

表4.4.2.7 设备维修记录实体表（SheBeiService）

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 |
| SheBeiServiceId | Int | 4 | 否 |
| SheBeiId | int | 4 | 是 |
| SheBeiName | Varchar | 50 | 是 |
| ServiceTime | Varchar | 50 | 是 |
| YuanYin | Varchar | 500 | 是 |
| YiJian | Varchar | 500 | 是 |
| ServiceUser | Varchar | 50 | 是 |
| Content | Varchar | 500 | 是 |

表4.4.2.8 设备库存记录实体表（SheBeiStorage）

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 |
| SheBeiStorageId | Int | 4 | 否 |
| SheBeiId | int | 4 | 是 |
| SheBeiName | Varchar | 50 | 是 |
| Num | int | 4 | 是 |
| Status | Varchar | 50 | 是 |

##### 4.4.2.1 数据库关系图

数据库关系图的作用是明确显示各个表之间的关系，还有各个表的结构。仔细说明各个表的主键设计和字段名、类型等信息。通过使用数据库关系表使整个数据库的每个表之间的关系一目了然，信息简洁明了。本数据库设计的关系图如下图所示4.4.2.1.1 数据库关系图

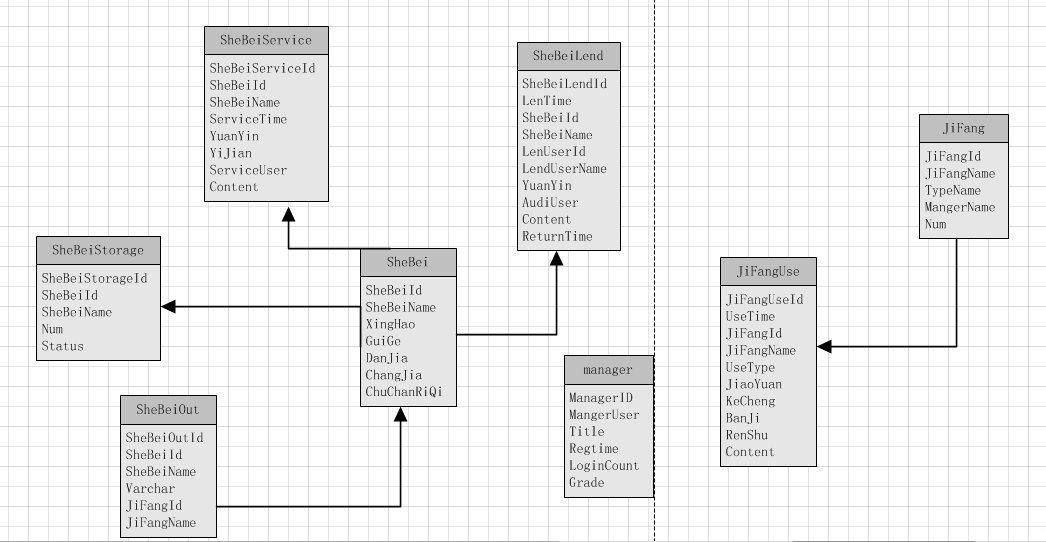


图4.4.2.1.1 数据库关系图

# 5系统的设计与实现

## 5.1 .NETS三层架构的技术划分

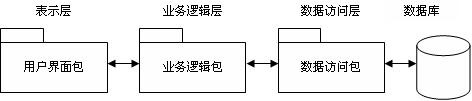


图5.1 .net三层结构技术划分

说明：三层架构(3-tier architecture) 通常意义上的三层架构就是将整个业务应用划分为：界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。区分层次的目的即为了“高内聚低耦合”的思想。在软件体系架构设计中，分层式结构是最常见，也是最重要的一种结构。微软推荐的分层式结构一般分为三层，从下至上分别为：数据访问层、业务逻辑层（又或称为领域层）、表示层。

## 5.2系统用户登陆注册模块设计与实现

系统用户需要登录之后才能对就是管理系统进行操作



图 5.2.1 用户登陆

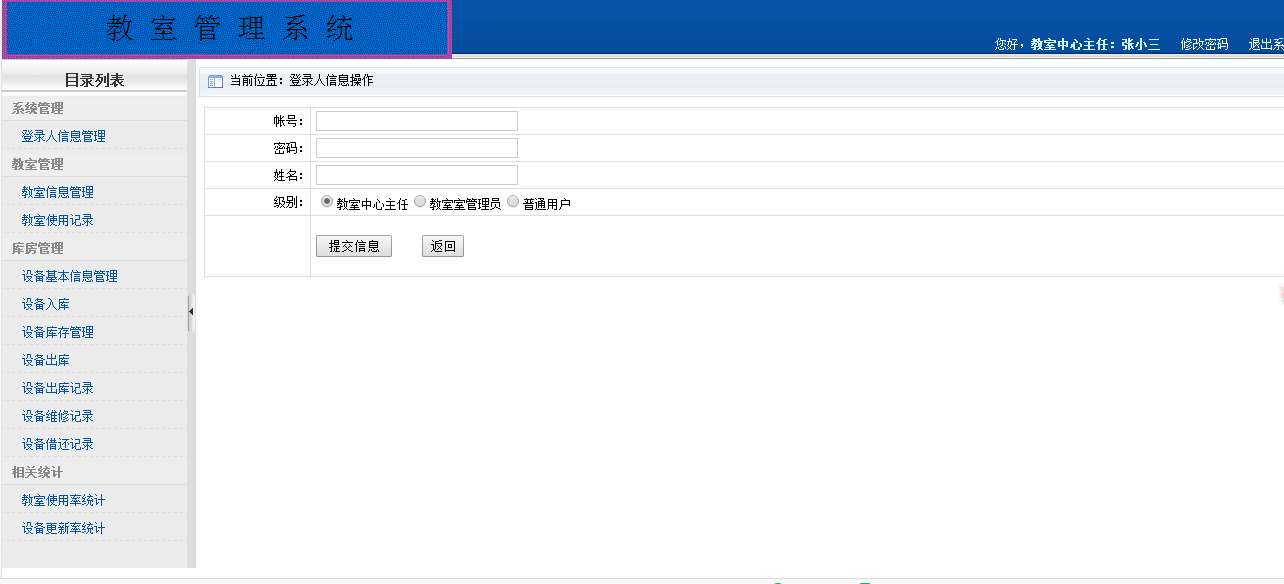


图5.2.2 系统用户注册

## 5.3 教室使用记录的实现

一个教室管理系统中作为核心功能就是教室信息的管理

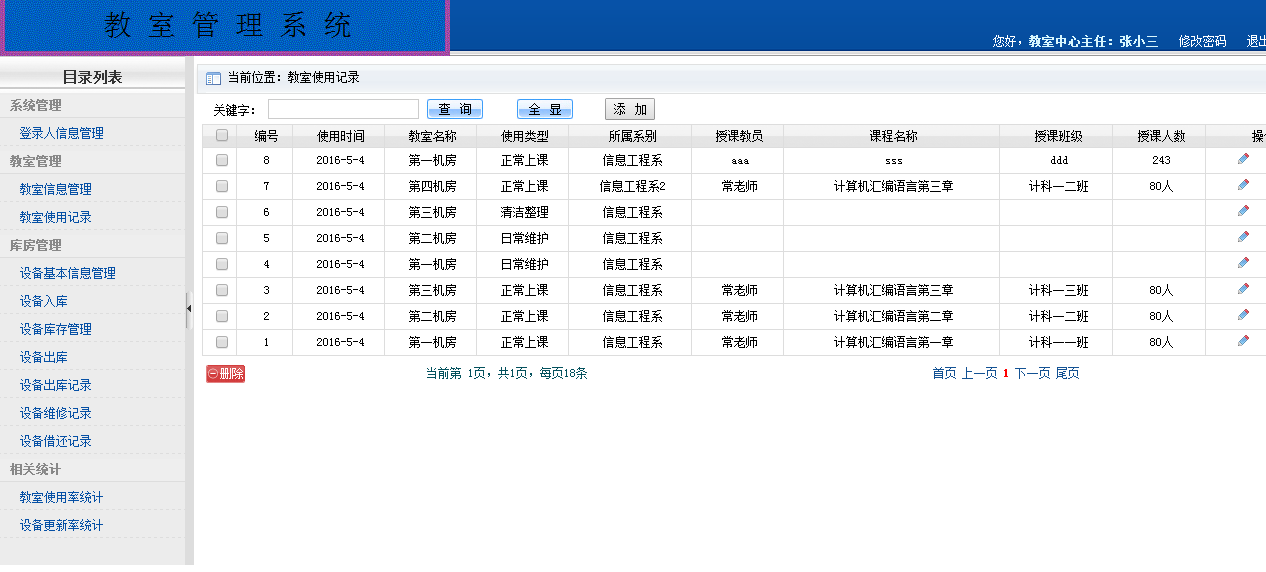


图5.3.1 教室使用记录

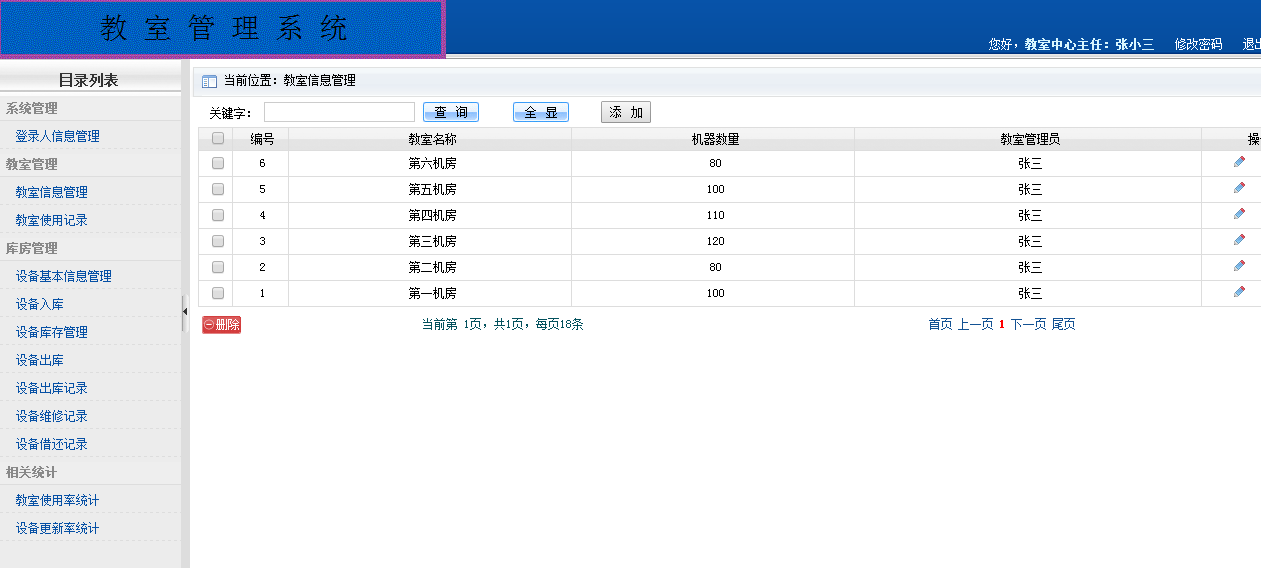


图5.3.2 教室信息管理

## 5.4 教室设备管理的设计与实现

每个教室都有附属的设备，教室设备管理的实现更完整地实现教室的管理



图 5.4.1 教室设备出、入库

## 5.5 设备维护的设计与实现

随着使用时间的推移，设备的维护管理能更好地统计设备的折旧情况



图5.5.1 设备维修记录



图5.5.2 设备借还记录



图5.6.1 库存管理

（2）管理员参考教室使用概率



图 5.6.2 教室使用率统计

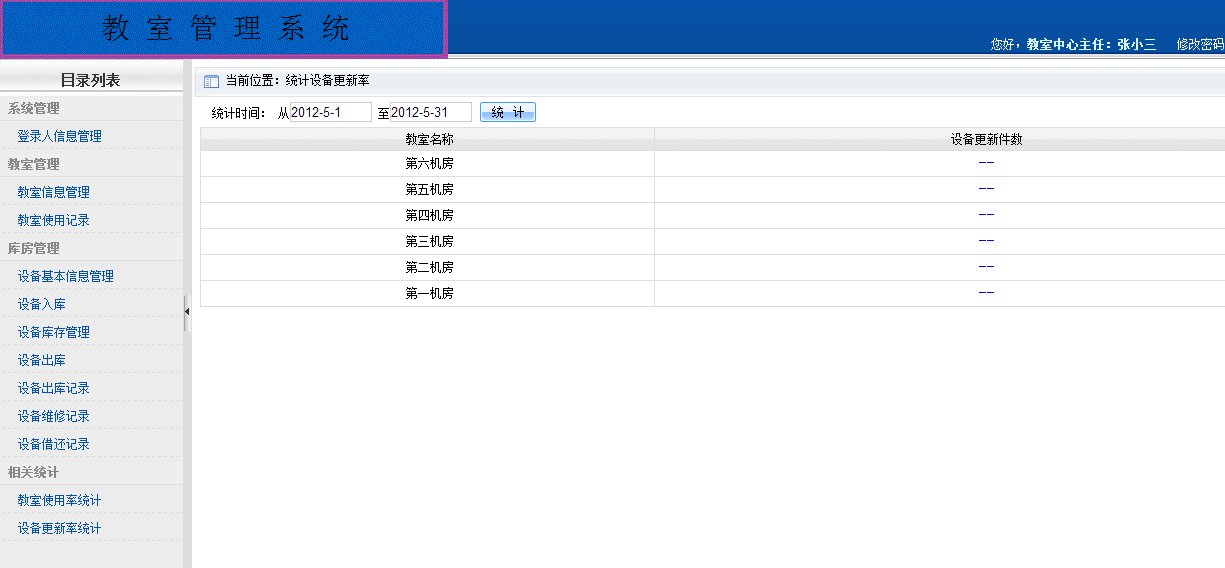


图 5.6.3 教室更新率统计

# 6安装与测试

## 6.1安装环境简介

本系统是由C#语言+SQL Server2008编写开发的。因此系统需要满足以下的运行环境。

（1）使用的操作系统为：本系统使用的操作系统为windows 7系统。

（2）使用的开发平台为：VisualStudio 2013。

（3）使用的数据库为：SQL Server2008。

## 6.2系统配置

（1）数据库连接设置：本系统使用的是.net的开源框架编写的，详细配置如下：

<session-factory> <property name="dialect"> //设置方言

org.hibernate.dialect.SybaseDialect

</property>

<property name="connection.url">//数据库URL设置

jdbc:jtds:sqlserver://127.0.0.1:1433/shopping</property>

//数据库的用户名设置

<property name="connection.username">sa</property>

//数据库的密码设置

<property name="connection.password"></property>

//数据库的驱动设置

<property name="connection.driver\_class">

net.sourceforge.jtds.jdbc.Driver

</property>

## 6.3 系统测试

测试编号：测试01

测试主题：系统用户注册

| **步骤名** | **描述** | **预期结果** | **执行结果** |
| --- | --- | --- | --- |
| 步骤 1 | 登陆系统进入登陆人信息管理 | 进入用户信息界面 | 进入用户信息界面 |
| 步骤2 | 点击添加按钮 | 进入用户注册界面 | 进入用户注册界面 |
| 步骤3 | “账号” 输入 “luo” | 输入框旁边提示输入正确 | 输入框旁边提示输入正确 |
| 步骤4 | “姓名”输入 “罗” | 输入框旁边提示输入正确 | 输入框旁边提示输入正确 |
| 步骤5 | “密码” 输入 “1236548” | 输入框旁边提示输入密码的字符长度在6~16位之间 | 输入框旁边提示输入密码的字符长度在6~16位之间 |
| 步骤6 | “密码” 输入 “123” | 输入框旁边提示输入正确 | 输入框旁边提示输入正确 |
| 步骤7 | “确认密码” 输入 “1234” | 输入框旁边提示两次密码输入不一致 | 输入框旁边提示两次密码输入不一致 |
| 步骤8 | “确认密码” 输入 “123” | 输入框旁边提示输入正确 | 输入框旁边提示输入正确 |
| 步骤9 | “级别”选择 “教室中心主任” | 输入框旁边提示选择正确 | 输入框旁边提示选择正确 |
| 步骤10 | 点击注册 | 提示注册成功 | 提示注册成功 |

测试编号：测试02

测试主题：系统用户登录

| **步骤名** | **描述** | **预期结果** | **执行结果** |
| --- | --- | --- | --- |
| 步骤 1 | 进入教室管理系统登陆界面 | 成功进入登陆界面 | 成功进入登陆界面 |
| 步骤2 | 账号名，密码为空，点击登录按钮 | 提示“账号或密码错误,请重新输入” | 提示“账号或密码错误,请重新输入” |
| 步骤3 | 输入正确账号名，密码为空，点击登录 | 提示“账号或密码错误,请重新输入” | 提示“账号或密码错误,请重新输入” |
| 步骤4 | 输入正确账号名，错误密码，点击登录 | 提示“账号或密码错误,请重新输入” | 提示“账号或密码错误,请重新输入” |
| 步骤5 | 输入错误账号名，错误密码，点击登录 | 提示“账号或密码错误,请重新输入” | 提示“账号或密码错误,请重新输入” |
| 步骤6 | 输入正确账号名，正确密码，点击登录 | 登录成功跳转到首页 | 登录成功跳转到首页 |

## 6.5总结

因为时间关系，系统还存在很多不足的地方，因为在前端的知识不是很熟悉所以系统的界面的操作性做的不太好，样式也比较落后，希望可以在以后的工作学习生活中不断学习，提高自己的专业能力。因为本系统还存在一些缺陷，总结还需要从下面几个方面进行改进：

1. 需要进一步的对系统进行测试，找出系统的缺陷；
2. 系统功能不是太完善，离正规的系统还有一定的差距，在以后系统的使用过程中去发现问题，根据需求添加新的功能对现有功能的不足进行完善，如设备管理，使用率统计管理等模块；
3. 系统的易用性不够好，需要再根据使用的使用习惯，界面的美化去逐步完善系统的界面和操作；
4. 对于数据库设计，没有考虑一些系统运行速率的问题，以后可以考虑使用数据库的存储过程来提高速度；

# 7结论

经|过几周的时间，毕业论文终于接近尾声了，从刚开始的时候不知从哪里入手到慢慢的熟悉，思路也逐渐清晰，完成整个论文的过程是很复杂的，完成这篇论文的时间是忙碌的，感觉时间过得也很快。想想这段时间，有很多感悟，

很多值得收藏的回忆与收获。随着毕业设计的逐渐完成，自己越是感觉，实践是自己最好的老师，实践可以将以前不能理解的理论知识，慢慢的掌握。

经过这次的毕业设计的编写，我对软件设计，数据库设计，编码，思维逻辑，软件测试等方面都加深了理解。我相信这对我以后的工作学习都是很有帮助的，毕业设计是严谨的，我们需要有认真严谨， 不断学习，脚踏实地的态度，不怕遇到难题，遇到难题不要轻易放弃，只要认真去发现，就一定会有收获，这也是我在这次 毕业设计过程中学到的感悟到的最好的收益。这既是一次对我四年来学习成果的测试，更是对我能力的一次提升。

在这次设计过程中，得到了很多人的帮助，这里有老师，有同学。使得我们的关系也更近了一步，学会听听别人对问题的见解。软件开发并不是一个人的工作，更是需要一个默契的团队的相互配合、协作。这点也是在我们整个大学生活老师一直在强调的团队精神。

# 参考文献

[1] 《WEB前段开发技术》储久良

[2] 《个体软件过程》Watts S.Humphrey

[3] 《JavaScript编程全解》井上诚一郎、土江拓郎.

[4] 《C#语言程序设计基础》 第二版 郑宇军

# 致谢

大学四年时光，眨眼之间就挥霍殆尽，犹记得那天满怀激情的那个年轻的自己出现在新生报到处，对人生未来的四年充满希冀。一点一滴的经历犹如在昨天发生，所幸已经流逝的青春也转化成踏入社会的一笔财富。

从开始的那一天算起，前前后后这个毕业设计已经花了一个月。在这个期间，指导老师章远尽心尽责，帮助我梳理主题和研究方向及需求。在完成设计期间涉及很多相关知识，在章指导的帮忙下最终解决问题。章指导凭借他务实的工作态度，严格要求学生完成设计。还有由衷感谢为我提供参考文献的作者，使我的作品添加上有力的依据。最后还要感谢我的家人，在我的成长路上一直相伴。

最后鉴于本人能力和知识水平的局限性，在论文作品中存在不足和亟需改正的地方，欢迎各位指教。

# 附录

绑定机房信息表源代码：

void Get\_Data()

{

SqlConnection cn = DB.OpenConnection();

SqlCommand cmd = new SqlCommand(sqltxt, cn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds, AspNetPager1.PageSize \* (AspNetPager1.CurrentPageIndex - 1), AspNetPager1.PageSize, "JiFang");

DataView dv = ds.Tables["JiFang"].DefaultView;

this.rptList.DataSource = dv;

this.rptList.DataBind();

}

分页源代码：

protected void AspNetPager1\_PageChanging(object src, Wuqi.Webdiyer.PageChangingEventArgs e)

{

AspNetPager1.CurrentPageIndex = e.NewPageIndex;

Get\_Data();

}

按机房名机房源代码：

protected void btnSearch\_Click(object sender, EventArgs e)

{

sqltxt = "select \* from JiFang where (JiFangName like '%" + txtKey.Text.Trim() + "%') order by JiFangId desc";

cmdtxt2 = "select count(0) from JiFang where (JiFangName like '%" + txtKey.Text.Trim() + "%')";

Get\_Pager\_Data();

}

复选框选择事件源代码：

protected void CheckBox1\_CheckedChanged(object sender, EventArgs e)

{

if (CheckBox1.Checked)

{

for (int i = 0; i < rptList.Items.Count; i++)

{

CheckBox chk = (CheckBox)rptList.Items[i].FindControl("CheckBox2");

if (chk != null)

{

chk.Checked = true;

}

}

}

else

{

for (int i = 0; i < rptList.Items.Count; i++)

{

CheckBox chk = (CheckBox)rptList.Items[i].FindControl("CheckBox2");

if (chk != null)

{

chk.Checked = false;

}

}

}

}