

**本科毕业设计（论文）**

**面向康复训练的随访数据管理系统**

**的设计与实现**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**年级班别 计科2班**

**学 号 3113005848**

**学生姓名 魏永倬**

**指导教师 杨卓**

**2017 年5月**

**面向康复训练的随访数据管理系统的设计与实现**

**李**

**计算机学院**

**摘 要**

随着计算机的快速发展和日益普及，各种软件系统遍及我们生活的每个角 落，为我们的学习和工作带来极大的方便。医院对神经损伤、骨骼损伤、肌肉损伤等这些需要康复训练的患者，会进行一些周期性的随访，了解患者的康复情况，并跟据这些随访信息的反馈，医生做出一些适当的康复训练调整，以便于让患者得到更好的康复效果。本设计主要针对上述信息设计出面向康复训练的随访数据管理系统。系统是用JSP制作的前端页面，SSH框架搭建的后台，MySql储存信息，以web的显示效果展示。系统中医生可以对患者进行随访和医疗，可以选择随访方式对患者进行随访，随访的时候还可以上传图片来得到一些随访的信息，患者可以评价医生的医疗效果，即时的更新医生的得分。有效地解决了康复训练的随访问题。

**关键字**：康复训练、随访、运动医疗

**Abstract**

With the rapid development of computers and the growing popularity of various software systems throughout every corner of our lives, it brings great convenience for our study and work.These patients require hospital rehabilitation nerve damage, bone damage, muscle injury, does some periodic follow-up, to understand the patient's rehabilitation and follow-up with them, according to the feedback information, physicians make some appropriate adjustments rehabilitation in order to enable patients to get better rehabilitation.This design is mainly designed for the above information for the rehabilitation of follow-up data management system. JSP system is made the front page, SSH framework set up backstage, MySql store information to display web display.System doctor can follow-up the patient and health care, you can choose the way of follow-up patients were followed up when you can also upload pictures to get some follow-up information, patients can evaluate the doctor's medical effects, real-time score updates physician.Effectively solve the problem of follow-up rehabilitation.

**Key words:** rehabilitation training, follow-up, sports medicine

**目 录**

[1 绪论 1](#_Toc13704)

[1.1题目背景及目的 1](#_Toc5097)

[1.2国内外研究前景 1](#_Toc10413)

[1.3论文构成及研究内容 2](#_Toc14680)

[2 可行性分析 3](#_Toc15967)

[2.1技术可行性分析 3](#_Toc24163)

[2.2经济可行性分析 3](#_Toc13433)

[2.3社会意义可行性分析 3](#_Toc1410)

[3 开发工具和技术 5](#_Toc30582)

[3.1 Eclipse 5](#_Toc5533)

[3.2 JSP 5](#_Toc13712)

[3.3 SSH框架 5](#_Toc9008)

[3.4 JSON和AJAX 6](#_Toc31699)

[3.5 Css+Div 6](#_Toc12179)

[3.6 JavaScript和jQuery 6](#_Toc13517)

[3.7 Mysql 7](#_Toc30562)

[4 需求分析 8](#_Toc24005)

[4.1功能分析 8](#_Toc29029)

[4.1.1医生 8](#_Toc11933)

[4.1.2病人 8](#_Toc26974)

[4.1.3系统管理员 8](#_Toc4554)

[4.2结构化分析 8](#_Toc16044)

[4.2.1系统流程图 8](#_Toc29958)

[4.2.2 数据流图 11](#_Toc19621)

[4.2.3数据字典 12](#_Toc175)

[4.2.4 E-R图 13](#_Toc29454)

[4.2.5 用例图 13](#_Toc29417)

[4.3面向对象分析 14](#_Toc10021)

[4.3.1面向医院的系统 14](#_Toc1273)

[4.3.2面向网站（医生）交流系统 14](#_Toc31248)

[5 总体设计 15](#_Toc13444)

[5.1计算机配置 15](#_Toc10160)

[5.1.1在电脑上配置java的运行环境JDK 15](#_Toc22404)

[5.1.2安装数据库MySql 15](#_Toc6444)

[5.1.3准备好开发工具Eclipse 或者My Eclipse 15](#_Toc22842)

[5.1.4下载web插件服务器，如tomcat等 15](#_Toc24749)

[5.1.5下载开发时要用到的jar包 15](#_Toc12138)

[5.2系统模块结构设计 15](#_Toc8228)

[5.2.1模块图 16](#_Toc21143)

[5.2.2各模块信息 16](#_Toc6926)

[5.3数据库设计 17](#_Toc6438)

[5.3.1 登录表 17](#_Toc29589)

[5.3.2警告信息表 17](#_Toc5263)

[5.3.3医生表 18](#_Toc21890)

[5.3.4病人表 18](#_Toc18682)

[5.3.5随访信息表 18](#_Toc9679)

[5.3.6评价表 19](#_Toc6918)

[5.3.7医疗表 19](#_Toc17387)

[5.3.8用户表 19](#_Toc27790)

[5.4设计模式 20](#_Toc4580)

[5.4.1 MVC设计模式 20](#_Toc26774)

[5.4.2 DAO设计模式 21](#_Toc25362)

[5.4.3 工厂模式 21](#_Toc16345)

[5.4.4单例模式 21](#_Toc11519)

[5.5数据库关系模型 21](#_Toc25228)

[6 详细设计 23](#_Toc29849)

[6.1功能模块的实现 23](#_Toc26948)

[6.1.1注册功能 23](#_Toc503)

[6.1.2登录功能 24](#_Toc11827)

[6.1.3个人信息修改功能 25](#_Toc30514)

[6.1.4求诊功能 27](#_Toc24845)

[6.1.5评价功能 28](#_Toc2852)

[6.1.6治疗功能 28](#_Toc29526)

[6.1.7随访功能 30](#_Toc28431)

[6.2 SSH设计思路 31](#_Toc26602)

[6.2.1 SSH代码块的实现流程 31](#_Toc24178)

[7 维护与测试 32](#_Toc10240)

[7.1系统曾出现的疑难杂症 32](#_Toc4762)

[7.2运行截图 32](#_Toc20720)

[结论 38](#_Toc24499)

[参考文献 39](#_Toc18822)

[致谢 41](#_Toc12133)

**1 绪论**

**1.1题目背景及目的**

**1.2国内外研究前景**

。

**1.3论文构成及研究内容**

**2 可行性分析**

**2.1技术可行性分析**

**2.2经济可行性分析**

**2.3社会意义可行性分析**

**3 开发工具和技术**

**3.1 Eclipse**

**3.2 JSP**

**3.3 SSH框架**

**3.4 JSON和AJAX**

**3.5 Css+Div**

**3.6 JavaScript和jQuery**

**3.7 MySql**

**4 需求分析**

**4.1功能分析**

4.1.1医生

4.1.2病人

4.1.3系统管理员。

**4.2结构化分析**。

4.2.1系统流程图

4.2.2 数据流图

4.2.3数据字典

4.2.4 E-R图

4.2.5 用例图

**4.3面向对象分析**

4.3.1面向医院的系统

医生不需要注册直接由管理员进行添加，而病人可以通过管理员添加和自己注册进入系统。作为一个医院的系统，主要是针对病人，让病人更加方便的了解医院的信息，对病人的服务要更加周到。一切以为病人为主，让病人对医院有一个满意的评价，吸引更多的病人前来治病。

4.3.2面向网站（医生）交流系统

医生和病人需要注册才能使用系统。这个系统对于医生和病人都是一个交流系统、目的在于为病人和医生服务。方便医生与病人之间的交互，足不出户也可以得到各地医生的治疗。他不局限于一个“点”，而是为了一个面。他是面向医院的升级，也是医疗康复系统的终极思想。

**5 总体设计**

**5.1计算机配置**

5.1.1在电脑上配置java的运行环境JDK

1、安装JDK

在官网上下载jdk1.8，选择指定的目录安装JDK。

2、配置环境变量

在Path 变量中编辑 %JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

在系统变量中新建CLASSPATH 变量填写 .;%JAVA\_HOME%\lib;

%JAVA\_HOME%\lib\tools.jar

5.1.2安装数据库MySql

5.1.3准备好开发工具Eclipse 或者My Eclipse

5.1.4下载web插件服务器，如tomcat等

5.1.5下载开发时要用到的jar包

1、所需的jar包如下，并且将jar包放入定义的web工程的WebContent\

WEB-INF\lib下

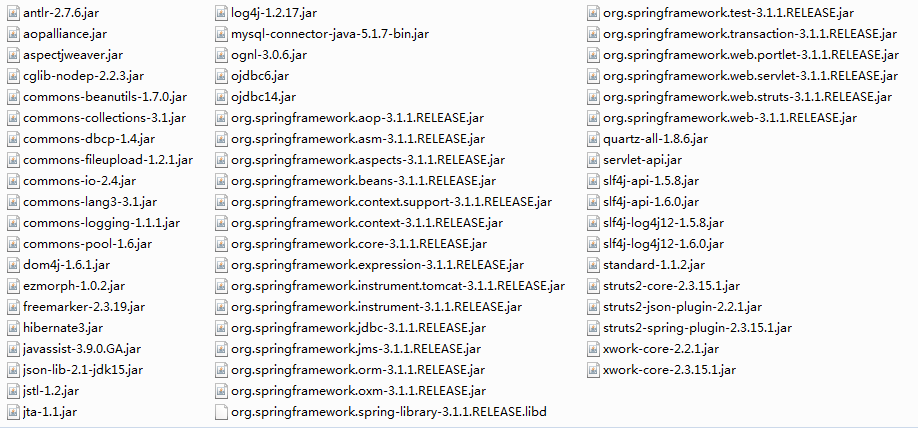


图5.1 web需要的jar包目录

**5.2系统模块结构设计**

5.2.1模块图

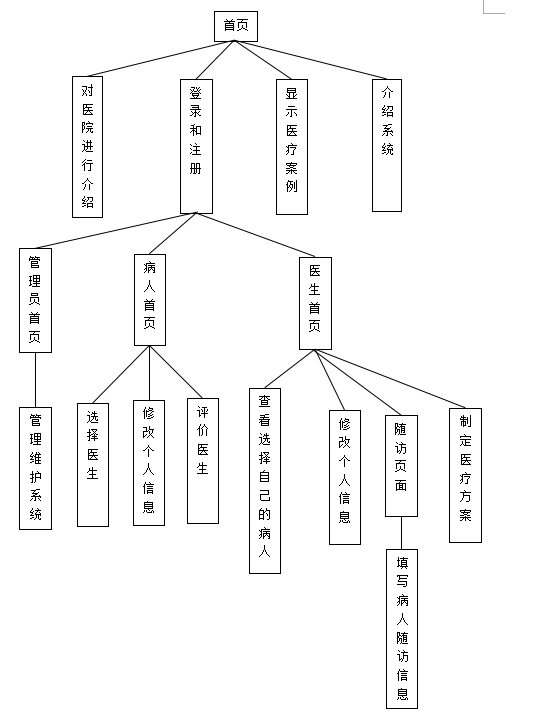


图5.2系统中的模块图

5.2.2各模块信息

1、主页

可以通过主页进入登录或者注册页面，此外，对系统进行介绍，显示医生的部分信息，这样吸引更多的病人使用系统，显示更多的治疗信息和康复训练信息让病人用的放心，增加病人对系统的信任度，增加医院的知名度。

2、 医生首页

医生可以修改个人信息，查看病人信息，为病人制定医疗方案，医生可以进行随访，去随访页面以不同的随访方式进行随访，医生查看管理员发布的警告信息。

1. 病人首页

病人可以修改个人信息，可以选择医生，可以为自己治疗的医生填写评价，反应医生的能力强弱。

1. 随访首页

主要是录入随访信息，以不同的方式得到随访信息。

1. 管理员首页

查看病人对医生评价，评分低对医生给于警告，维护系统，查看系统中成员的资料。

**5.3数据库设计**

5.3.1 登录表

**表5.1 登录表的信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_login | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 编号 | Int(11) | 主键 | 非空 |
| username | 用户名 | Varchar(20) | 唯一 | 非空 |
| password | 密码 | Varchar(20) | - | 非空 |
| logintime | 登录时间 | Date | - | 非空 |

5.3.2警告信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_docwarn | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 编号 | Int(11) | 主键 | 非空 |
| doctorid | 医生编号 | Int(11) | 外键 | 非空 |
| warnMsg | 警告信息 | Varchar(255) | **-** | 非空 |
| warnDate | 创建时间 | Date | - | 非空 |

**表5.8 警告信息表的信息**

5.3.3医生表

**表5.3 医生表的信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_doctor | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 编号 | Int(11) | 主键 | 非空 |
| userid | 用户表编号 | Int(11) | 外键 | 非空 |
| subject | 主治科目 | Varchar(20) | - | 非空 |
| describe | 医生简介 | Varchar(1000) | - | 非空 |
| grade | 评分 | Int(4) | - | 非空 |

5.3.4病人表

**表5.4 病人表的信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_patient | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 编号 | Int(11) | 主键 | 非空 |
| userid | 用户表编号 | Int(11) | 外键 | 非空 |
| illness | 病情 | Varchar(100) | - | 非空 |
| medhistory | 病史 | Varchar(600) | - | 非空 |

5.3.5随访信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_flupMessage | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 随访编号 | Int(11) | 主键 | 非空 |
| doctorid | 主治医生编号 | Int(11) | 外键 | 非空 |
| patientid | 病人编号 | Int(11) | 外键 | 非空 |
| message | 随访信息 | Varchar(2000) | - | 非空 |
| method | 随访方式 | Varchar(20) | - | 非空 |
| date | 随访时间 | Date | - | 非空 |

**表5.5 随访信息表的信息**

5.3.6评价表

**表5.6 评价表的信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_access | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 编号 | Int(11) | 主键 | 非空 |
| patientid | 病人编号 | Int(11) | 外键 | 非空 |
| doctorid | 医生编号 | Int(11) | 外键 | 非空 |
| content | 评价内容 | Varchar(1000) | - | 非空 |
| grade | 评价等级 | Int(4) | ➀ | 非空 |
| ➀评分为1-100之间 | | | | |

5.3.7医疗表

**表5.7 医疗表的信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_examinebill | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 编号 | Int(11) | 主键 | 非空 |
| doctorid | 医生编号 | Int(11) | 外键 | 非空 |
| patientid | 病人编号 | Int(11) | 外键 | 非空 |
| diagonse | 主题 | Varchar(255) | - | - |
| content | 治疗内容 | Varchar(255) | - | - |
| createDate | 创建时间 | Date | - | - |

5.3.8用户表

**表5.2 用户表的信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | kf\_user | | | |
| 列名 | 名称 | 数据类型 | 约束条件 | 空/非空 |
| id | 编号 | Int(11) | 主键 | 非空 |
| name | 姓名 | Varchar(20) | - | 非空 |
| sex | 性别 | Varchar(4) | ➀ | 非空 |
| age | 年龄 | Int(2) | - | 非空 |
| height | 身高 | Int(10) | - | 非空 |
| weight | 体重 | Int(10) | - | 非空 |
| wechat | 微信 | Varchar(20) | - | 非空 |
| phone | 电话号码 | Int(11) | - | 非空 |
| qq | QQ号码 | Int(15) | - | 非空 |
| email | 邮箱 | Varchar(40) | - | 非空 |
| address | 住址 | Varchar(50) | - | 非空 |
| IdCard | 身份证 | Varchar(30) | - | 非空 |
| loginid | 登录表id | Int(11) | 外键 | 非空 |
| registerdate | 注册时间 | Date | - | 非空 |
| kind | 用户类别 | Varchar(4) | ➁ | 非空 |
| ➀性别只有：“男”、“女”组成。  ➁用户类别只能有：“医生”,“ 病人”,“ 管理员”,“访客” 四种类型。 | | | | |

**5.4设计模式**

5.4.1 MVC设计模式

MVC全名是Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

M是指业务模型，V是指[用户界面](http://baike.baidu.com/view/362528.htm)，C则是控制器，使用MVC的目的是将M和V的实现代码分离，从而使同一个程序可以使用不同的表现形式。MVC被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

5.4.2 DAO设计模式

DAO设计模式是属于J2EE数据层的操作，使用DAO设计模式可以简化大量代码，增强程序的可移植性。DAO接口中定义了所有的用户操作，如添加记录、删除记录及查询记录等。不过因为是接口，所以仅仅是定义，需要子类实现。

利用框架的各种设计模式对程序运行时的“大脑”进行减压，可以帮助我们减少代码冗余，提高运行速度，便于维护，规范编程。

5.4.3 工厂模式

工厂模式是我们最常用的实例化对象模式，是用工厂方法代替new操作的一种模式。工厂模式在Java程序系统可以说是随处可见。因为工厂模式就相当于创建实例对象的new，我们经常要根据类Class生成实例对象，如A a=new A() 工厂模式也是用来创建实例对象的，所以以后new时就要多个心眼，是否可以考虑使用工厂模式，虽然这样做，可能多做一些工作，但会给你系统带来更大的代码可扩展性和尽量少的代码修改量。

5.4.4单例模式

单例模式就是确保某个类只有一个实例，并且这个=类自行实例化并向整个系统提供实例。在计算机系统中，线程池、缓存、日志对象、对话框、打印机、显卡的驱动程序对象等经常被设计成单例。这些应用都或多或少具有资源管理器的功能。每台计算机可以有若干个打印机，但只能有一个Printer Spooler，以避免两个打印作业同时输出到打印机中。每台计算机可以有若干通信端口，系统应当集中管理这些通信端口，以避免一个通信端口同时被两个请求同时调用。

总之，选择单例模式就是为了避免不一致状态，避免政出多头。

而单例模式常见的又分为饿汉式和懒汉式。饿汉式在类创建的同时就已经创建好一个静态的对象供系统使用，以后不再改变，所以天生是线程安全的。而懒汉式比较懒，只有当调用getInstance的时候，才回去初始化这个单例，所以懒汉式是线程不安全的。

**5.5数据库关系模型**

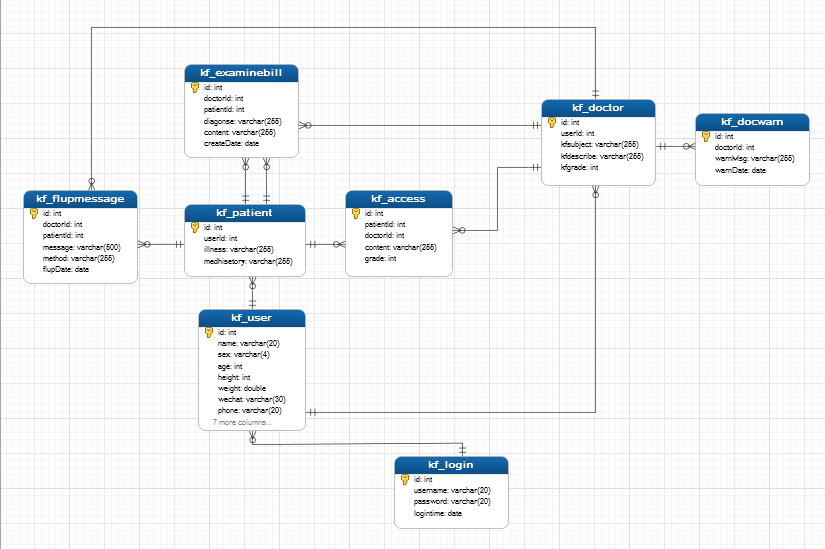


图5.3 数据库关联关系图

**6 详细设计**

**6.1功能模块的实现**

6.1.1注册功能

6.1.2登录功能

6.1.3个人信息修改功能

6.1.4求诊功能

6.1.5评价功能

6.1.6治疗功能

6.1.7随访功能

**6.2 SSH设计思路**

6.2.1 SSH代码块的实现流程

**7 维护与测试**

**7.1系统曾出现的疑难杂症**

**表**7.1 **系统出现的错误整理**

|  |  |  |
| --- | --- | --- |
| **问题** | **问题说明** | **解决程度** |
| 页面注册提交错误 | 用户在注册数据类型的填错，导致页面和后台无法交互。 | 已解决 |
| 登陆跳转过程中session混乱现象 | 不同的用户同时登录系统时出现用户名相同的现象，个人信息相同等现象。 | 已解决 |
| 页面兼容性问题 | 页面没经过兼容性处理，建议使用流行的浏览器访问，例如IE8+，chrome，opera等（兼容性问题还没有处理）。 | 没解决 |
| 数据库写入错误，页面返回信息提示，后台不会报错 | 修改信息的时候因为级联的关系，导致同时操作几张表是出现不同程度的错误。 | 已解决 |
| 退出后，系统回退刷新，session未清楚 | 账号已经退出的情况下，再次刷新页面session遗留，没有做到退出的效果 | 已解决 |
| Jar包版本问题，出现的后台内部错误 | 两个jar包，因为版本不同，出现后台混乱的错误，单个jar包支持的框架不一样。 | 已解决 |
| Tomcat因为不明原因会产生奔溃现象 | 本来tomcat是可用的，经过一段时间，发布进去的的工程无法运行。 | 已解决 |

**7.2运行截图**

**结论**

**参考文献**

1. 谷钰，杨艳斌，王泽生.Eclipse插件体系结构的研究[J].电脑知识与技术，2009-11-05：17-20.
2. 刘磊，朱敏，徐彦军.基于Struts框架的Java Web应用开发技术[J].计算机应用与软件，2007-05：1-10.

**致谢**

在此次毕业设计的过程中杨卓老师和战教授不辞幸苦的督促我们完成毕业设计，从选题到设计再到最后的成果老师时刻都在督促我们认真完成毕业设计，还有想法让我们参加几个比赛，但是由于我们的能力有限和付出的可能有点少辜负了老师，感到法非常的抱歉。但是我们还是非常的感谢指导老师为我们的付出，感谢老师不舍余力的对我们因材施教，在此表示由衷的谢意。也感谢同学们不厌其烦的帮助，感谢小组成员的帮助,他们帮助了我很多，曾经失落的时候鼓励过我，激励过我。我再次感谢再此过程中帮助过我的老师，同学。