# **摘 要**

随着计算机和网络技术的发展与普及，人们传统的医疗方式也迎来了其新一轮的变革，人们生活中使用纸质病历的限制越来越大，所以电子病历是一个很值得研究的课题。本系统的开发与实现，通过使用网络技术等新兴技术来实现将人们的健康记录以及诊疗结果转化为电子记录，解决传统纸质病历受到场地、使用和存储方面的限制。

本文详细的描述了如何采用JavaWeb技术开发一个实用的电子病历系统，实现用户个人信息以及病历信息的修改以及打印，后台管理等核心功能且实现以高级搜索、模糊搜索和智能建议等特色功能。本系统使用Java+MySQL进行开发，使用JavaScript和区块链技术等时下流行的前端技术实现网站的核心功能。

本论文从需求阶段开始，充分考虑到了用户对病历信息的实际需求，详细描述了从分析到实现的每一个步骤。最开始是系统前台，设计了系统的前台功能，注册登录以及修改个人信息等；然后设计了后台管理系统，用于管理系统中的各个模块所涉及的数据，对用户和系统进行管理；在系统开发完成后，进行了简单的黑盒测试并使用Junit测试工具进行单元测试，测试结果均达到预期的效果。本系统的设计和开发符合软件的开发流程和相关法规，并且能够满足用户对于一个物品交易网站的需要，具有一定的现实意义。

关键词： 电子病历；JavaWeb；MySQL；IntellijIDEA，区块链

**Title:** thedesign and implementation of electronic medical record system

# **ABSTRACT**

With the development and popularization of computer and network technology, people's traditional medical way has also ushered in a new round of changes, and people have more and more restrictions on using paper medical records in their lives. Therefore, electronic medical records are a topic worth studying. The development and implementation of this system, through the use of network technology and other emerging technologies to achieve people's health records and diagnosis and treatment results into electronic records, to solve the traditional paper medical records are limited by the location, use and storage.

This paper describes in detail how to use JavaWeb technology to develop a practical electronic medical record system, to achieve the user's personal information and medical record information modification and printing, background management and other core functions and to achieve advanced search, fuzzy search and intelligent advice features. This system USES Java+MySQL for development, using JavaScript and block chain technology and other popular front-end technology to achieve the core functions of the website.

Starting from the demand stage, this paper fully considers the actual needs of users for medical records information, and describes each step from analysis to implementation in detail. At the very beginning, I was the front desk of the system. I designed the functions of the front desk of the system, including registration, login and modification of personal information. Then the background management system is designed to manage the data involved in each module of the system and manage the users and the system. After the completion of system development, a simple black box test was conducted and Junit test tool was used for unit test. The design and development of the system conforms to the software development process and related regulations, and can meet the needs of users for a goods trading website, has certain practical significance.

**Key words:** electronic medical record; JavaWeb; MySQL; IntellijIDEA, blockchain

目录

[**摘 要** 1](#_Toc34558918)

[**ABSTRACT** 2](#_Toc34558919)

[第一章绪论 4](#_Toc34558920)

[1.1课题的来源及意义 4](#_Toc34558921)

[1.2国内外发展状况 6](#_Toc34558922)

[1.2.1 国内发展状况 6](#_Toc34558923)

[1.2.2国外发展状况 6](#_Toc34558924)

[1.3系统可行性研究 7](#_Toc34558925)

[1.3.1经济可行性 7](#_Toc34558926)

[1.3.2技术可行性 8](#_Toc34558927)

[1.3.3社会可行性（法律可行性） 9](#_Toc34558928)

[1.4本文的组织结构图 10](#_Toc34558929)

[1.5本章小结 10](#_Toc34558930)

[第二章 系统分析 11](#_Toc34558931)

[2.1系统业务需求分析 11](#_Toc34558932)

[2.2系统性能需求分析 12](#_Toc34558933)

[2.3系统安全性需求分析 12](#_Toc34558934)

[2.4系统运行环境需求分析 13](#_Toc34558935)

[2.5系统数据流图 14](#_Toc34558936)

[2.5.1系统顶层数据流图 15](#_Toc34558937)

[2.5.2一层数据流图 15](#_Toc34558938)

[2.6系统数据字典 16](#_Toc34558939)

[2.7系统的用例图及用例描述 17](#_Toc34558940)

[2.7.1用例图 17](#_Toc34558941)

[2.8本章小结 20](#_Toc34558942)

[第三章 系统设计 20](#_Toc34558943)

[3.1系统功能分析 20](#_Toc34558944)

[3.1.1系统功能概述 20](#_Toc34558945)

[3.1.2系统功能模块设计 21](#_Toc34558946)

[3.2系统功能模块图 22](#_Toc34558947)

[3.3系统主要功能模块流程图 22](#_Toc34558948)

[3.4系统类模型 27](#_Toc34558949)

[3.5数据库设计 28](#_Toc34558950)

[3.5.1数据库实体关系图（E-R图）设计 28](#_Toc34558951)

[3.5.2数据库总体关系设计图 32](#_Toc34558952)

[3.5.3数据库表设计 32](#_Toc34558953)

[3.6动态建模 35](#_Toc34558954)

[3.6.1账号管理设计时序图 35](#_Toc34558955)

[3.7本章小结 38](#_Toc34558956)

[第四章 系统实现 38](#_Toc34558957)

[4.1本系统开发过程中技术特色声明 38](#_Toc34558958)

[4.2主界面的设计与实现 39](#_Toc34558959)

[4.3前台子系统的设计与实现 39](#_Toc34558960)

[4.3.1用户登录模块的设计与实现 39](#_Toc34558961)

[4.3.2用户注册模块的设计与实现 40](#_Toc34558962)

[4.4管理员子系统的设计与实现 42](#_Toc34558963)

[4.4.1 管理员登录模块的设计与实现 42](#_Toc34558964)

[4.4.2管理系统主界面的设计与实现 42](#_Toc34558965)

[4.5特色功能的设计与实现 43](#_Toc34558966)

[4.5.1高级搜索功能 43](#_Toc34558967)

[4.5.2随访管理功能 44](#_Toc34558968)

[4.6本章小结 44](#_Toc34558969)

[第五章 系统测试 45](#_Toc34558970)

[5.1测试的目的 46](#_Toc34558971)

[5.2测试的方法 46](#_Toc34558972)

[5.2.1黑盒测试 46](#_Toc34558973)

[5.2.2单元测试 46](#_Toc34558974)

[5.3系统测试 47](#_Toc34558975)

[5.4测试结论 47](#_Toc34558976)

[5.5本章小结 48](#_Toc34558977)

第一章绪论：

## 1.1课题的来源及意义

电子病历(EMR,Electronic Medical Record)，作为一种重要的临床信息资源，已越来越受到现代医疗机构重视，是其高效、优质的开展临床诊疗、医学科研以及医疗管理工作的必需，同时也是居民健康档案的主要信息来源，是临床信息系统的核心及标志 [1 。近几年电子病历受到空前的关注，很多地方开始争先恐后的建设以电子病历为核心的卫生信息平台，在我国的十二五卫生信息化建设工程规划中，有明文指示，健康档案数据库和电子病历数据库的建立是我国建设区域化医疗平台的基础。2011 年医政司制定了电子病历的基本规范，并在全国范围内 110 家医院开展电子病历的试点工作，目前EMR软件正处于成长期。

电子病历是新医改的试点内容之一，它包含了首页，病历记录，检查检验结果，手术记录，护理记录等，它不仅指静态病历信息，还包括提供相应的服务。是以电子化管理的有关个人终生健康和医疗保健行为的信息，涉及病人信息的采集，运输，传输，处理和利用的所有过程信息。它是实现现代化医院管理目标的重要措施，对完善医院管理模式有这重要影响和深远意义。电子病历的有效应用会促进区域医疗信息的交换和共享，也会便利广大患者和医生，对有效利用医疗资源，进一步提高医疗质量，保障医疗安全，为人民群众提供安全、有效、方便、价廉的医疗服务，促进社会和谐有重大的意义。

## 1.2国内外发展状况

### 1.2.1 国内发展状况

在我国香港，台湾地区，电子病历应用较为普遍，香港卫生局动用8亿给香港医院管理局提升病历系统；2012年底台湾地区70％的诊所何80％的医院建立起电子病历系统。大陆自1995年提出“金卫工程”，各地陆续开展了电子病历的研究工作。国家卫生部也相继出台了有关《电子病历基本规范》，《电子病历功能规范》等规范和通知文件，加快了电子病历的推广应用。但是在运行过程中并不顺利，尚存在一些问题。

首先是经费不足，系统无法投入大量经费，没有足够的经费保障，根本无法开展或进一步深层次挖掘功能。还有缺乏统一的技术标准，我国在电子病历缺乏统筹性，没有出台一套在框架结构，技术标准，安全维护等方面的标准，使研发者的工作难以继续。

### 1.2.2国外发展状况

1960年美国就有了电子病历概念，主要是自动化门诊病案系统，1992年发展到电子化健康记录HER（eelectronic health record,EHR）,并把HL7（health level seven）作为国家健康信息体系，推广到电子病历，到了21世纪，世界鸽过对电子病历已经相当重视，各国政府都开始对电子病历进行深入研究以及大力推广，比如美国就投入了近200亿美元用于加强建设现代化临床电子病历，英国，加拿大等成立了像NHS（national health service）的研发机构，2006年，日本实现了在60％的400张床以上的医院和60％的诊所实行无纸化电子病历。此外，英国，德国，荷兰，丹麦，芬兰等欧洲国家在电子病历应用的广度和慎独上均有独到之处，日常使用电子病历的比率高达95％以上。

## 1.3系统可行性研究

### 1.3.1经济可行性

首先，在本系统的开发过程中所使用的开发工具如IDEA和MYSQL都是开源免费软件，所使用的开发环境也是简单方便，这在开发过程中节省了很大的空间时间和经济成本。其次，这个系统对于硬件的要求不是很高，所以，在正常的服务器上都可以运行，由于JAVA语言有着跨平台，高适应性的特性，所以运行系统的服务器可以选择linux操作系统，这会节省我们的时间和费用。

根据以上所述，本系统的开发和实现在经济方面上可行没有问题。

### 1.3.2技术可行性

首先，本系统使用了免费集成开发工具IDEA进行开发，IDEA全程IntelliJ IDEA,是JAVA变成语言开发的集成环境，IDEA在业界被公认为最好的java开发工具，尤其在智能代码助手，代码自动提示，重构，J2EE支持，各类版本工具，CVS正和，代码分析，创新的GUI设计等方面的功能。在语言方面，java是一门面向对象的编程语言，具有功能强大和简单易用两个特征。Java语言作为静态面向变成语言的代表，极好的实现了面向对象理论，在android，金融，网站开发，嵌入式领域，大数据技术，科学应用种，java都展示出了它独到的一面，所以本系统使用java进行开发。

在数据库方面选用了MYSQL，MYSQL是一种开源的关系型数据库管理系统，使用了最常用的数据库管理语言SQL进行数据库管理。因为其开源性，运行速度，可靠性，适应性，所以本系统使用了MYSQL进行开发

### 1.3.3社会可行性（法律可行性）

“电子病历系统“的开发，会带给一定的群体或个体用户带来很大的方便，社会实用性强，网站在开发过程种未侵害和损害其他企业和个人以及国家的权益和所有权，本系统在开发过程中使用的团建符合开源许可证要求。

根据以上所述，本系统的开发和实现在法律上切实可行。

## 1.4本文的组织结构图

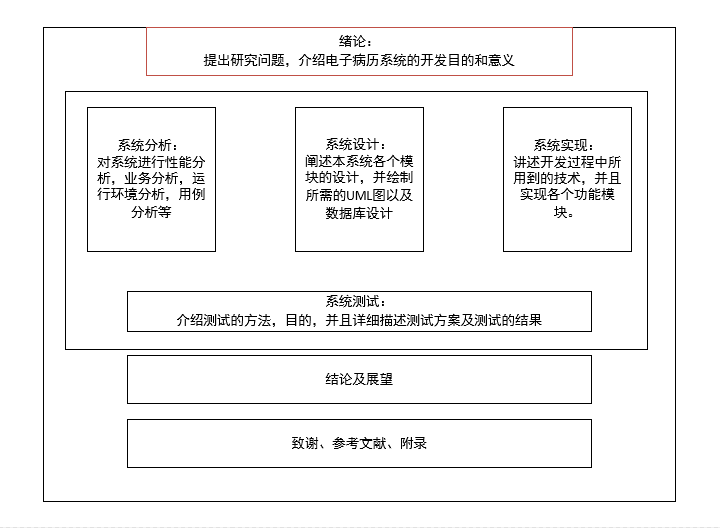


图1.1组织结构图

## 1.5本章小结

本章节种，首先描述了“电子病历”这个课题的来源以及意义，并且讲述了国内外的前景和应用，又从经济可行性，技术可行性，社会可行性三个方面进行可行性分析。在进行本章任务时，查阅了相关的文献和资料，作为基础部分，也为后续的设计和实现营造了良好的基础。

# 第二章 系统分析

## 2.1系统业务需求分析

“电子病历系统”所包含的基本业务有：

1. 前台系统包括注册和登录模块，个人信息界面等。
2. 用户分为普通用户，医生用户，管理员用户。
3. 管理员用户可以使用后台管理系统，可以对病历和用户进行管理
4. 普通用户有基本的访问权限，可以浏览打印病历，修改个人信息。医生有更高的访问权限，可以浏览打印病历，修改自己和普通用户的个人信息，管理员有最高权限。
5. 系统后台面向管理员用户，包含用户管理，物品管理等，实现基本的增删改查功能。
6. 病历可以根据普通用户的指标或信息对其进行智能建议
7. 医生用户可以对病历进行标注，区分传染性和非传染性疾病。
8. 管理员可以上传并且整合病历至数据库，方便组成病历联盟链供其他医院使用病历。
9. 系统前台面向医生和普通用户，分为两大模块，用户注册填写信息模块，病历信息完善模块。

## 2.2系统性能需求分析

根据上述的业务需求，并且结合实际情况和可能出现的性能需求，对所欲实现的系统在性能上提出了以下几个方面的要求：

1. 精度

精度值得是系统的显示精度，如系统种出现的数字（如血压，体温等）要完整的显示正确的位数和单位。

1. 时间特征要求
2. 系统正常响应时间不能大于200ms；
3. 更新处理时间应小于300ms；
4. 数据的上传和下载时间应该小于1s；
5. 数据库应当优化提高病历的SQL操作效率

## 2.3系统安全性需求分析

1)系统级安全：采用了访问ip段的限制，登录次数限制等

2)功能性安全：用户在操作病历时需要审核，上传附件不能超过3m等

3)数据域安全：用户只能访问自己的病历记录，医生只能访问病历的部分字段

4)数据库安全：此系统采用了联盟链技术，极大的保证了数据库数据的安全性以及共享信息安全性。

## 2.4系统运行环境需求分析

系统运行时对于环境的要求有：

1. 硬件环境最低需求
2. 服务器:
3. 处理器：1.5GHz或更快
4. 内存：2GB或以上
5. 硬盘空间：20GB或以上
6. 接入互联网
7. 客户端：
8. 处理器：1.5GHz或更快
9. 内存：2GB或以上
10. 显示器分辨率在800\*600以上
11. 键鼠等输入设备
12. 服务器软件环境需求：
13. 操作版本：win7以上版本或2.6以上内核发行版linux
14. JDK版本：JDK1.8
15. 数据库版本：MYSQL 5.7

## 2.5系统数据流图

数据流图是[结构化分析方法](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E5%88%86%E6%9E%90%E6%96%B9%E6%B3%95)中使用的工具,它以图形的方式描绘数据在系统中流动和处理的过程，由于它只反映系统必须完成的逻辑功能，所以它是一种[功能模型](https://baike.baidu.com/item/%E5%8A%9F%E8%83%BD%E6%A8%A1%E5%9E%8B)。在结构化开发方法中，数据流图是需求分析阶段产生的结果。

[数据流程图](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%B5%81%E7%A8%8B%E5%9B%BE)包括：

a．指明数据存在的数据符号，这些数据符号也可指明该数据所使用的媒体；

b．指明对数据执行的处理的处理符号，这些符号也可指明该处理所用到的机器功能；

c．指明几个处理和（或）数据媒体之间的数据流的流线符号；

d．便于读、写数据流程图的特殊符号。

### 2.5.1系统顶层数据流图

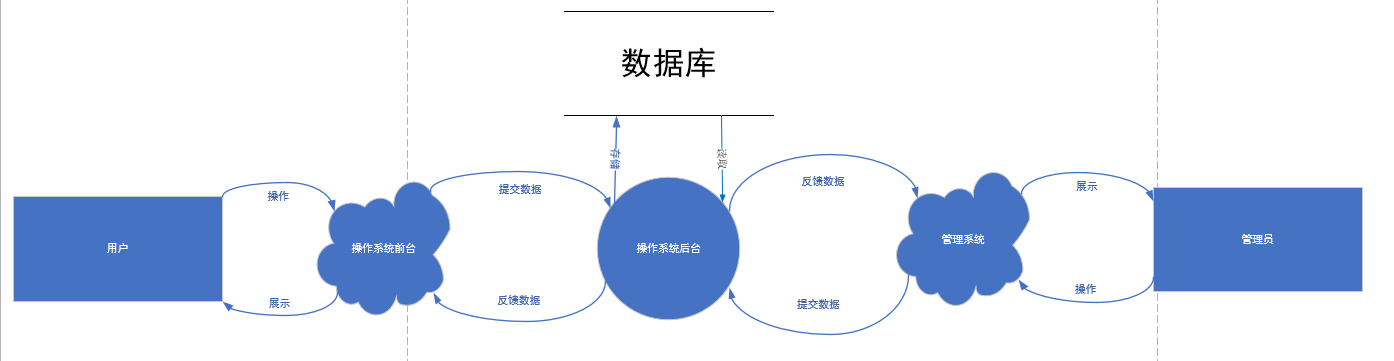
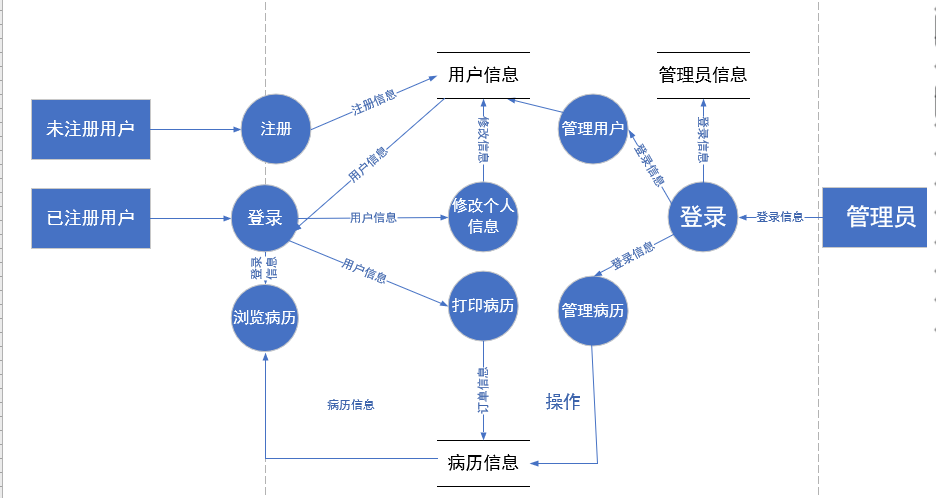


图2-1顶层数据流图

### 2.5.2一层数据流图

（详细操作和关系的数据流图）（例）



## 2.6系统数据字典

表2.6.1数据项

表2-1 数据项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名 | 数据项含义 | 别名 | 数据类型 | 数值含义 |
| Medical record information | 系统中的病历信息 | 病历详情 | varchar | 可变长度字符串 |
| author\_id | 系统中的所属用户编号 | 用户编号 | int | 用户编号 |
| create\_time | 数据项的创建时间 | 创建时间 | datetime | 年月日T时分秒 |
| IS\_User Registration | 用户是否注册 | 注册信息 | int | 0未注册 1注册 |
| IS\_AVALIBLE | 用户是否可用 | 用户状态 | int | 1正常 2冻结 |
| PIC\_IMG | 用户头像存储路径 | 用户头像 | varchar | 路径字符串 |
| order\_status | 病历状态 | 病历状态 | int | 0查询中，1未在查询 |
| complete\_time | 病历填写时间 | 完成时间 | datetime | 年月日T时分秒 |
| User profile | 系统中的所属用户的特指个人信息 | 用户个人信息 | varchar | 可变长度字符串 |
| Administrator id | 系统中的管理员编号 | 管理员编号 | int | 管理员编号 |

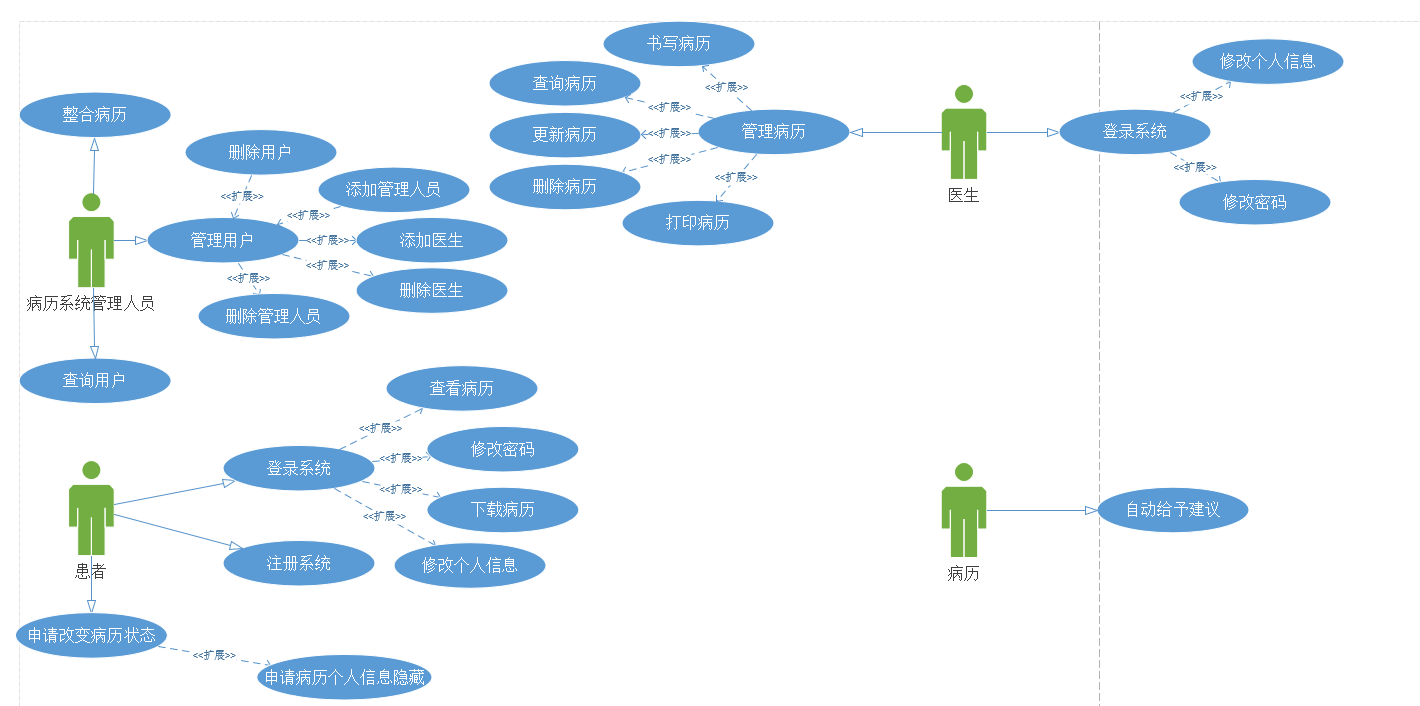
表2.6.2数据结构

表2-2 数据结构

|  |  |  |
| --- | --- | --- |
| 数据结构名称 | 含义说明 | 组成 |
| 登陆用户 | 通过浏览器访问系统的已注册的人 | 用户名、密码、注册时间等可用用户信息 |
| 管理员 | 可以使用系统后台管理系统的用户 | 用户名、密码、权限 |
| 病历信息 | 医生用户填写的普通用户的有关信息 | 病史，血压，肺活量，视力，身高，体重等有关信息 |

## 2.7系统的用例图及用例描述

### 2.7.1用例图

2.7.2用例描述

1. 用户注册

当用户访问本网站时可以注册，注册的用例描述如表2-3所示。

表2-3 注册用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户注册 | |
| 用例ID | YH\_001 | |
| 参与者 | 用户 | |
| 描述 | 用户注册系统过程 | |
| 优先级 | 较高 | |
| 前置条件 | 用户需在未注册状态 | |
| 后置条件 | 用户注册信息保存到数据库 | |
| 主事件流 | | |
| 用户 | | 系统 |
| 1．进入或转入用户注册画面 | |  |
| 2．用户输入用户名，密码，邮箱等信息 | |  |
|  | | 1. 系统检查用户名是否存在 |
|  | | 4. 系统检查两次输入的密码是否相同 |
|  | | 1. 提示注册成功，转向用户界面 |
| 异常流 | | |
| 异常流 | | 系统中已经存在此用户名 |
| 系统 | | 给出提示信息“此用户名存在” |
| 异常流 | | 用户名格式不合法 |
| 系统 | | 给出提示信息“用户名格式错误” |
| 异常流 | | 两次输入的密码不相同 |
| 系统 | | 给出提示信息“两次输入的密码不相同” |
| 异常流 | | 验证码输入错误 |
| 系统 | | 给出提示信息“验证码错误” |

2）用户登录

注册完用户之后可以通过邮箱，密码登录，登录之后可以享受用户的对应权限，如表2-4所示。

表2-4 登录用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户登录 | |
| 用例ID | YH\_002 | |
| 参与者 | 用户 | |
| 描述 | 用户登录系统过程 | |
| 优先级 | 较高 | |
| 前置条件 | 用户需在未登录状态 | |
| 后置条件 | 将用户信息放入会话中 | |
| 主事件流 | | |
| 用户 | | 系统 |
| 1．进入或转入用户登录画面 | |  |
| 2．用户输入用户名和密码 | |  |
|  | | 3．系统检查用户名，密码是否正确 |
|  | | 4．读取用户的基本信息放入会话中 |
| 异常流 | | |
| 异常流 | | 系统中不存在该用户名 |
| 系统 | | 给出提示信息“请输入正确的用户名” |
| 异常流 | | 用户名或密码错误 |
| 系统 | | 给出提示信息“用户名或密码错误” |

3）用户管理

用户管理包括用户查询，用户添加。用户查询可以通过邮箱、用户名、昵称、性别，ID，身份证等条件进行不同程度的查询，用户管理用例描述如表2-5所示。

表2-5 用户管理用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户管理 | |
| 用例ID | YH\_\_003 | |
| 参与者 | 管理员 | |
| 描述 | 管理员进行用户管理 | |
| 优先级 | 较高 | |
| 前置条件 | 管理员进入用户管理界面 | |
| 后置条件 | 管理用户功能不存在问题 | |
| 主事件流 | | |
| 管理员 | | 系统 |
| 1．单击用户管理链接 | |  |
|  | | 2．显示用户管理主画面，下拉选框中包括用户查询，用户添加，用户删除等功能 |
| 可选事件流 | | |
| 用户查询 | | |
| 管理员 | | 系统 |
| 1．输入邮箱，点击查询 | | 2．显示符合条件的用户及用户信息 |
| 3．输入电话，点击查询 | | 4．显示符合条件的用户及用户信息 |
| 5．输入用户名，点击查询 | | 6．显示符合条件的用户及用户信息 |
| 1. 输入昵称，点击查询 | | 8.显示符合条件的用户及用户信息 |
| 9.输入性别，点击查询 | | 10.显示符合条件的用户及用户信息 |
| 11.输入ID，点击查询 | | 12. .显示符合条件的用户及用户信息 |
| 13.输入身份证，点击查询 | | 14. .显示符合条件的用户及用户信息 |
| 用户添加 | | |
| 15.输入添加用户的基本信息，点击提交 | | 16.添加成功 |
| 用户删除 | | |
| 17.输入用户的ID及用户名，点击删除 | | 18.删除成功 |

4）病历管理

病历包括病历查询，病历删除。病历查询可以对用户名，用户性别、ID，用户身份证及时间进行不同程度的查询，病历管理用例描述如表2-6所示。

表2-6 公告管理用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 病历管理 | |
| 用例ID | BL\_001 | |
| 参与者 | 管理员 | |
| 描述 | 管理员进行病历管理 | |
| 优先级 | 高 | |
| 前置条件 | 管理员进入病历管理界面 | |
| 后置条件 | 管理病历功能没有问题 | |
| 主事件流 | | |
| 管理员 | | 系统 |
| 1．单击病历管理按钮 | |  |
|  | | 2．显示病历管理主画面，包括病历查询和删除功能 |
| 可选事件流 | | |
| 查询病历 | | |
| 管理员 | | 系统 |
| 1．输入用户名，点击查询 | | 2．显示符合条件的病历及病历详情 |
| 3．输入ID，点击查询 | | 4．显示符合条件的病历及病历详情 |
| 5．输入时间，点击查询 | | 6．显示符合条件的病历及病历详情 |
| 7.输入性别，点击查询 | | 8. 显示符合条件的病历及病历详情 |
| 删除病历 | | |
| 7.点击删除病历按钮 | | 8.提示删除成功 |

## 2.8本章小结

本章的主要内容是对系统的需求分析，介绍了系统完整体验所需的硬件、软件以及安全性要求。展示了系统的基本数据流图和一层数据流图，编写了数据字典，介绍了系统在开发和使用过程中的部分数据项。最后展示了系统的用例图，介绍系统的用例关系，详细进行了用例描述，为后续的系统设计做好详细的准备。

# 第三章 系统设计

## 3.1系统功能分析

### 3.1.1系统功能概述

根据系统的需求分析，结合实际情况具体细化，本网站要实现的功能主要有：

1）用户可以通过网站前台实现注册和登录；

2）普通用户通过浏览器登录后可以浏览系统中自己的病历信息；

3）普通用户通过浏览器登录后可以修改个人信息；

4）普通用户通过浏览器登录后可以打印病历；

5）医生用户通过浏览器登录后可以浏览、查询、修改用户病历信息；

6）医生用户通过浏览器登录后可以打印病历；

7）管理员用户可以通过管理系统对用户进行添加，删除，查询，浏览；

8）用户可以申请隐藏自己的部分个人信息，以保证人身信息安全；

9）网站建有用户个人页面，页面包含用户详细信息、修改及打印记录等；

10）网站有检索功能，可以实现模糊查询；

11）网站拥有后台管理系统供管理员用户使用；

12）管理员用户可以通过管理系统对病历进行查询，删除，浏览，整合；

12）管理系统可以直观的展示网站的统计信息等数据供管理员查看；

13）医生用户可以对传染性疾病进行标注；

14）病历系统可以通过对各项指标的数值来对普通用户提出智能建议；

针对上述功能，可以将网站划分为2个子系统，其中共有6个主要模块。子系统包括前台用户子系统和后台管理子系统。其中前台用户子系统含有注册登录模块、用户个人信息模块、用户病历信息模块；后台管理子系统包括登录模块、用户管理模块、病历信息管理模块。

前台主要模块为病历信息模块和用户个人信息模块，用户个人模块是用户使用网站时产生的数据的汇总和展示模块，可供用户对个人信息进行查看和修改，并实现打印病历和记录操作历史的功能。

### 3.1.2系统功能模块设计

针对上述功能，可以将网站划分为2个子系统，其中共有7个主要模块。子系统包括前台用户子系统和后台管理子系统。其中前台用户子系统含有注册模块、登录模块、用户个人信息操作模块、用户病历信息操作模块；后台管理子系统包括登录模块、用户管理模块、病历信息管理模块。

前台主要模块为病历信息模块和用户个人信息模块，用户个人模块是用户使用网站时产生的数据的汇总和展示模块，可供用户对个人信息进行查看和修改，并实现打印病历和记录操作历史的功能。

## 3.2系统功能模块图

基于以上的功能模块设计分析，本系统包括的功能模块结构图如下：

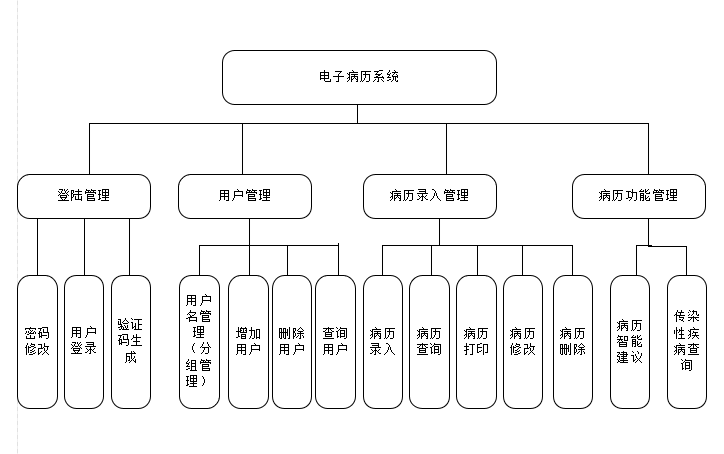


图3-1 病历管理系统功能模块结构图

## 3.3系统主要功能模块流程图

1）注册模块

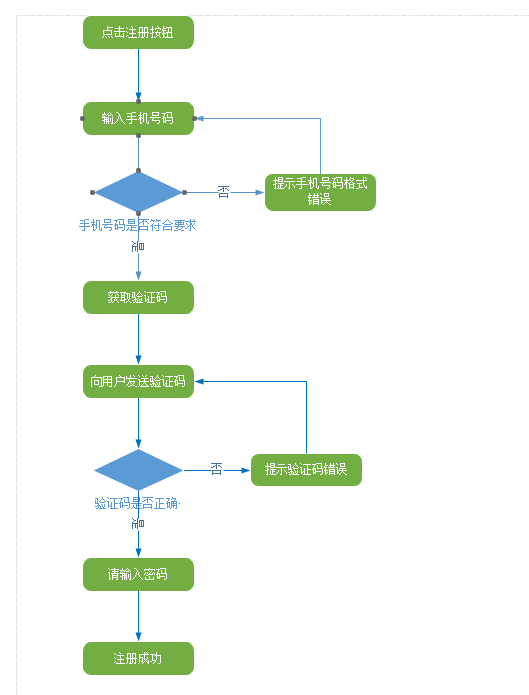


图3-2 用户注册流程图

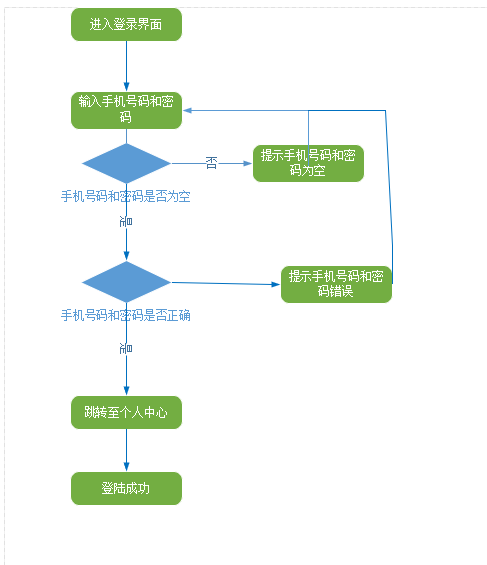
1. 登录模块

图3-3 用户登录流程图

1. 用户个人信息操作模块

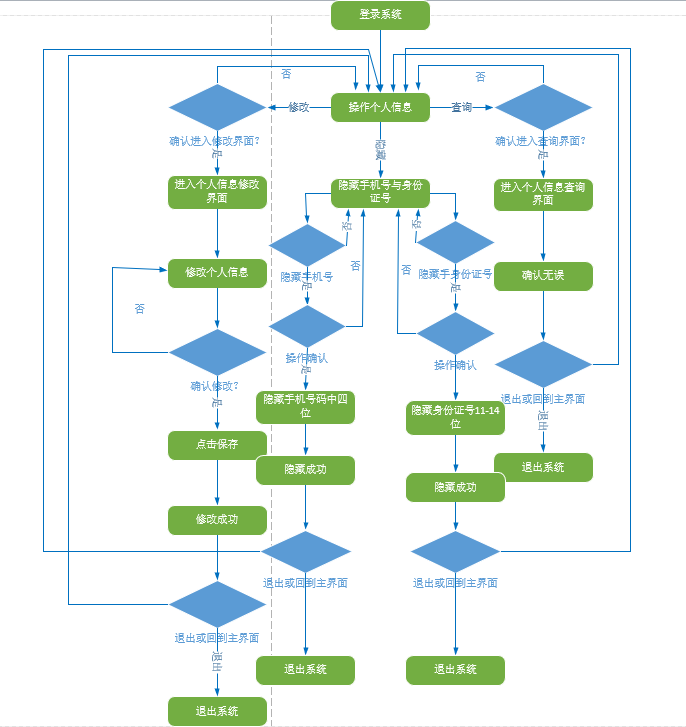


图3-4 用户个人信息操作流程图

1. 用户病历信息操作模块

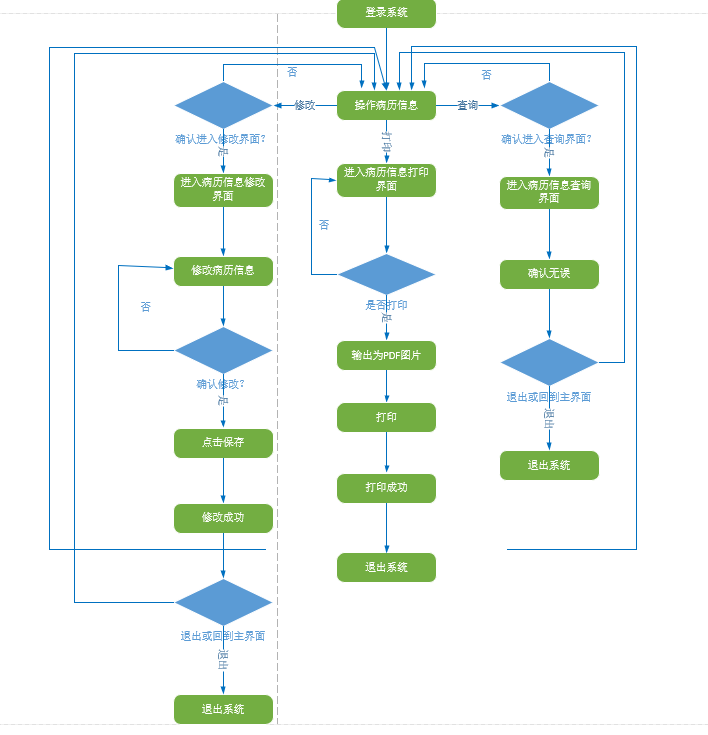


图3-5 用户病历信息操作流程图

1. 管理员子系统操作流程图

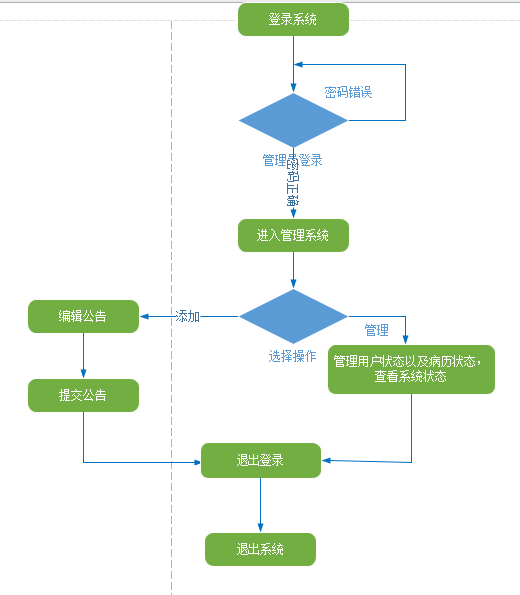
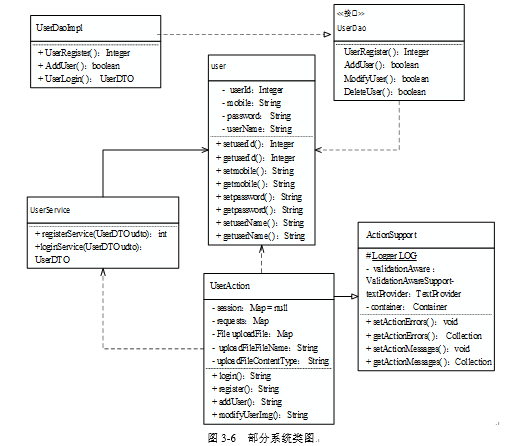


图3-6 管理员子系统操作流程图

## 3.4系统类模型

（绘制类图）（例）



## 3.5数据库设计

### 3.5.1数据库实体关系图（E-R图）设计

电子病历管理系统网站主要的实体有医生用户、病历，普通用户，管理员等，每个实体有相应的属性，实体之间的关系如下。

1. 电子病历管理系统总体E-R图

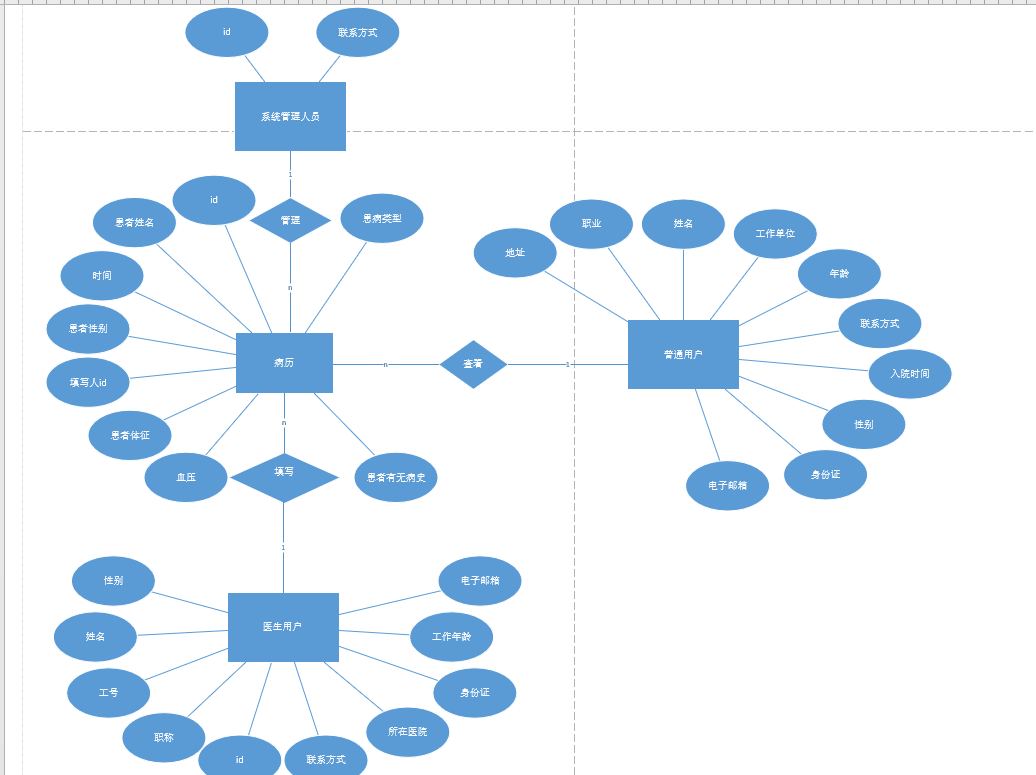


图3-8 电子病历管理系统总体E-R图

以下分别展示每个实体与属性的E-R图

1. 医生用户E-R图

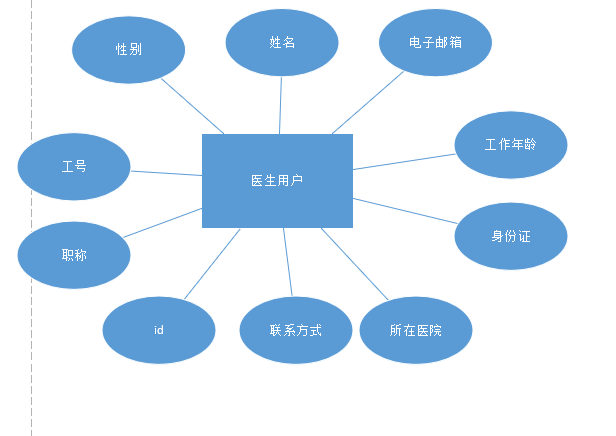


图3-9 医生用户E-R图

1. 病历E-R图

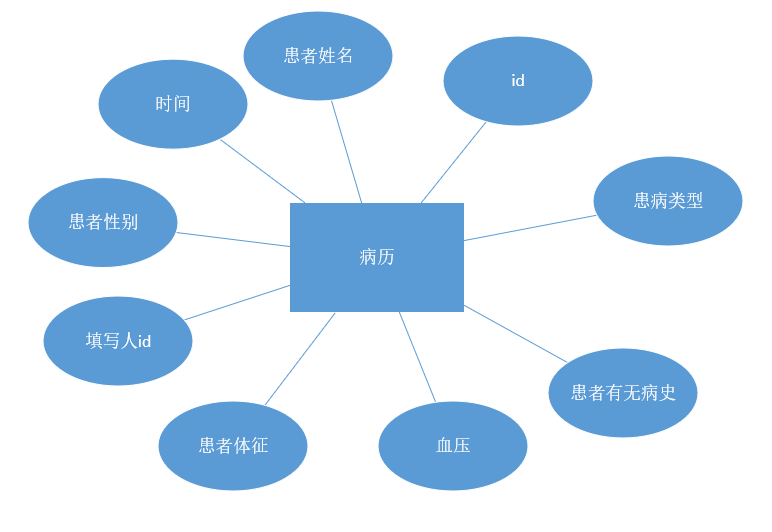


图3-10 病历E-R图

1. 普通用户

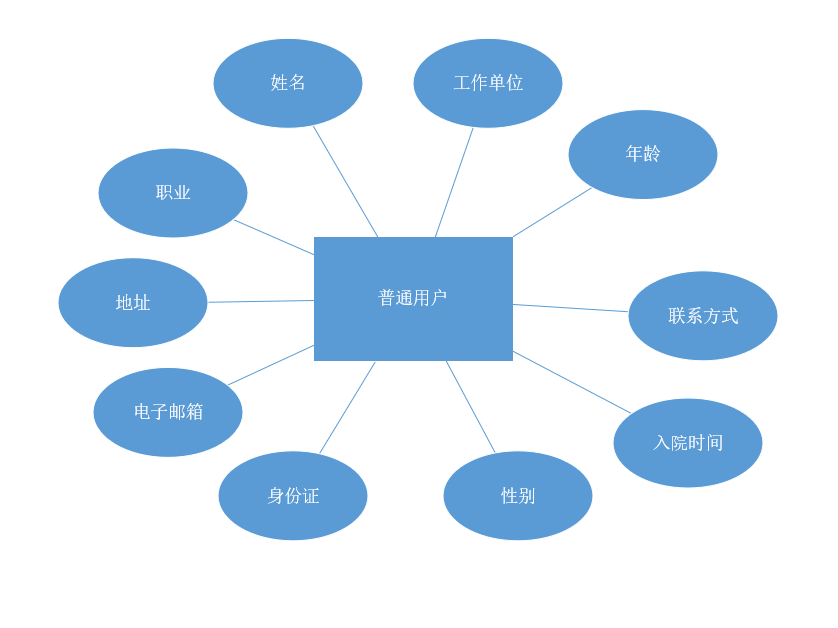


图3-11 普通用户E-R图

1. 管理员

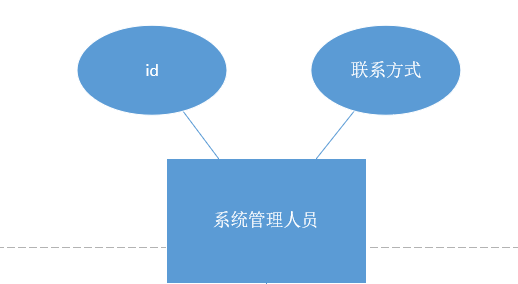


图3-12 管理员E-R图

### 3.5.2数据库总体关系设计图

（主外键，属性，表名，表间关系（1对1，1对多等））（例）

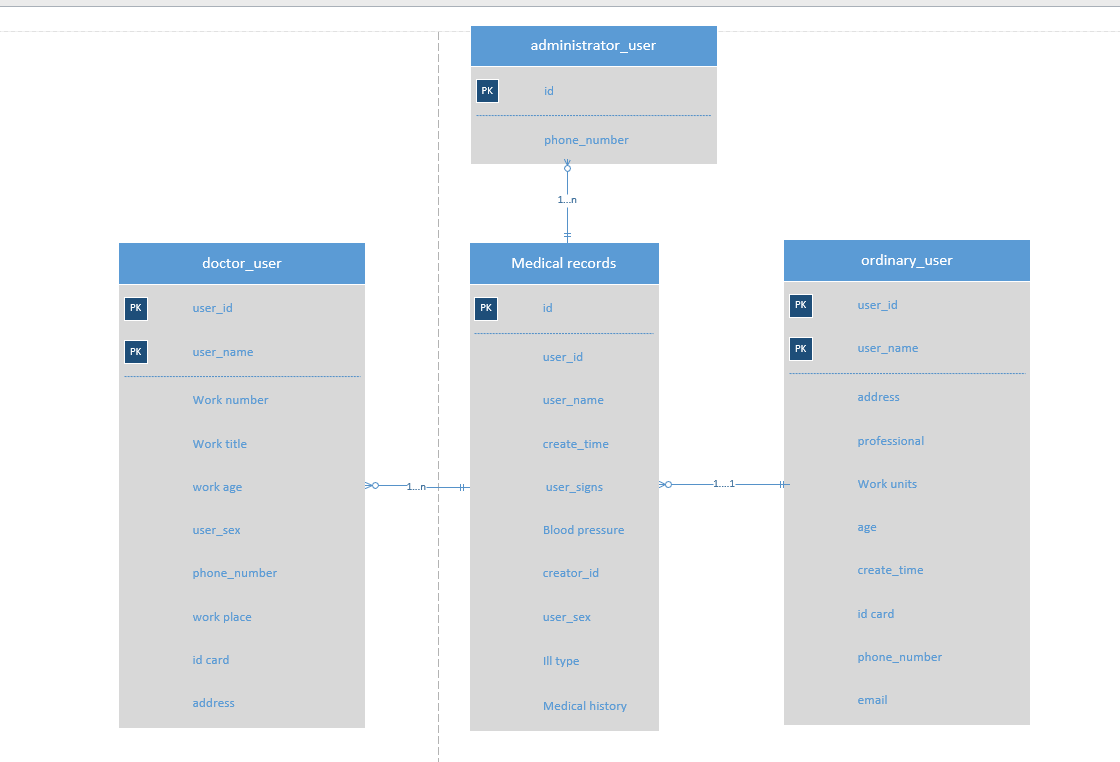


图3-13数据库总体关系图

### 3.5.3数据库表设计

在设计程序之前需要设计好系统所涉及的数据的表结构并将系统用到的模拟数据存储在数据库中，用于的系统开发和测试。基于以上分析得出的E-R图，本网站的数据库表有4个，以下为详细表结构。

1. 使用前台系统的普通用户在数据库存储时对应的是ordinary\_user表，表结构设计如下。

表3-1 ordinary\_user表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| user\_id | Int | 9 | not |  | yes |  | 用户编号 |
| Phone\_mobile | varchar | 11 |  |  |  |  | 用户电话 |
| email | varchar | 50 | not |  |  |  | 用户邮箱 |
| password | varchar | 50 | not |  |  |  | 用户密码 |
| user\_name | varchar | 50 |  |  |  |  | 用户名 |
| professional | varchar | 50 |  | 0 |  |  | 用户职业 |
| User\_sex | tinyint | 1 |  | 0 |  |  | 用户性别 |
| Work units | Int |  |  |  |  |  | 用户工作地点 |
| create\_time | datetime |  |  |  |  |  | 用户创建时间 |
| Id card | int | 18 |  |  |  |  | 身份证 |
| city\_id | varchar | 5 |  |  |  | yes | 城市 |
| province\_id | varcha | 5 |  |  |  | yes | 省份 |
| address | varchar | 20 |  |  |  |  | 详细地址 |
| Creator\_id | int | 9 |  |  |  |  | 填写人id |

2) 使用前台系统的医生用户在数据库存储时对应的是doctor\_users表，表结构设计如下

表3-2 doctor\_user表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| user\_id | Int | 9 | not |  | yes |  | 用户编号 |
| Phone\_mobile | varchar | 11 |  |  |  |  | 用户电话 |
| email | varchar | 50 | not |  |  |  | 用户邮箱 |
| password | varchar | 50 | not |  |  |  | 用户密码 |
| user\_name | varchar | 50 |  |  |  |  | 用户名 |
| professional | varchar | 50 |  | 0 |  |  | 用户职业 |
| User\_sex | tinyint | 1 |  | 0 |  |  | 用户性别 |
| Work units | Int |  |  |  |  |  | 用户工作地点 |
| create\_time | datetime |  |  |  |  |  | 用户创建时间 |
| Id card | int | 18 |  |  |  |  | 身份证 |
| city\_id | varchar | 5 |  |  |  | yes | 城市 |
| province\_id | varcha | 5 |  |  |  | yes | 省份 |
| address | varchar | 20 |  |  |  |  | 详细地址 |
| Word\_id | int | 12 |  |  |  |  | 工号 |
| Work\_age | int | 4 |  |  |  |  | 工龄 |

3） 还有一个重要的用户是管理员用户，管理员用户是后台管理系统的用户，管理员信息存储在admin表中，它包括的各个字段、类型等如下：

表3-3 admin表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| admin\_id | int | 5 | not |  | yes |  | 管理员编号 |
| admin\_email | varchar | 15 | not |  |  |  | 管理员邮箱 |
| admin\_name | varchar | 15 | not |  |  |  | 管理员名 |
| admin\_password | varchar | 10 | not |  |  |  | 管理员密码 |
| admin\_type | tinyint | 1 | not | 0 |  |  | 管理员类型 |
| last\_login\_time | datetime |  | not |  |  |  | 上次登录时间 |

4）本系统的病历信息表储存在medical\_record中，它包括的各个字段、类型等如下：

表3-4 medical\_record

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| user\_id | Int | 9 | not |  |  | Yes | 用户编号 |
| ID | int | 8 |  |  | Yes |  | 病历编号 |
| user\_name | varchar | 50 |  |  |  |  | 用户名 |
| professional | varchar | 50 |  | 0 |  |  | 用户职业 |
| User\_sex | tinyint | 1 |  | 0 |  |  | 用户性别 |
| create\_time | datetime |  |  |  |  |  | 创建时间 |
| Blood pressure | int | 5 |  |  |  |  | 血压 |
| MODIFY\_DATE | Datetime |  |  |  |  |  | 修改时间 |
| MODIFIER\_id | Int | 10 |  |  |  |  | 修改者id |
| Ill type | Varchar | 20 |  |  |  |  | 患病类型 |
| Medical history | Varchar | 50 |  |  |  |  | 病史 |
| user\_signs | varchar | 50 |  |  |  |  | 用户体征 |
| vital capacity | Int | 5 |  |  |  |  | 肺活量 |

## 3.6动态建模

动态模型，是指描述系统各量之间随时间变化而变化的规律的数学表达式。一般用微分方程或差分方程来表示。描述与操作时间和顺序有关的系统特征、影响更改的事件、事件的序列、事件的环境以及事件的组织的模型。一般用含有连续或离散时间变量的数学方程来表达。借助时序图、状态图和活动图，可以描述系统的动态模型。动态模型的每个图均有助于理解系统的行为特征。

时序图（Sequence Diagram），又名序列图、循序图，是一种UML交互图。它通过描述对象之间发送消息的时间顺序显示多个对象之间的动态协作。它可以表示[用例](https://baike.baidu.com/item/%E7%94%A8%E4%BE%8B/163511)的行为顺序，当执行一个用例行为时，其中的每条消息对应一个类操作或[状态机](https://baike.baidu.com/item/%E7%8A%B6%E6%80%81%E6%9C%BA/6548513)中引起转换的触发事件。

病历管理系统的各个功能模块的动态建模如下：

### 3.6.1账号管理设计时序图

账号管理模块中包括账号注册、账号登录、账号信息的修改。登录账号和修改账号信息均在用户注册账号后，以下是注册账号的时序图：

1）注册账号

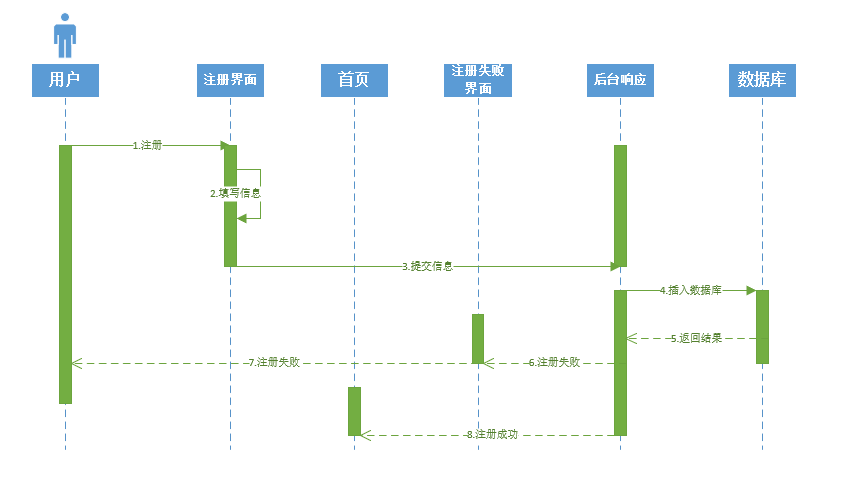


图3-14 账号注册时序图

2）账号登录

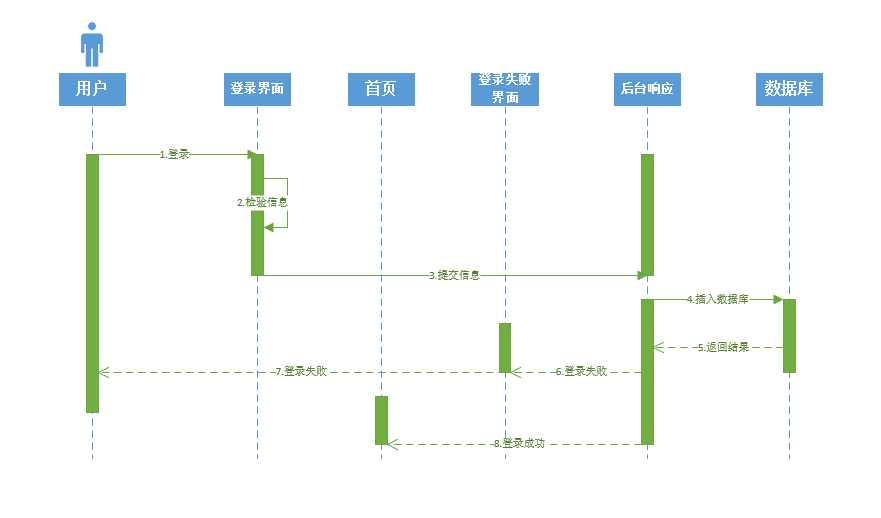
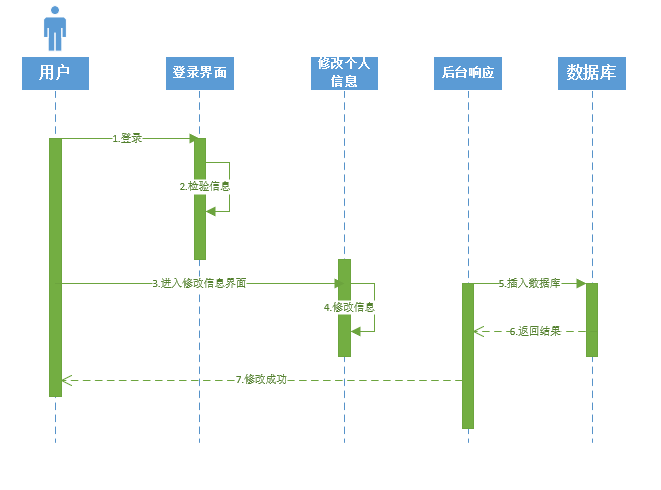


图3-15 账号登录时序图

3）用户个人信息修改

图3-16 用户个人信息修改时序图

4)用户病历信息修改

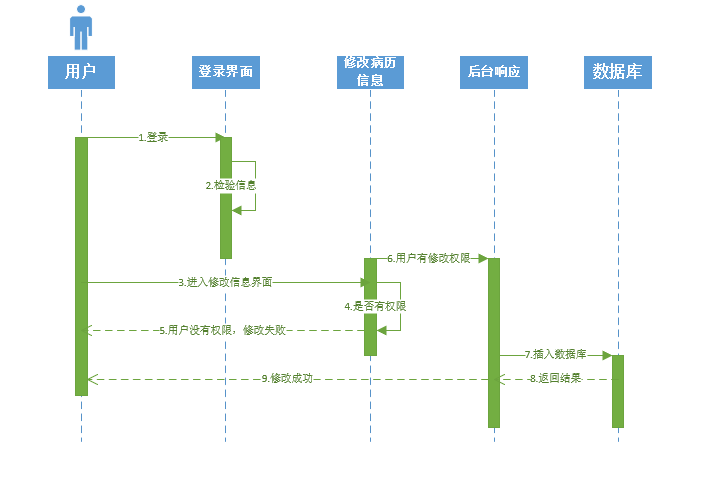


图3-17 用户病历信息修改时序图

## 3.7本章小结

本章对电子病历系统进行了分析与设计：分析系统功能，阐述系统设计的思想以及系统总体结构的设计和数据库的设计。

# 第四章 系统实现

## 4.1本系统开发过程中技术特色声明

本系统使用了java语言进行开发，拥有安全性高，可以信赖的字节码的检验机制，并且具有简单性，语法简捷，自动化内存管理，跨平台可移植，同时具有稳定性，完善的字节码安全机制，可靠的异常处理，而且它是面向对象的语言，始终关注应用中数据和操纵数据的算法，是一门动态语言，可动态编译，可动态加载，可动态执行，随需而变。

本系统使用数据库系统，大大提高了应用开发的效率，减轻数据库系统管理人员维护负担，当用户逻辑改变，数据的逻辑结构也需要改变时，由于数据库系统提供了数据语序之间的独立性，从而简化了应用程序的编制，又大大减少了应用程序的维护和修改。包括数据的完整性，安全性，多用户并发控制，故障恢复等，都由DBMS执行。既便于数据的集中管理，控制冗余，提高数据的利用率和一致性，又利于应用程序的开发和维护。

同时我们应用了区块链技术来保证数据库的灵活和安全性，它具有分布式去中心化存储，信息公开、内容不可篡改，信任共享功能三大特点，它是一个包含任何人都无法篡改、公开透明信息的系统，可以轻松的实现对信息的追踪。

## 4.2主界面的设计与实现

1）主界面的效果如图4-1所示：



图4-1 主界面

## 4.3前台子系统的设计与实现

### 4.3.1用户登录模块的设计与实现

1）用户登录模块的功能

用户登录模块：用户需要使用注册好的邮箱和密码进行登录，登录是用户开始操作的前提，用户只有在登录状态下才可以使用系统的主要功能，如修改信息和打印病历信息等。

2）用户登录模块的实际效果

用户登录界面如图4-2所示：

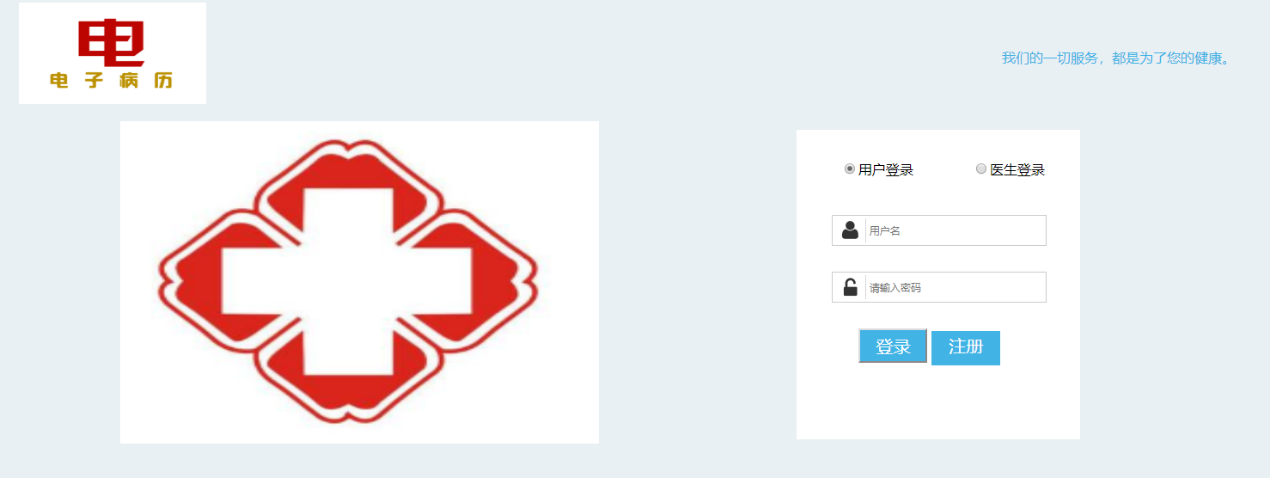


图4-2 用户登录界面

### 4.3.2用户注册模块的设计与实现

1）用户注册模块的功能

注册模块：实现用户注册功能，用户必须要使用邮箱和密码登陆系统后才能使用系统进行其他操作，用户名和密码需要使用系统的注册模块进行注册。

2）用户注册模块的实际效果

用户注册界面如图4-3所示：



图4-3 用户注册界面

3）用户个人信息修改界面

用户个人信息修改界面如图4-4所示



图4-4 用户个人信息修改界面

4）用户病历信息修改界面



图4-5 用户病历信息修改界面

## 4.4管理员子系统的设计与实现

### 4.4.1 管理员登录模块的设计与实现

1）管理员登录模块实现的功能

管理员登录模块主要实现管理员用户的登录功能，管理员的权限大于普通和医生用户。

2）管理员登录模块的实现效果图



图4-6 管理员登录界面

### 4.4.2管理系统主界面的设计与实现

管理系统主界面的实现：



图4-7 管理系统主界面

管理员可以编辑，修改，删除，查找病历，并可以添加或者删除用户。

## 4.5特色功能的设计与实现

### 4.5.1高级搜索功能



图4-8 管理系统高级搜索功能界面

此功能可以配合管理员用户来搜索病历或者找到重复无用的病历内容。支持多种搜索以及模糊搜索。

### 4.5.2随访管理功能



图4-9 管理系统随访管理功能界面

此功能可以对病历或者特殊患者进行跟进和备注，并直观的显示病人的信息和病历编写时间，并支持查询功能和打印功能。

## 4.6本章小结

以上内容就是电子病历系统的各个功能的实现，系统分为两个子系统，用户子系统和管理员子系统，分别包含相对应的模块，并介绍了详细的功能设计。本章介绍了系统的页面的设计方式和运行结果，并介绍了用户和管理员在各自的子系统进行相应操作的基本流程和页面截图。

# 第五章 系统测试

软件测试是伴随着软件的产生而产生的。早期的软件开发过程中软件规模都很小、复杂程度低，软件开发的过程混乱无序、相当随意，测试的含义比较狭窄，开发人员将测试等同于“调试”，目的是纠正软件中已经知道的故障，常常由开发人员自己完成这部分的工作。对测试的投入极少，测试介入也晚，常常是等到形成代码，产品已经基本完成时才进行测试。到了上世纪80年代初期，软件和IT行业进入了大发展，软件趋向大型化、高复杂度，软件的质量越来越重要。这个时候，一些软件测试的基础理论和实用技术开始形成，并且人们开始为软件开发设计了各种流程和管理方法，软件开发的方式也逐渐由混乱无序的开发过程过渡到结构化的开发过程，以结构化分析与设计、结构化评审、结构化程序设计以及结构化测试为特征。人们还将“质量”的概念融入其中，软件测试定义发生了改变，测试不单纯是一个发现错误的过程，而且将测试作为软件质量保证(SQA)的主要职能，包含软件质量评价的内容，Bill Hetzel在《软件测试完全指南》(Complete Guide of Software Testing)一书中指出：“测试是以评价一个程序或者系统属性为目标的任何一种活动。测试是对软件质量的度量。”这个定义至今仍被引用。软件开发人员和测试人员开始坐在一起探讨软件工程和测试问题。

软件测试已有了行业标准(IEEE/ANSI )，1983年IEEE提出的软件工程术语中给软件测试下的定义是：“使用人工或自动的手段来运行或测定某个软件系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别”。这个定义明确指出:软件测试的目的是为了检验软件系统是否满足需求。它再也不是一个一次性的，而且只是开发后期的活动，而是与整个开发流程融合成一体。软件测试已成为一个专业，需要运用专门的方法和手段，需要专门人才和专家来承担。

## 5.1测试的目的

软件测试整体是验证功能的实现、可用性，检查程序的错误，最终目的是为了提高用户体验；

1)软件测试是为了发现错误而执行程序的过程。

2)测试是为了证明程序有错，而不是证明程序无错。(发现错误不是唯一目的)

3)一个好的测试用例在于它发现至今未发现的错误。

4)一个成功的测试是发现了至今未发现的错误的测试。

## 5.2测试的方法

### 5.2.1黑盒测试

黑盒测试，顾名思义即为将软件测试环境模拟为不可见的“黑盒”。通过数据输入观察数据输出，检查软件内部功能是否正常。测试展开时，数据输入软件中，等待数据输出。数据输出时若与预计数据一致，则证明该软件通过测试，若数据与预计数据有出入，即便出入较小亦证明软件程序内部出现问题，需尽快解决。

首先对各个模块进行黑盒测试，在各种条件下运行模块是否能正常运行。

### 5.2.2单元测试

单元测试即为将整个软件分解为各个单元，随后对单元进行测试。此类测试策略的优点在于所需分析数据较少，且针对性较强，程序开发者于开发过程中可通过操作经验明确出现问题的大致区域，随后针对此类问题对相关单元展开分析，进行问题排查。但需注意的是，某些程序中无具体单元驱动程序，即单个单元无法有效驱动，易出现问题，若针对此类软件展开测试，需重点注意此类分解单元。

JUnit是一个开源的java语言的单元测试框架，专门针对java语言设计, 使用最广泛, JUnit是标准的单元测试架构，此次选用Junit来进行单元测试。

### 5.3系统测试

#### 5.3.1黑盒测试

1）前台登录模块测试

前台登录测试数据如表5-1所示，测试结果如图5-1、5-2所示。

表5-1前台测试数据

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能模块 | 前台登录 | | | | |
| 测试目的 | 验证是否输入合法的信息，允许合法登录，阻止非法登录 | | | | |
| 测试数据 | 用户名dzbl@qq.com 密码66666 | | | | |
| 操作步骤 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 输入用户名，密码，点击“登录”。 | 用户名=dzbl  密码=66666 | “账号密码错误” | 系统提示”账号密码错误” | 正常 |
| 2 | 输入用户名，密码，点击“登录”。 | 用户名=dzbl@qq.com  密码=666 | “账号密码错误” | 系统提示“账号密码错误” | 正常 |
| 3 | 输入用户名，密码，点击“登录”。 | 用户名=dzbl@qq.com  密码=66666 | 进入系统 | 进入系统 | 正常 |

## 5.4测试结论

（通过5.3测试之后对软件的简单评价）

## 5.5本章小结

（阐述本章内容）

（之后还可以加一个大总结作为论文心得体会）

结论及展望

（简述软件功能，优点，缺点，以及前景）

致谢

（感谢老师，感谢学校，感谢同学，感谢父母）

参考文献

[1] 安徽师范大学. 基于区块链的电子病历安全搜索方法[P]. 2019-01-01.

[2] 石家庄铁道大学. 基于区块链的电子病历访问控制方法:CN201910590599.1[P]. 2019-10-11.

[3] 森美嘉（深圳）医疗健康科技发展有限公司. 一种基于区块链的电子病历管理方法[P]. 2018-12-14.

[4] 暨南大学. 基于区块链的医疗电子病历分布式管理系统及其建设方法[P]. 2019-01-15.

[5] 深圳市网心科技有限公司. 一种基于区块链的电子病历管理方法、装置、系统及设备:CN201910718783.X[P]. 2019-11-05.

[6] 杭州银江智慧医疗集团有限公司. 电子病历安全系统[P]. 2015-03-04.

[7] 赛拓信息技术有限公司. 电子病历管理系统[P]. 2019-02-15.

[8] 广州慧扬健康科技有限公司. 电子病历档案资料管理系统[P]. 2017-10-20.

[9] 南通市第一人民医院. 一种电子病历管理系统:CN201620056895.5[P]. 2016-08-31.

[10] 天津柳叶峰医药技术开发有限公司. 一种电子病历管理系统:CN201521125090.3[P]. 2016-06-22.

[11]Wang Yong Liu Like Lu Hao.Thinking on the development of computer-based patient record system. West China Hospital of Si Chuan University,Chengdu610041.

[12]Bruce Eckle.Thinking in java[M].Upper Saddle River , New Jersey,USA:Prentice

Hall,2009:61-74.

[13]Zoya Ali. Designing Object Oriented Software Applications within the Context of Software Frameworks[D]. Ohio State University,2011.