珠海科技学院

**数据库课程设计报告**

|  |  |
| --- | --- |
| 学院： | 计算机学院 |
| 年级专业名称： | 2019级 软件工程 专业 |
| 课程设计科目： | 数据库课程设计 |
| 题目： | 校园医务室服务系统 |
| 学号、姓名： | 04191315何翔；04191315董康平；04191316赖韦帆；04191317黄文斌；04191321杨卓霖 |
| 指导教师： | 郭晓燕 |
| 完成时间： | 2021年 月 |

目 录

[1.系统开发平台](#_Toc14155)

[2.数据库规划](#_Toc18851)

[2.1 任务陈述](#_Toc4848)

[2.2 任务目标](#_Toc8893)

[3.系统定义](#_Toc10789)

[3.1 系统边界](#_Toc16817)

[3.2 用户视图](#_Toc15155)

[4.需求分析](#_Toc29164)

[4.1 用户需求说明](#_Toc30960)

[4.1.1 数据需求](#_Toc25482)

[4.1.2 事务需求](#_Toc24895)

[4.2 系统需求说明](#_Toc14918)

[4.2.1软件环境](#_Toc22700)

[4.2.2硬件环境](#_Toc24890)

[4.2.3初始数据库大小](#_Toc15293)

[4.2.4数据库增长速度](#_Toc12638)

[4.2.5记录查找的类型和平均数量](#_Toc23533)

[4.2.6性能](#_Toc10523)

[4.2.7安全性](#_Toc19825)

[5.数据库逻辑设计](#_Toc9593)

[5.1 E-R图](#_Toc21561)

[5.2 数据字典](#_Toc20903)

[5.2.1实体和联系属性](#_Toc29120)

[5.2.2属性描述](#_Toc26618)

[5.3 关系表](#_Toc25265)

[6.数据库物理设计](#_Toc30603)

[6.1 索引](#_Toc4719)

[6.2 视图](#_Toc15389)

[6.3 安全机制](#_Toc31731)

[6.3.1系统安全](#_Toc16892)

[6.3.2数据安全](#_Toc27309)

[7.应用程序设计](#_Toc7656)

[7.1 功能模块](#_Toc25408)

[7.1.1计算功能](#_Toc334)

[7.1.2存储功能](#_Toc31572)

[7.1.3输入及输出功能](#_Toc30047)

[7.1.4交互功能](#_Toc24794)

[7.2 界面设计](#_Toc7107)

[7.2.1登录界面](#_Toc5228)

[7.2.2管理员用户界面](#_Toc2552)

[7.2.3医生用户界面](#_Toc14256)

[7.2.4学生用户界面](#_Toc15521)

[7.3 事务设计](#_Toc6129)

[7.3.1 Datetime类型数据查询](#_Toc19022)

[7.3.2医保信息处理](#_Toc7741)

[7.3.3数据批量录入](#_Toc31111)

[8.测试和运行](#_Toc22997)

[9.总结](#_Toc15385)

[参考文献](#_Toc16046)

# 1.系统开发平台

**题目：**企业库存管理系统

**数据库：**MySQL 8.0+

**应用开发工具：** IDEA

**操作系统：** Microsoft Windows 10 Home Edition

# 2.数据库规划

## 2.1 任务陈述

校园医务室服务系统的目的是为了通过收集和存储指定对象的信息数据，及控制相关人员对以上数据的操作，辅助校医务室工作人员完成对基本医疗信息的管理工作，并在必要的时候向更高管理层提供所需要的财务等信息。另外该系统还具有对医疗保险和过敏药物两种特定信息的管理功能。

## 2.2 任务目标

控制（添加，更新，删除）有关学生信息

控制（添加，更新，删除）有关医生信息

控制（添加，更新，删除）有关药物信息

控制（添加，更新，删除）有关库存信息

添加，查询有关医疗记录信息

添加，查询有关过敏药物信息

添加，查询有关处方信息

查询，统计有关学生信息

查询，统计有关医生信息

查询，统计有关药物信息

查询，统计有关药房信息

对药房库存信息提供报表

对有关医疗记录信息提供报表

对有关过敏药物信息提供报表

对校医务室财务信息提供报表

自动计算相关费用，核对药物信息

# 3.系统定义

## 3.1 系统边界

**学生**

校医务室服务系统

医疗记录

查询

查询

记录

查询

提交报表

查询&更新

控制

提交报表

**学生信息系统**

**学校财务系统**

查询

**药品**

控制

药房库存

属于

**管理者用户**

控制

**人事管理系统**

查询

**医生**

图1. 系统边界

## 3.2 用户视图

表1 各类用户功能需求

|  |  |
| --- | --- |
| 用户视图 | 功能需求 |
| 学生用户 | 列出该用户的学生基本信息  列出指定敏感药物的详细信息  列出指定医疗记录的详细信息  输出该用户医疗记录信息  输出该用户过敏药物信息 |
| 医生用户 | 列出该用户的基本信息  列出指定学生的基本信息  列出指定学生的过敏药物信息  列出指定学生的医疗记录信息  列出选定药物的详细信息  自动计算相关费用  自动核对药物信息  更新库存信息  录入新处方信息  录入新库存信息  录入新医疗记录信息  录入新过敏药物信息  输出指定药房库存信息  输出指定学生的医疗记录信息  输出指定学生的过敏药物信息 |
| 管理员用户 | 列出指定药物信息  列出指定药房信息  列出指定学生信息  列出指定医生信息  列出指定医疗记录信息  录入新药物信息  录入新药房信息  录入新学生信息  录入新医生信息  删除选定学生信息  删除选定医生信息  更新库存信息  更新学生信息  输出指定学生信息  输出指定医生信息  输出指定药房库存信息 |

# 4.需求分析

## 4.1 用户需求说明

### 4.1.1 数据需求

**(1)学生用户视图:**

基本信息： 学号，姓名，性别，年龄，班级，学院，专业，联系电话。

过敏药物信息： 编号，名称，型号，发现日期，诊断医师

医疗记录信息： 时间，处方编号，自费费用，医保费用，诊断医师

学号唯一，可决定多条过敏药物信息和医疗记录信息。处方编号唯一，每条过敏记录由编号和学号唯一确定。学生只可以对以上信息进行查询，不可以进行修改。

**（2）医生用户视图:**

医生信息： 编号，姓名，性别，年龄，所在药房编号

学生信息： 学号，姓名，性别，年龄，班级，学院，专业，联系电话

药物信息： 编号，名称，型号，价格，库存量，医保比例，注意事项

处方信息： 编号，药房编号，时间，治疗方案，自费费用，医保费用

过敏药物信息： 编号，学号，名称，型号，诊断医师

医疗记录信息： 时间，处方编号，总费用，诊断医师

学号及各编号唯一；每条治疗方案中包含多条处方信息。值班医生可对处方信息，医疗记录信息进行更新和查询，但只可查询基本信息，学生信息和药物信息。

**(3)管理用户视图:**

药物信息： 编号，名称，型号，价格，库存量，医保比例，注意事项

学生信息： 学号，姓名，性别，年龄，班级，学院，专业，联系电话

处方信息： 编号，药房编号，时间，治疗方案，自费费用，医保费用

医生信息： 编号，姓名，性别，年龄，所在药房编号

药房信息： 药房编号，药物编号，现库存量

过敏药物信息： 编号，学号，名称，型号，诊断医师

医疗记录信息： 时间，处方编号，总费用，诊断医师

各编号唯一；每条医疗记录可包含多个处方编号；药方信息中，相同药物编号可对应多条药房信息。每条值班信息由医生编号，药房编号和时间唯一确定。校医院管理者可对药物信息和药房信息进行更新和查询，但只可查询医疗记录信息。

### 4.1.2 事务需求

**（1）学生用户视图:**

数据查询：列出该用户的学生基本信息

列出选定敏感药物的详细信息

列出选定医疗记录的详细信息

数据输出：输出该用户医疗记录信息

输出该用户过敏药物信息

**（2）医生用户视图:**

数据查询：列出该用户的基本信息

列出指定学生的基本信息

列出指定学生的过敏药物信息

列出指定学生的医疗记录信息

列出选定药物的详细信息

数据录入：录入新处方信息

录入新库存信息

录入新医疗记录信息

录入新过敏药物信息

更新库存信息

数据输出：输出该用户医疗记录信息

输出该用户过敏药物信息

输出指定药房库存信息

输出指定学生的医疗记录信息

输出指定学生的过敏药物信息

其它事务：自动计算相关费用

自动核对药物信息

**(3)管理用户视图:**

数据查询：列出指定药物信息

列出指定药房信息

列出指定学生信息

列出指定医生信息

列出指定医疗记录信息

数据录入：录入新药物信息

录入新药房信息

录入新学生信息

录入新医生信息

数据输出：输出指定学生信息

输出指定医生信息

输出指定药房库存信息

数据更新/删除：删除选定学生信息

删除选定医生信息

更新药房库存信息

更新学生信息

## 4.2 系统需求说明

### 4.2.1软件环境

软件环境需要支持Window 10操作系统，及SQL Server 2008数据库

### 4.2.2硬件环境

Intel(R) Core(TM)i5@ 1.80GHz , 8GB内存,256G SSD。

### 4.2.3初始数据库大小

大约有学生20名，医生5名，管理用户3名；

药物目录共包含40种药物，药房4处；其它记录信息为0。

### 4.2.4数据库增长速度

学生为每一年添加1000名左右，同时删除1000名左右；医生及其他信息增长速度不定。

### 4.2.5记录查找的类型和平均数量

查找医疗记录的情况………大约每天20次

查找学生信息的情况………大约每天10次

查找药物记录的情况………大约每天30次

查找过敏药物记录的情况………大约每天10次

查找药房信息记录的情况………大约每月1次

### 4.2.6性能

单个记录查询时间少于1秒，高峰期少于5秒

多个记录查询时间少于5秒，高峰期少于10秒

更新/保存记录时间少于1秒，高峰期少于5秒

### 4.2.7安全性

每个用户在进入系统时都必须有口令保护；

每个用户分配特定的用户视图所应有的访问权限；

不同用户根据其事先定义的角色不同进入不同的用户视图界面。

# 5.数据库逻辑设计

## 5.1 E-R图

学生

医生

医疗记录

处方

就诊

就诊

过敏

添加

添加

药品

维护

药房

属于

属于

属于

图2 系统E-R图

## 5.2 数据字典

### 5.2.1实体和联系属性

学生student：学号SNo，姓名SName，性别SSex，年龄SAge，班级Grade，

学院College，专业Major，联系电话Phone,密码Password ；

医生doctor ：编号DNo，姓名DName，性别DSex，年龄DAge，所在药房Room,

密码Password ；

药品medicine：编号MNo，名称MName，型号MStyle，价格MPrice,注意事项MTip

医保比例selfPercent

药房Pharmacy ： 编号PHNo，位置PHName

医疗记录Record：编号RNo，处方编号PNo，时间RDate，自付费用SelfPrice，

医保费用FreePrice，学号SNo，诊断医师编号DNo

处方Prescription： 编号PNo，时间PDate，学号SNo，诊断医师编号DNo

库存量StoreAmount：药物编号MNo，药房编号PHNo，现库存量

过敏药物Allergy: 药物编号MNo，学号SNo，处方编号PNo

### 5.2.2属性描述

表2 关系属性说明

|  |  |  |  |
| --- | --- | --- | --- |
| 实体 | 属性 | 类型及大小 | 空 |
| student | 学号SN | CHAR(8) | 否 |
| 姓名SName | varchar(16) |  |
| 性别SSex | CHAR(2) |  |
| 年龄SAge | int |  |
| 班级Grade | varchar(20) |  |
| 学院College | varchar(20) |  |
| 专业Major | varchar(20) |  |
| 联系电话Phone | varchar(15) |  |
| 密码Password | varchar(8) |  |
| doctor | 编号DNo | CHAR(8) | 否 |
| 姓名DName | varchar(16) |  |
| 性别DSex | CHAR(2) |  |
| 年龄DAge | int |  |
| 所在药房Room | CHAR(4) |  |
| 密码Password | varchar(8) |  |
| medicine | 编号MNo | CHAR(8) | 否 |
| 名称MName | varchar(30) |  |
| 型号MStyle | MStyle CHAR(20) |  |
| 价格MPrice | int |  |
| 医保比例selfPercent | int |  |
| Prescription | 编号PNo | CHAR(8) | 否 |
| 时间PDate | datetime | 否 |
| 学号SNo | CHAR(8) |  |
| 诊断医师编号DNo | CHAR(8) |  |
| Record | 编号RNo | CHAR(8) | 否 |
| 处方PNo | CHAR(8) | 否 |
| 时间Date | datetime |  |
| 自费费用SelfPrice | int |  |
| 医保费用FreePrice | int |  |
| 学生SNo | CHAR(8) |  |
| 医生DNo | CHAR(8) |  |
| Allergy | 药物MNo | CHAR(8) | 否 |
| 学生SNo | CHAR(8) | 否 |
| 处方PNo | CHAR(8) |  |
| StoreAmount | 药物MNo | CHAR(8) | 否 |
| 药房PHNo | CHAR(8) | 否 |
| 现库存量Amount | int |  |

## 5.3 关系表

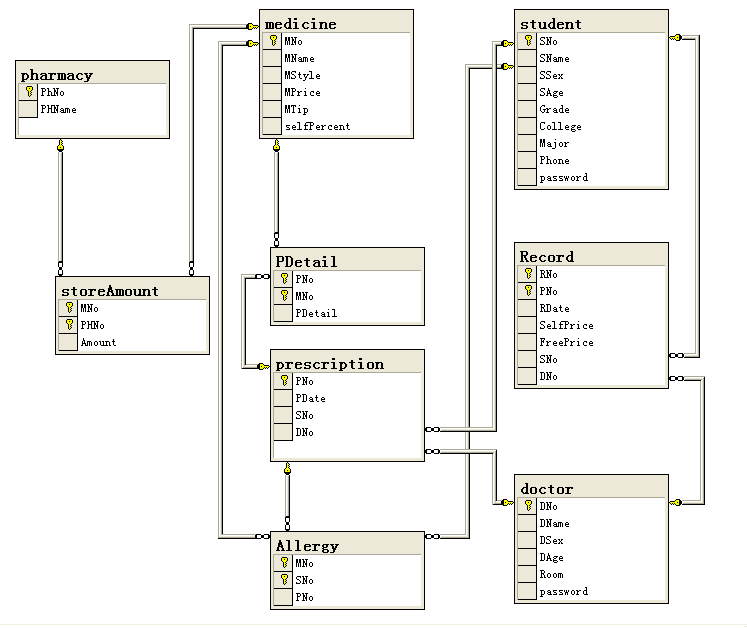


图3 主、外键约束

# 6.数据库物理设计

## 6.1 索引

Hospital数据库中索引如表3所示：

表3 索引列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ObjectName** | **IndexName** | **ID** | **AverageFreeBytes** | **LogicalFragementation** |
| Allergy | PK\_\_Allergy\_\_060DEAE8 | 1 | 7964.0 | 0.0 |
| storeAmount | PK\_storeAmount | 1 | 7690.0 | 0.0 |
| storeAmount | ph\_key | 4 | 7830.0 | 0.0 |
| PDetail | PK\_PDetail | 1 | 7935.0 | 0.0 |
| Record | PK\_\_Record\_\_023D5A04 | 1 | 7706.0 | 100.0 |
| Record | pno\_key | 8 | 7970.0 | 0.0 |
| medicine | PK\_\_medicine\_\_7A9C383C | 1 | 4712.0 | 100.0 |
| student | PK\_\_student\_\_76CBA758 | 1 | 7719.0 | 100.0 |
| doctor | PK\_\_doctor\_\_78B3EFCA | 1 | 7844.0 | 100.0 |
| pharmacy | PK\_pharmacy | 1 | 8021.0 | 100.0 |
| prescription | PK\_\_prescription\_\_7E6CC920 | 1 | 7823.0 | 0.0 |

在全部索引中除每张表的主键索引外还包括另外两个单独建立的索引：

（1）storeAmount 中的ph\_key索引：在查询某种药品的库存量多为针对某以药房查询，而且在添加库存是也是按照药房批量更新添加，因此建立了此索引；

（2）Record 中的pno\_key索引：虽然每一医疗记录都有自己的编号，但大多数情况下是对具体的处方信息进行查询，而且一条有自费和医保费用构成的财务记录也是由处方编号而定的，所以建立了此索引。

## 6.2 视图

Hospital数据库中主要建立了以下两个视图：

①CREATE VIEW dbo.AllergyDetial AS

SELECT dbo.Allergy.MNo, dbo.Allergy.SNo, dbo.Allergy.PNo, dbo.medicine.MName,

dbo.medicine.MStyle, dbo.medicine.MTip, dbo.student.SName,

dbo.prescription.PDate, dbo.doctor.DNo, dbo.doctor.DName

FROM dbo.Allergy INNER JOIN

dbo.medicine ON dbo.Allergy.MNo = dbo.medicine.MNo INNER JOIN

dbo.student ON dbo.Allergy.SNo = dbo.student.SNo INNER JOIN

dbo.prescription ON dbo.Allergy.PNo = dbo.prescription.PNo AND

dbo.student.SNo = dbo.prescription.SNo INNER JOIN

dbo.doctor ON dbo.prescription.DNo = dbo.doctor.DNo

②CREATE VIEW dbo.PreDetail AS

SELECT dbo.prescription.PNo, dbo.prescription.PDate, dbo.prescription.SNo,

dbo.prescription.DNo, dbo.Record.RNo, dbo.Record.SelfPrice, dbo.Record.FreePrice,

dbo.medicine.MName, dbo.medicine.MStyle, dbo.medicine.MPrice,

dbo.medicine.selfPercent, dbo.student.SName, dbo.doctor.DName

FROM dbo.prescription INNER JOIN

dbo.Record ON dbo.prescription.PNo = dbo.Record.PNo INNER JOIN

dbo.student ON dbo.prescription.SNo = dbo.student.SNo AND

dbo.Record.SNo = dbo.student.SNo INNER JOIN

dbo.doctor ON dbo.prescription.DNo = dbo.doctor.DNo AND

dbo.Record.DNo = dbo.doctor.DNo CROSS JOIN

dbo.medicine

建立以上视图主要是因为：

（1）基于以上视图实现复杂查询，以避免书写频繁执行连接操作；

（2）以上视图所包含的表的数据关系到许多用户，视图可以隐藏基本表；可以禁止所有用户访问数据库表，而要求医生等用户只能通过视图操作数据；

（3）在一定程度上保护高级语言应用程序不受某些数据库结构修改的影响。

## 6.3 安全机制

### 6.3.1系统安全

为学生和医生用户在数据库中存储登录口令，登录时必须经验证一致才可进入系统；

由于管理员用户有频繁的插入删除操作且该类用户数量有限，所以为每一个管理员用户在数据库中建立单独的登录名和用户名，并为他们指定已建立的Manager角色（具有表student，doctor，medicine，storeAmount的全部权限）。在管理员登录时，其密码由SQLSever实现验证，由于SQLSever的登录密码是经加密保存的，所以具有较高的安全性。

### 6.3.2数据安全

在该系统中，任何有关删除或添加数据的操作都需要二次确认才可以在数据库中真正执行。通过高级语言应用程序的限制，不同的用户只能查询修改其有权访问的数据。

# 7.应用程序设计

## 7.1 功能模块

### 7.1.1计算功能

(1) 统计具有指定属性的群体所含个体的总量；

(2) 医疗费用中医保金额和自费金额的自动计算和区分；

(3) 过敏药物的判定及其与治疗药物的冲突选择。

### 7.1.2存储功能

存储有关对象的具体信息及学生与处方，处方与医生，医生与药房，药房与药品的关系，并能方便快速地实现对以上信息的删除增加等操作。

### 7.1.3输入及输出功能

输入的限定信息等数据经处理后，可以在显示器上按事先定义的格式显示，特别是在就诊过程中可以跟踪当前状态(挂号，诊疗，结算，取药)。另外还能通过指定设备存储账单等报表，以便打印或长期保存。

### 7.1.4交互功能

本系统具有仿Windows窗口交互界面，会在工作人员进行输入等操作时给予必要的提示，或者在操作发生错误时给予必要的提示和帮助。另外在医生用户的最终结算操作中具有自动计算找零的仿收银台功能。

## 7.2 界面设计

### 7.2.1登录界面





图4 登录界面

### 7.2.2管理员用户界面

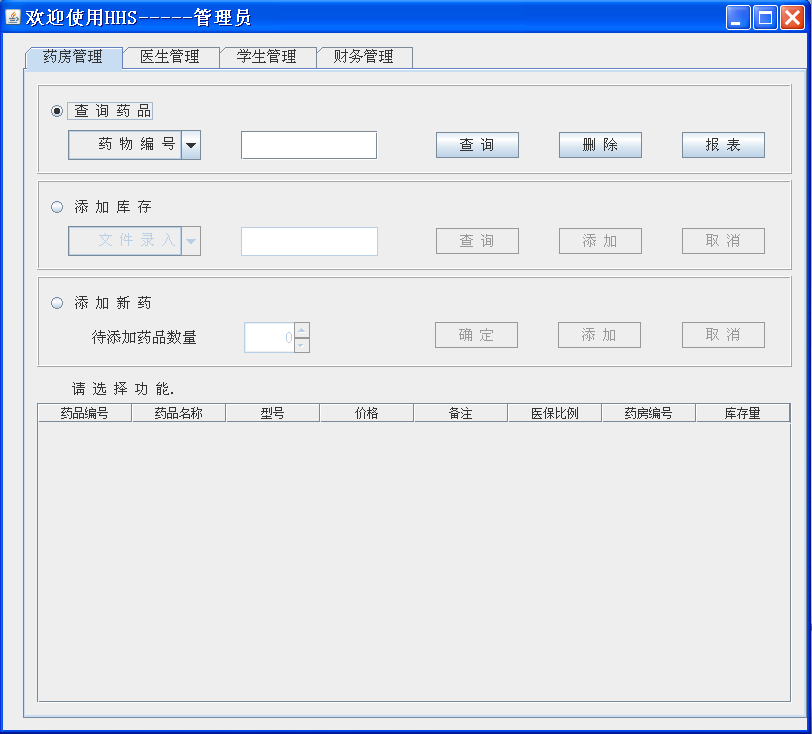


图5 管理员界面

(1)查询药物

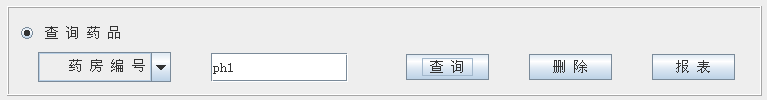




图6 查询药物

(2)报表

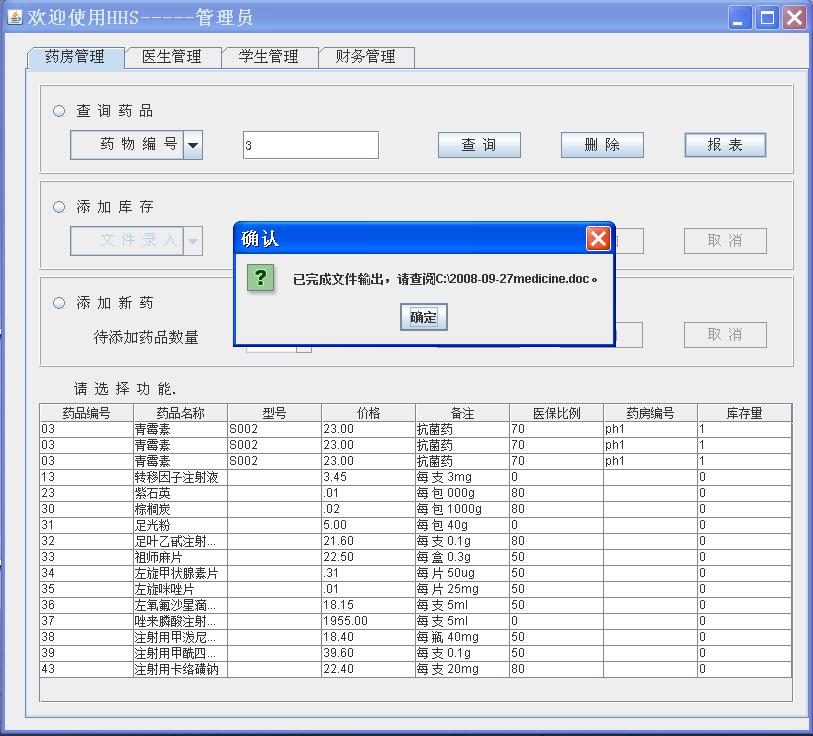
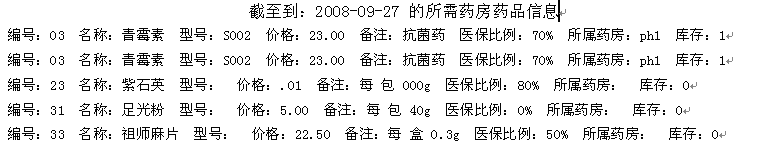
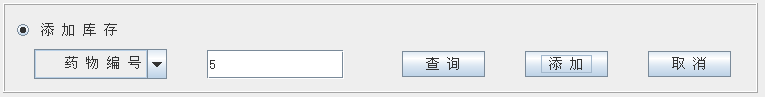


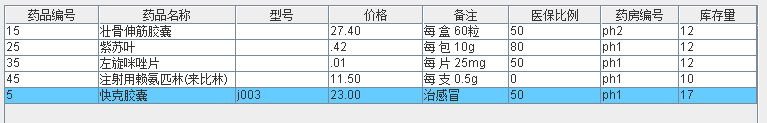
图7 统计报表

生成的报表如下：



(3)添加库存





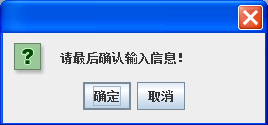


图8 添加药品库存

(4)添加新药

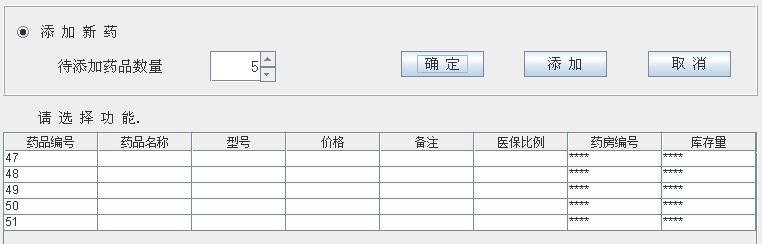


图9 添加新药

（5）医生管理

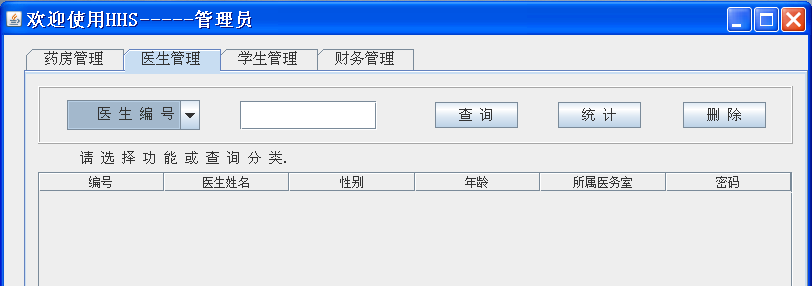


图10 管理医生

（6）学生管理

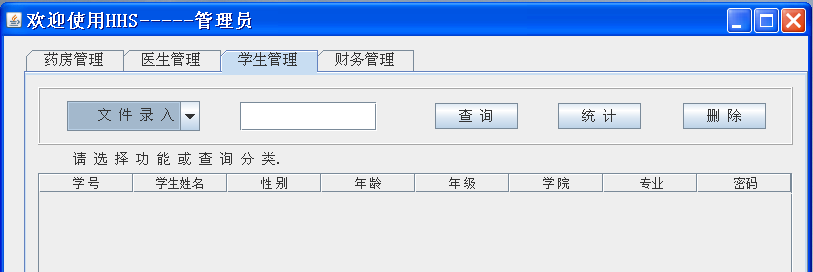


图11 管理学生

（7）财务管理

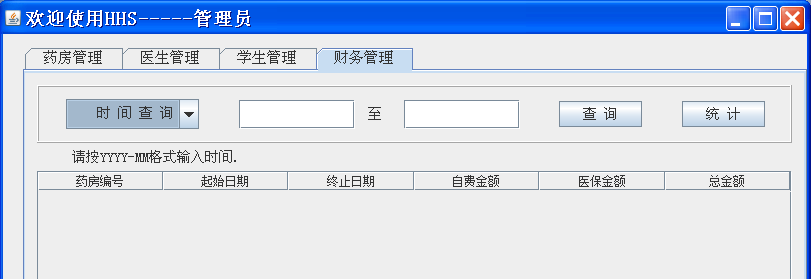


图12 财务管理

### 7.2.3医生用户界面

(1)挂号

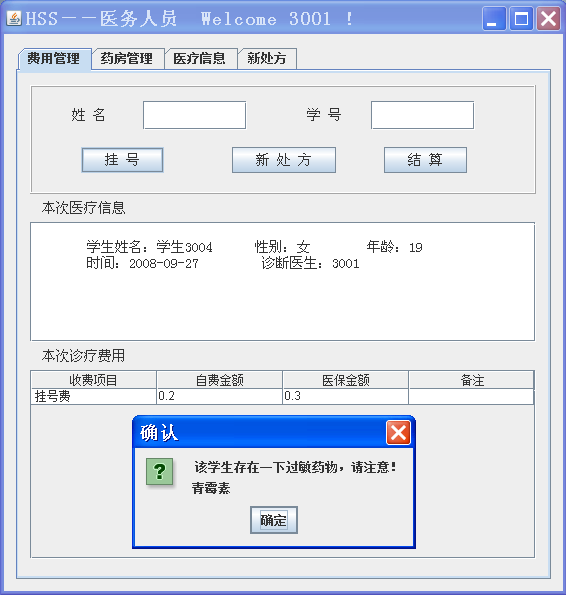


图13 挂号

(2)新处方

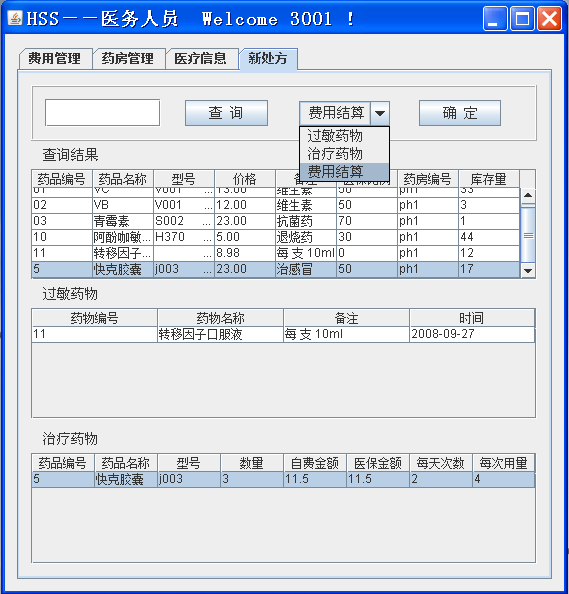
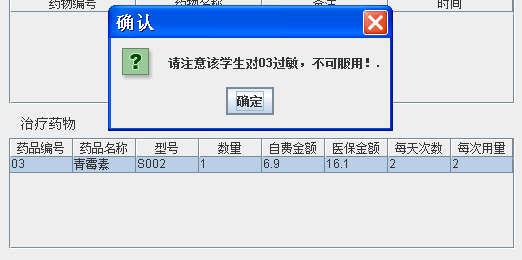
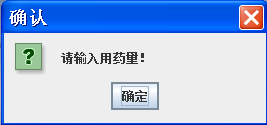
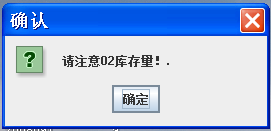


图14 开处方

部分提示信息：



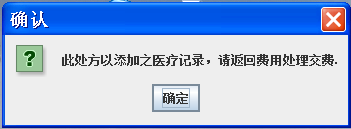


图15 开处方提示信息

(3)医疗信息

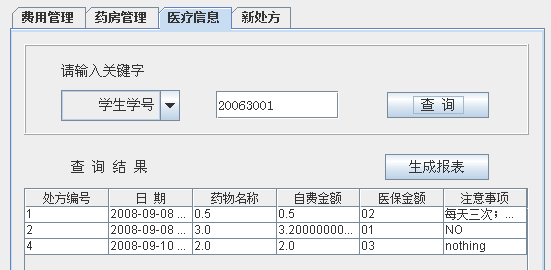
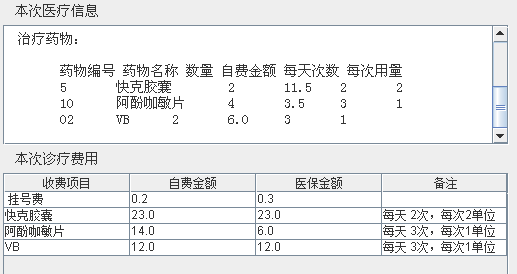


图16 医疗信息

(4)结算



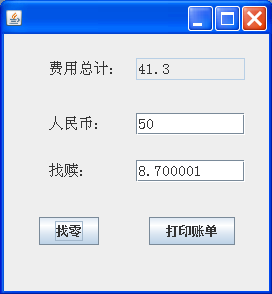
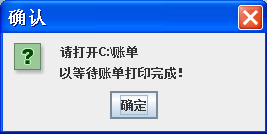
 

图17 费用结算

最终打印的账单：

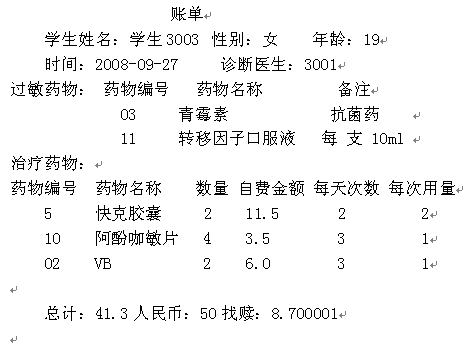


图19 打印账单

### 7.2.4学生用户界面

(1)基本信息



图20 学生基本信息

(2)敏感药物



图21 学生过敏药物

(3)医疗记录



图22 学生医疗记录

## 7.3 事务设计

### 7.3.1 Datetime类型数据查询

时间是对于医疗记录是很重要的属性，本系统的数据库在存储时间时使用的是其提供的Datetime类型，这样就可以通过调用函数DATEDIFF，方便的查询某一时间点或某一时间段的指定信息。以下是以学生查询自己某一时间段的医疗记录的实例：

**if**(!todate.trim().equals("YYY-MM-DD")&&!fromdate.trim().equals("YY-MM-DD"))

{RSTemp=RSTemp+" and DATEDIFF(dd,record.Rdate,'"+fromdate

+"')<=0 and DATEDIFF(dd,record.Rdate,'"+todate+"')>=0";}

...

studentDemo.open("sa", "");

String sql3="select Record.RNo,Record.PNo,RDate,PDetail.MNo,

Record.SelfPrice,Record.FreePrice,PDetail.PDetail"

+" from Record,prescription,PDetail"

+" where Record.PNo=Prescription.PNo " +RSTemp

+" and PDetail.PNo=Prescription.PNo and record.SNo="+**this**.SId;

ResultSet RecordSearchResult=studentDemo.executeQuery(sql3);

RecordSearchResult.next();

### 7.3.2医保信息处理

随着医疗保险改革的不断深入，对于在校学生医保信息的管理也越来越受到各方的重视。怎样有效的掌握和利用医疗保险信息，以及实现个人医疗信息档案的信息化、规范化和集成化，实现计算机的智能化管理，提高相关工作人员的工作效率成为了需要解决的重要问题。对于医保信息的利用是本系统的特色之一，主要包括根据药品分类自动生成医保比例和根据医保比例计算自费金额两部分内容。前者已在文件录入中有所体现，后者则可以在医生开处方时提供方便。

(1)以下是医生在结算费用时的界面：



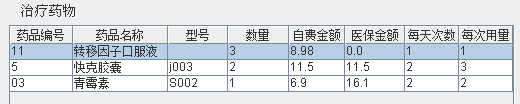




图23 有医保时费用结算

(2)高级程序代码中的关键部分如下：

//获取价格和医保比例并转化为可计算数据类型

String PHMPrice=vector5.elementAt(3).toString().trim();

**float** price=Float.parseFloat(PHMPrice);

String PHMTip=vector5.elementAt(4).toString().trim();

String PHMPercent=vector5.elementAt(5).toString().trim();

**float** percent=Float.parseFloat(PHMPercent);

**float** self=(price\*(100-percent))/100; //计算自费单价和医保单价vector7.add(""+self);

**float** a=Float.parseFloat(selfall); //结算

**int** num=Integer.parseInt(vector7.elementAt(3).toString().trim())

vector0.add(""+a\*num);sellAll=sellAll+a\*num;

### 7.3.3数据批量录入

由于在数据库构建初期以及每年新学生信息录入时，有大量格式一致的数据需要录入，所以本系统提供了文件录入功能。现以添加药物信息为例，分析文件录入功能：

(1)药物分类信息是从政府网站下载的Excel文件中转化而得，各属性具体数据后的特殊符号是为方便高级程序分析而后期添加的，具体录入文件实例如下：



图24 药品数据导入源文件

(2)在高级程序中，主要是由文件读取，按行分析，构造SQL，向数据库中添加四步操作实现文件录入功能。部分编程代码如下：

**public** **void** AddMed(String inform) //按行分析从文件中读取的一行信息并加入数据库中

{ **int** a=0,b=0;

a=inform.indexOf("#");//分析得药品名称

**try**{**if**(a==-1)System.*out*.println("this line no name");//本行无信息

**else**{name = inform.substring(0, a);

b=inform.indexOf("$");//分析得规格

**if**(b==-1) System.*out*.println("there is no type. ");

**else**{type = inform.substring(a+1, b).trim();}

... ...

a=inform.indexOf("类");//分析得类别

**if**(a==-1) System.*out*.println("there is no lei. ");

**else**{lei = inform.substring(b+1, a).trim();

**if**(lei.equals("甲"))per=80;**else** **if**(lei.equals("乙"))per=50;

**else** per=0;//分析药品分类，决定医保比例，按已定百分比录入数据库

}}}**catch**(Exception e) {}

}

**public** **void** doit()//打开文件，读取数据，添加至数据库

{ **try**

{ BufferedReader inputStream=**new** BufferedReader(**new** FileReader("input.txt"));

String temp=**null**;

//依据从数据库得到的信息，为新药分配编号

String getNo="select max(MNo)+2 from medicine";

FileDemo.open("sa", "");

ResultSet MN0Result=FileDemo.executeQuery(getNo);

MN0Result.next();

NO = MN0Result.getInt(1);

**for** (**int** i=1;i<=numberOfAll;i++)

{ temp = inputStream.readLine();

**this**.AddMed(temp);//调用按行分析方法，为新药物个属性赋值

String insertMdcicine=

"INSERT medicine(MNo,MName,MStyle,MPrice,Mtip,selfPercent) " +

"VALUES(" +NO+",'" +name+"','"+""+"','"+price

+"','每 "+unit+""+type+"','"+per+"')";

FileDemo.executeUpdate(insertMdcicine);

NO++;

}

FileDemo.close();

inputStream.close();

}**catch**(Exception e){}

}

# 8.测试和运行

根据软件生命周期，医务室服务系统测试被分为了单元测试，集成测试和系统测试三个阶段：

(1)单元测试：该阶段主要是针对在NetBeans生成的界面中各控件的监听操作和程序所需SQL语句进行的。在编码阶段完成后就着手进行的单元测试主要是为了快速定位错误以降低修改缺陷的成本。

(2)集成测试：该阶段主要是将本系统的各个部件的联合测试，以决定它们能否在一起共同工作。本阶段的测试工作包括：测试通过不同的用户身份是否可以正确登录相应用户界面；模拟不同的错误操作检验是否可以给出准确的提示；各种报表的存储位置和格式是否可以按要求实现等内容。

(3)系统测试：本阶段是在几名同学的帮助下完成的。首先向他们讲述了系统需求和所需操作方法；然后请他们使用该系统，测试尽可能多的情况；最后根据同学的测试感受主要对系统提示方式和内容方面作出了改进。

由于数据来源有限，在大量数据测试方面只对药品信息作了测试：200条新药物信息可以在3秒中内实现录入。考虑到药品在初次建立数据库后一般不会出现大于100条信息的同时录入，以上系统性能是可以接受的。

# 9.总结

本系统的特色主要是医保信息和过敏信息的管理，这一方面是考虑到医疗保险和人们日常生活的关系越来越密切，在大学生中实行医保制度已是必然；另一方面是为了实现对医疗记录和药物信息的充分利用。二者在系统中的具体功能体现已在上文中进行了描述。

由本次系统的开发过程经历了改进阶段，所以得到了很多的经验和教训。首先，本次开发的初期花费了很多时间做需求分析等准备工作，在数据库建立和程序编写阶段不但没有感觉时间紧，反而各项任务都能较有序进行。其次，不得不承认，虽然初期准备工作做得很多，在数据库E-R图编制和建表阶段还是遇到了障碍。现在分析一下，觉得一方面是因为数据库课程该部分内容掌握的不是很熟练，另一方面应该是经验不足造成了需求分析做得不够精准，特别是系统规模定位过大，使得最初设想的个别功能几经努力也没能实现。另外，和前几次系统开发过程相比，不细心的毛病虽然有所减轻，但还是因此浪费了不少时间，特别是SQL语句的where后所跟信息中“＝”和“like”的使用，在刚刚开始时未能注意到区别。最后，由于程序编写过程中，各变量、方法、类的命名都基本与自然语意一致，在方法调用和错误修改时感觉非常方便。

本系统最终提交的版本与前面版本的一个明显区别是界面有所扩大。以前一直认为小界面可以使各控件比较紧凑，经修改以后感觉其实较大的界面会操作起来更方便，看着更舒服。另外，最终版本在功能选择方式实现方面不再使用JTable提示方式，而是改用多个JPanel经选择有效实现的，这使得本系统操作更加简洁易懂。

总而言之，我很珍惜这次开发经历，因为在首次尝试的系统改进过程后，对界面，数据库等方面的内容有了新的认识，这对我以后的程序开发将会非常有帮助。

# 参考文献

1. 宋金玉，陈萍，陈刚.数据库原理与应用.2版.北京：清华大学出版社，2014
2. 路游，于与宗.数据库系统课程设计.北京：清华大学出版社，2009