中国药科大学

**《信息系统分析与设计》**

**课程设计报告**

信息系统与信息管理专业21级

组长：**2020212701潘炫伊**

组员： **2020212300王思璟**

2023-12-19

目录

[一、前言 3](#_Toc153900545)

[二、开发环境和开发方法 3](#_Toc153900546)

[三、需求分析 3](#_Toc153900547)

[3.1 业务描述和用例分析 3](#_Toc153900548)

[3.2 UML活动图分析 6](#_Toc153900549)

[3.3 领域对象分析 8](#_Toc153900550)

[四、系统设计 9](#_Toc153900551)

[4.1 数据库设计 9](#_Toc153900552)

[4.2系统类设计 10](#_Toc153900553)

[4.3 主要业务流程的面向对象设计 11](#_Toc153900554)

[4.4 结构功能设计 12](#_Toc153900555)

[五、系统实现 13](#_Toc153900556)

[5.1主要功能页面及实现过程 13](#_Toc153900557)

[5.2关键技术实现 17](#_Toc153900558)

[六、总结与展望 17](#_Toc153900559)

[附录：主要算法或代码 17](#_Toc153900560)

# 一、前言

我们小组的选题为医疗信息系统。医疗信息系统（Medical Information System）是指通过信息技术手段来管理和处理医疗机构的各种信息，包括患者信息、医疗记录、医疗设备信息等。

在过去，医疗行业主要依赖纸质档案和手工操作来管理患者信息和医疗记录。这种方式存在很多问题，如信息容易丢失、难以共享、操作效率低等。由于技术有限，我们组着重于对患者信息和药物信息的管理。系统主要是为了方便护士对患者信息和药房药品信息的查看，医院入职的护士可以通过系统注册认证身份，以此有权来查看患者和药品的信息，便于及时关注患者的症状变化和药物管理情况，节时节力。

医疗信息系统的前景非常广阔。随着信息技术的不断进步和医疗行业的不断发展，医疗信息系统将会越来越普及和应用。未来，医疗信息系统可能会结合人工智能、大数据分析等技术，实现更加智能化和个性化的医疗管理，更加高效地管理和处理医疗信息，提高医疗服务的质量和效率。

# 二、开发环境和开发方法

开放平台：JAVA

数据库：SQL Server

运行环境：Eclipse

开发方法：JAVA SWING、JDBC

# 三、需求分析

## 3.1 业务描述和用例分析

（1）参与者的确定

在医疗信息系统中，可以归纳出的问题有：

● 用户注册登录使用本系统（仅限录入系统信息的护士）

● 护士用户对患者信息进行管理

● 护士用户对药品信息进行管理

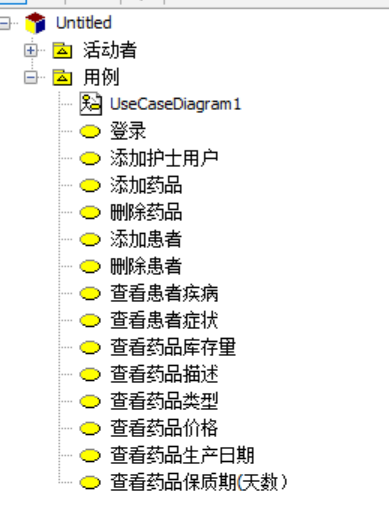
● 安全退出登录

根据以上问题，本系统所涉及的操作主要有：护士信息的管理、护士登录系统后，有权查看患者信息和药品信息。

在本例的UML建模中，可以创建以下角色：

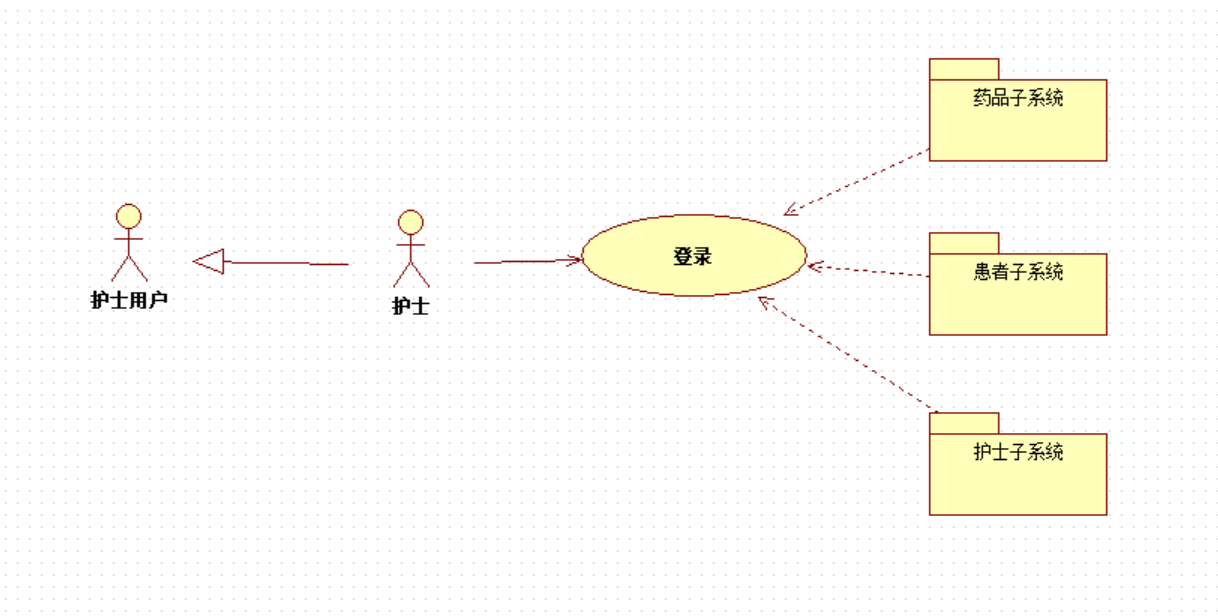


（2）创建用例



（3）创建参与者与用例的关系图

登录子系统用例图

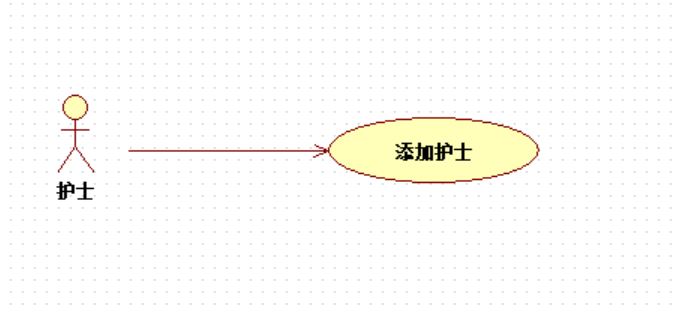


在登录子系统中，参与者包括了护士，在登录系统后包括患者子系统和药品子系统，

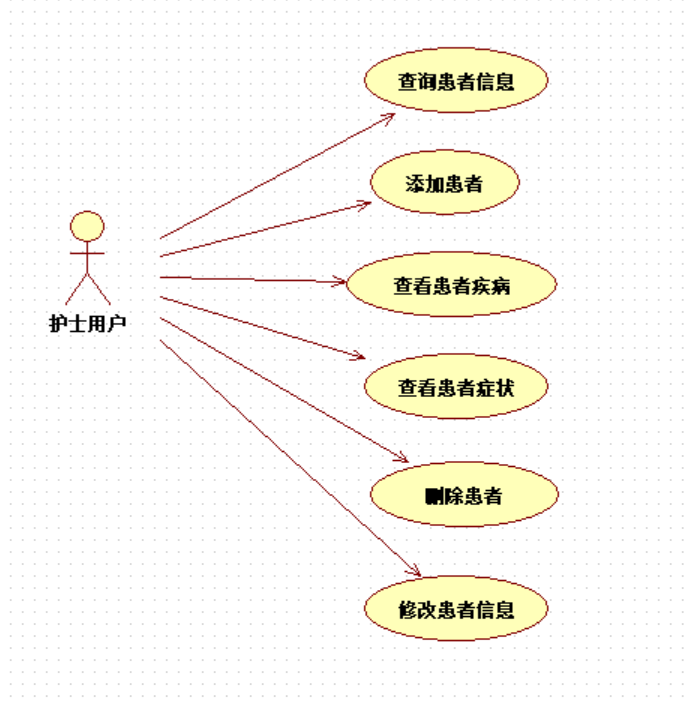
如果数据库中无该护士信息，护士需注册，包括护士子系统。

护士子系统用例图

在护士子系统用例图中，护士实现用户注册，添加新的护士用户。

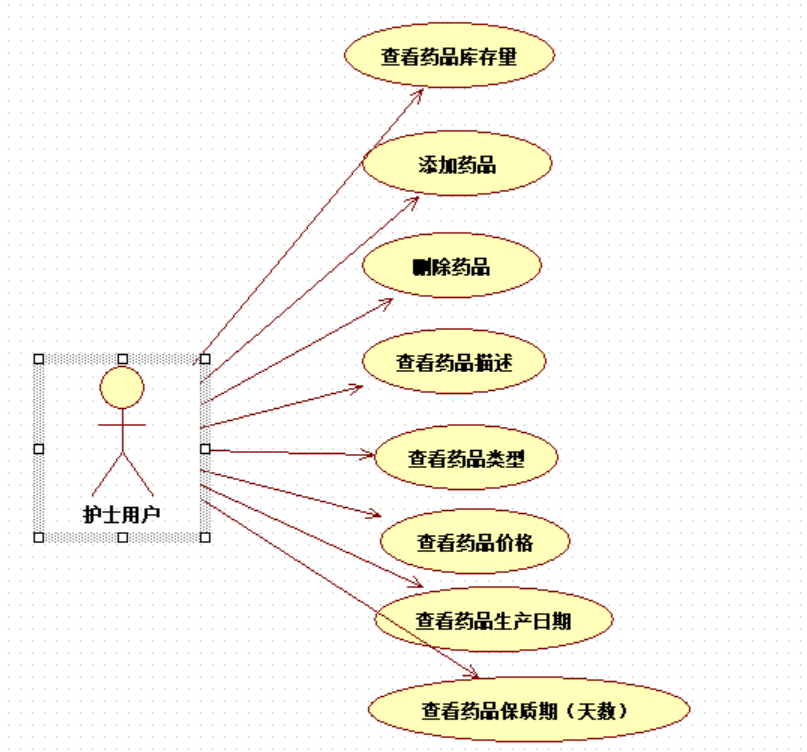


患者子系统用例图



在患者子系统用例图中，护士用户可以查看、修改、删除该患者的信息。

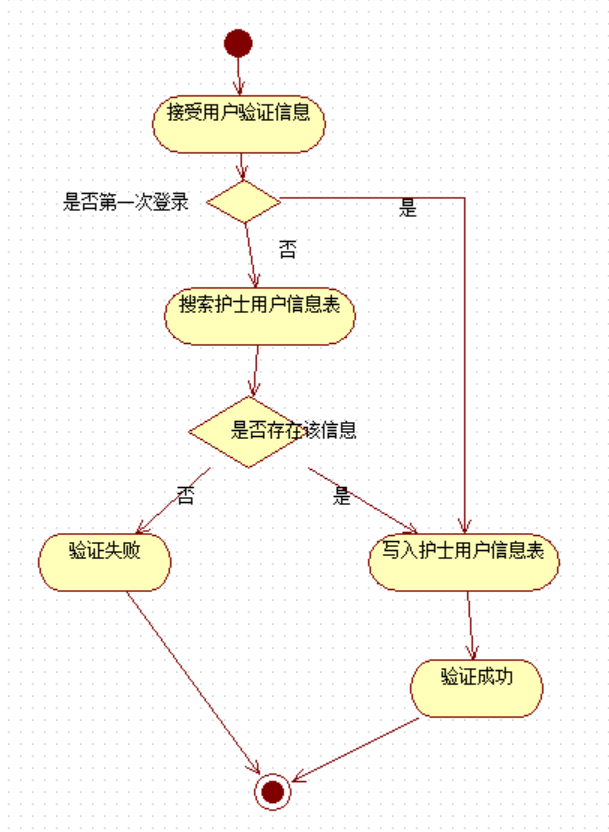
药品子系统用例图



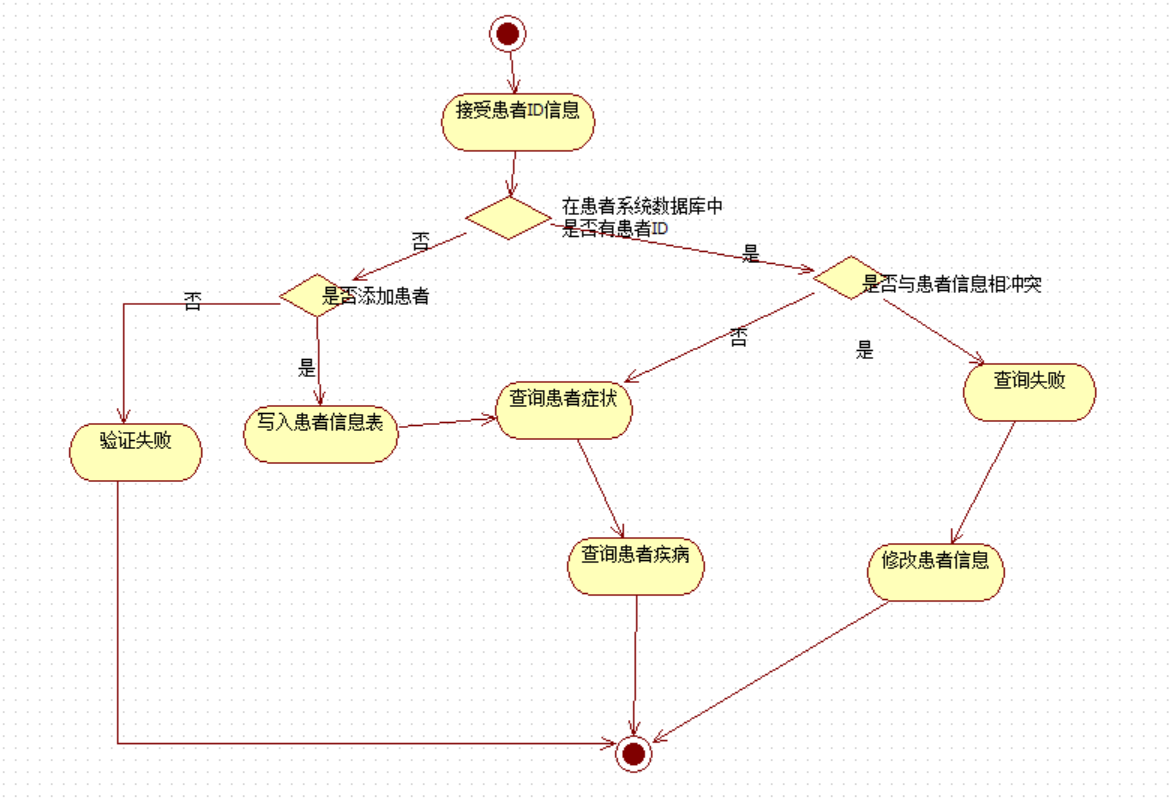
在药品子系统用例图中，护士用户可以查看、修改、删除该药品的信息。

## 3.2 UML活动图分析

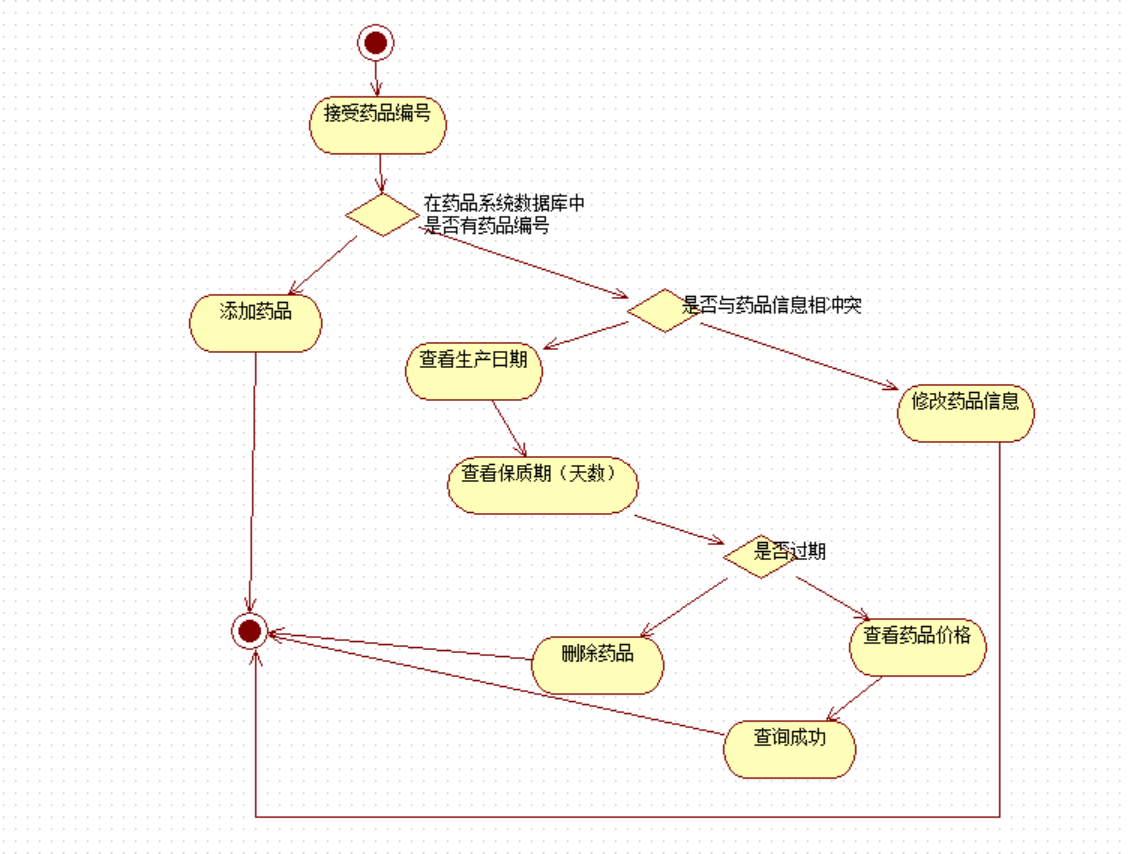
（1）验证登录身份活动图



（2）查看患者信息活动图



（3）查看药品信息活动图



## 3.3 领域对象分析

在医疗信息管理系统中，根据所提供的信息，可以分析出以下领域类：

Nurser（护士）

属性：id,name,age,sex,EnrollDate

方法：getNurserbyID, getNurUser(),isUserExist(),getMaxUserID(),

insertNurser(),insertNurUser()

Patient（患者）

属性：id,name,age,sex,disease,symptom,admission\_time

方法：getPatientbyID(),inserPatient()

Medicine（药品）

属性：id,name,storenumber,price,produceddate,expriationdate,address,decs,type

方法：getMedicinebyID(),inserMedicine(),deleteMedicinebyID()

# 四、系统设计

## 4.1 数据库设计

E-R图：

E-R图转换成关系模式：

1. 护士关系模式

护士信息（**护士id**、姓名、年龄、性别、入职时间）

护士用户信息（**护士用户id**、姓名、用户名、密码、护士id）

主键：护士id、护士用户id

外键：护士id

1. 患者关系模式

患者信息（**患者id**、姓名、年龄、性别、疾病、症状、入院时间）

主键：患者id

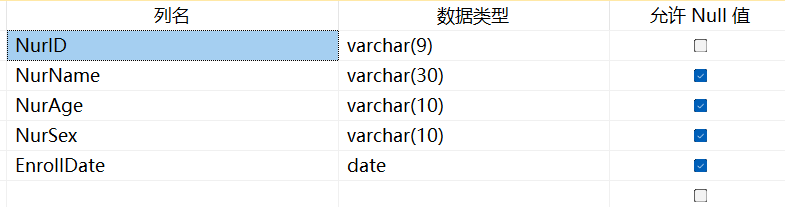
1. 药品关系模式

药品信息（**药品id**、药品名称、库存量、生产日期、保质期、生产地址、药品描述、药品类型）

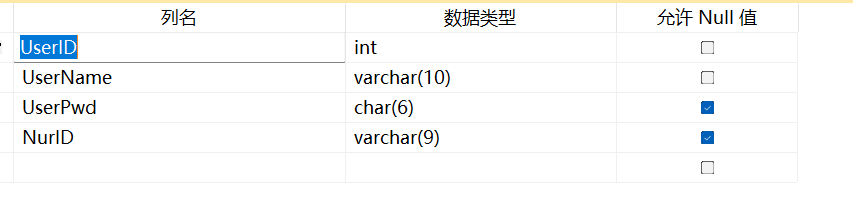
主键：药品id

数据库物理结构设计：

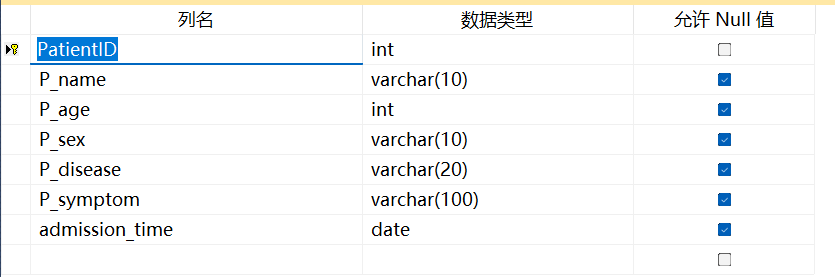
1. Nurser表



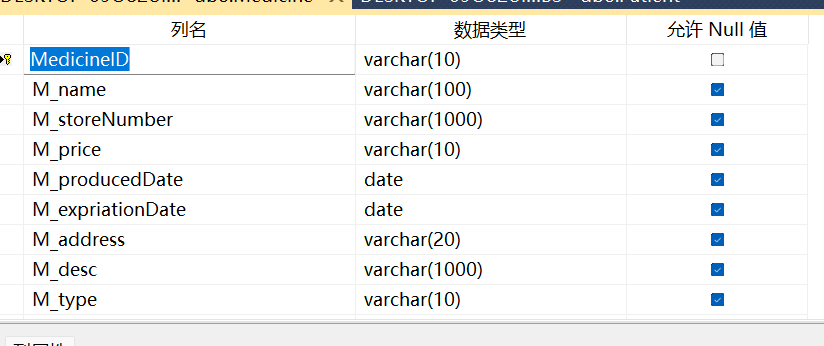
1. NurUser表



1. Patient表

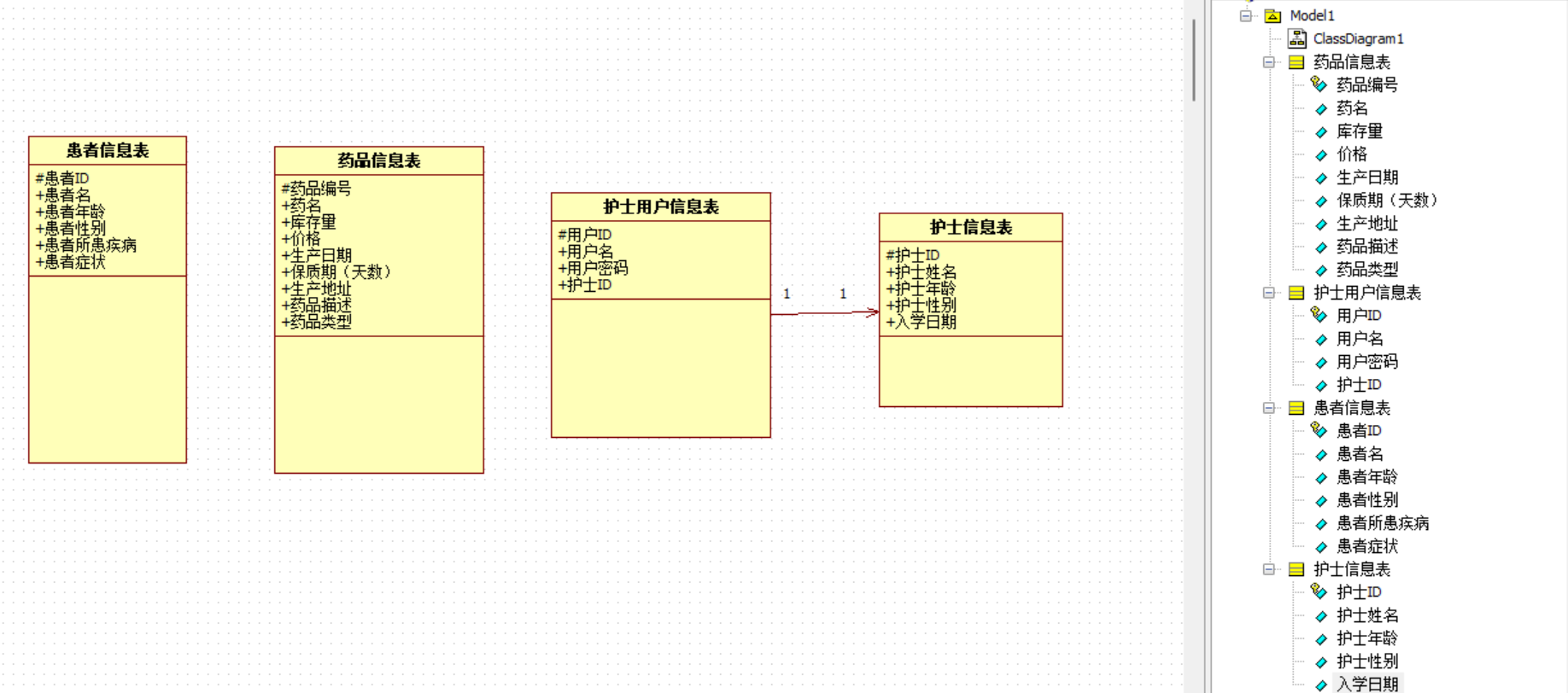


1. Medicine表



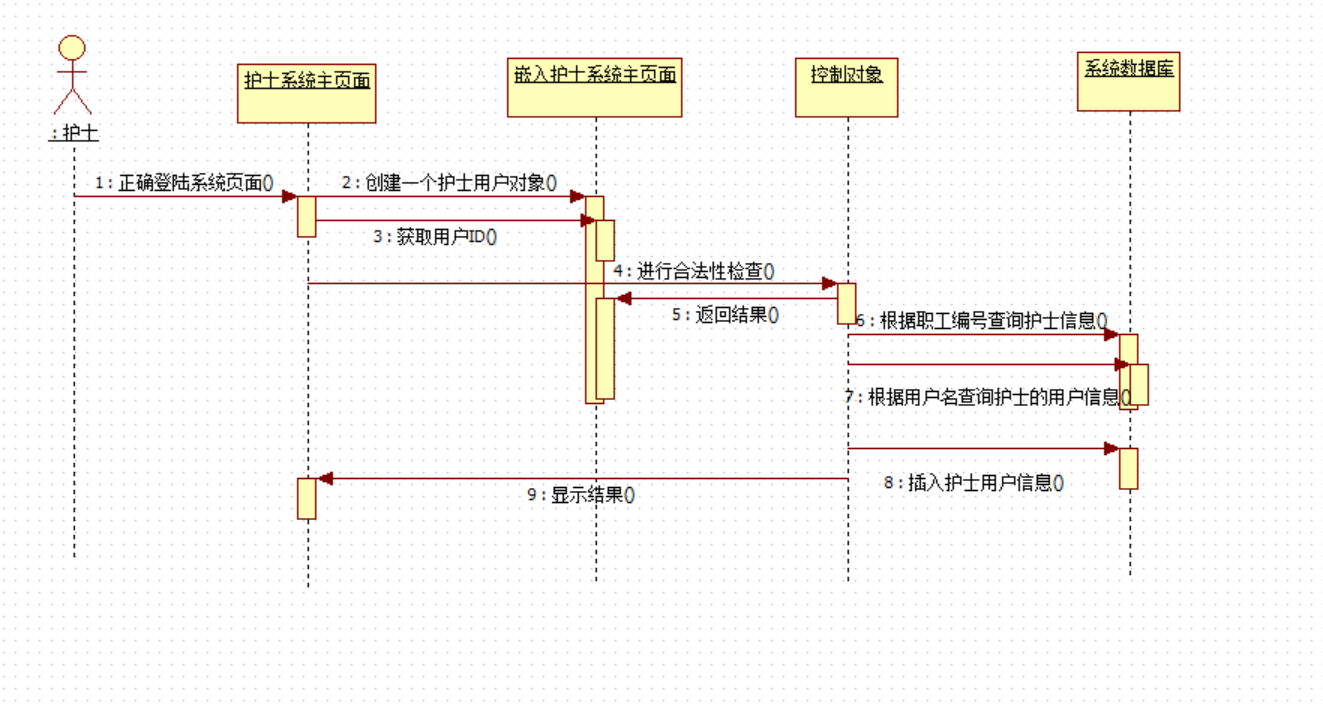
## 4.2系统类设计

类关系图：

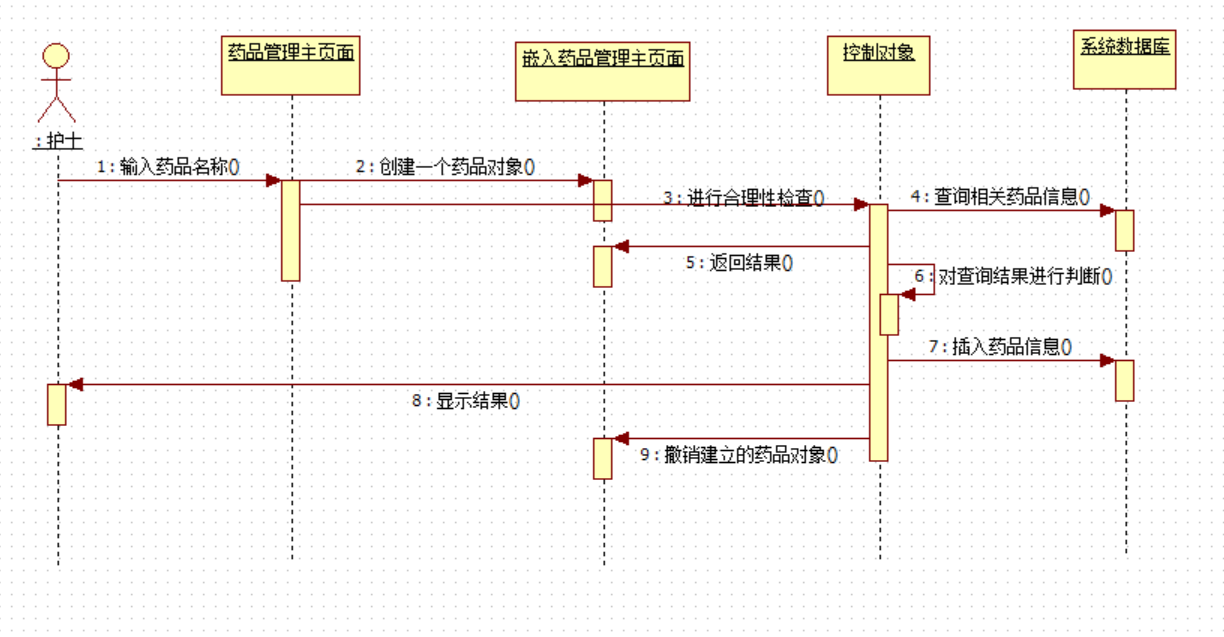


## 4.3 主要业务流程的面向对象设计

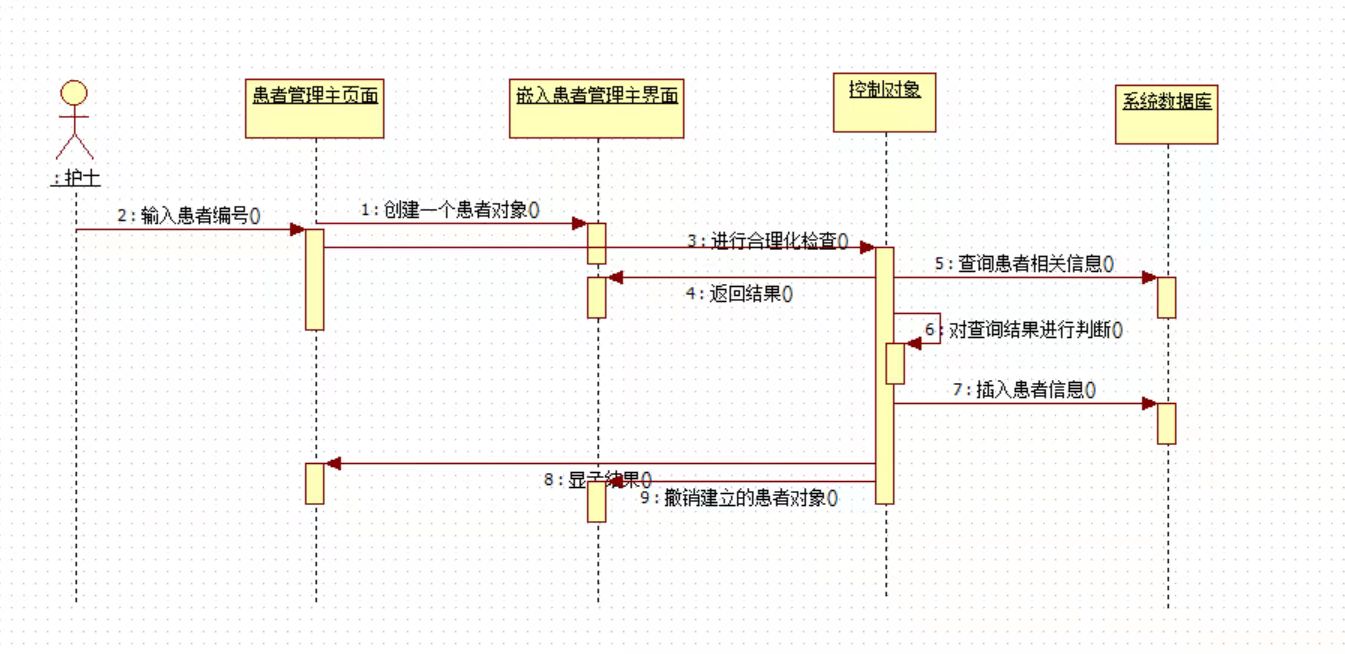
（1）护士管理序列图



（2）药品查询序列图

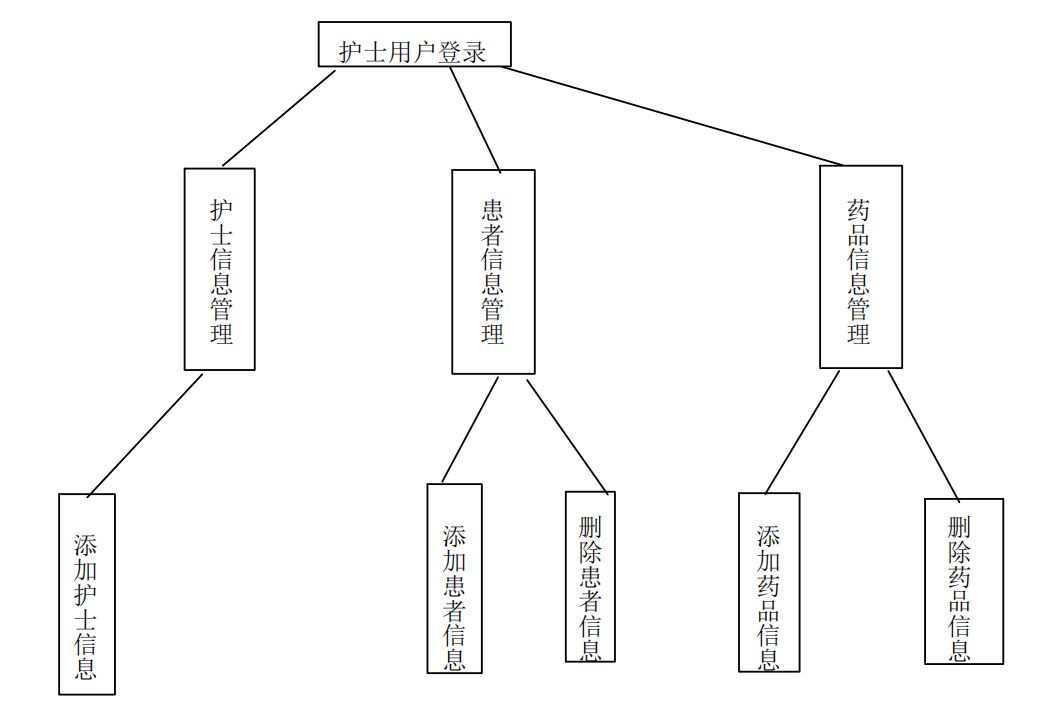


（3）患者查询序列图



## 4.4 结构功能设计

护士用户功能设计：



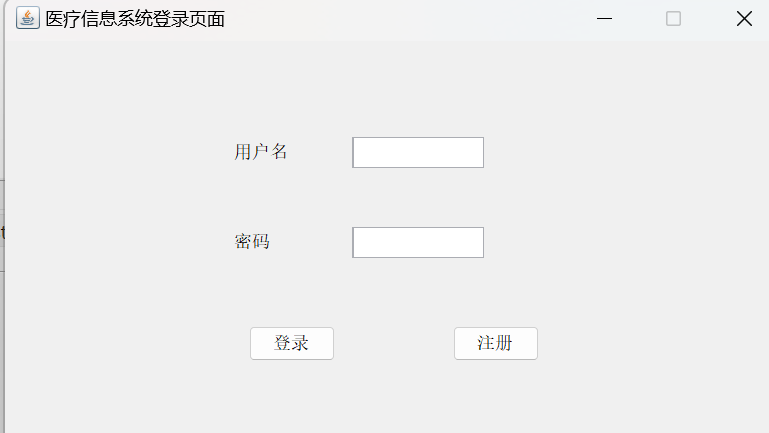
# 五、系统实现

## 5.1主要功能页面及实现过程

1.护士登录

登录：护士输入用户名和密码，系统与数据库交互查看是否有该用户。若有，实现系统登录。

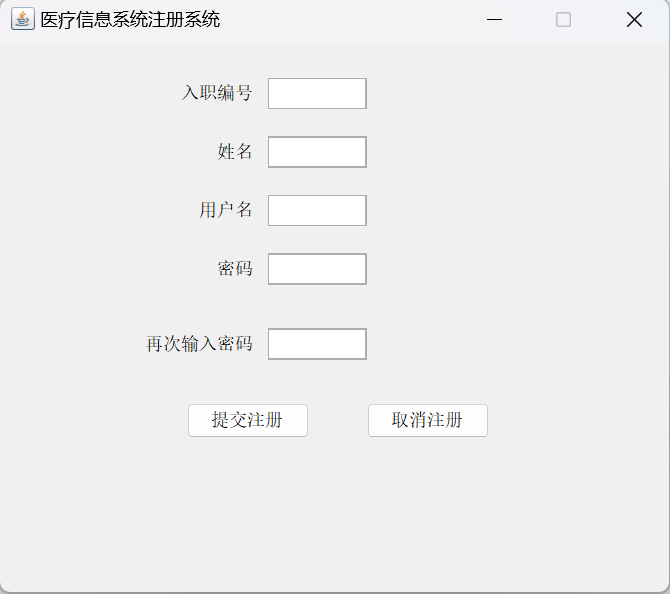
注册：跳转到注册界面



（2）护士注册

确定注册：提交护士注册信息，系统与数据库交互查看护士库中是否有该护士信息，注册成功返回登录界面。用户ID数据库自增字段。

取消注册：关闭界面。



（3）用户选择查看信息

患者：跳转到患者查询界面。

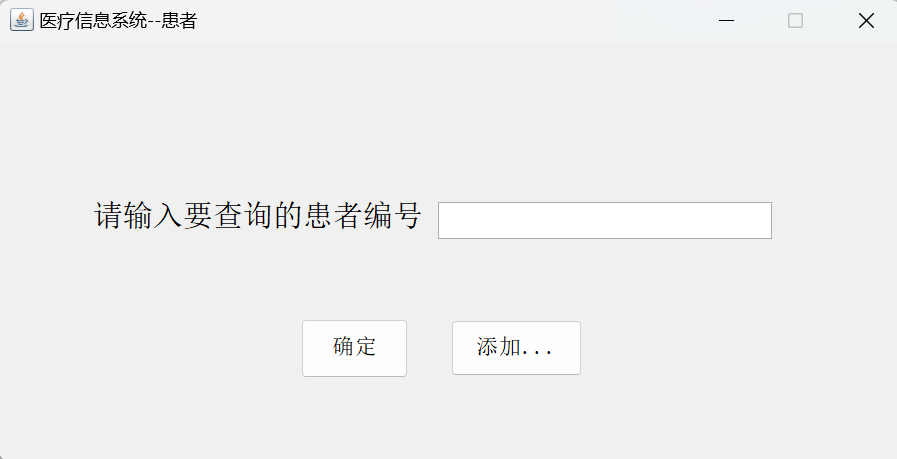
药品：跳转到药品查询界面



（4）患者查询界面

确定：用户输入患者入院的编号，系统与数据库交互查看患者库中是否有该患者信息，若有，跳转到患者信息页面

添加：跳转到用户对患者信息进行修改、添加、删除页面

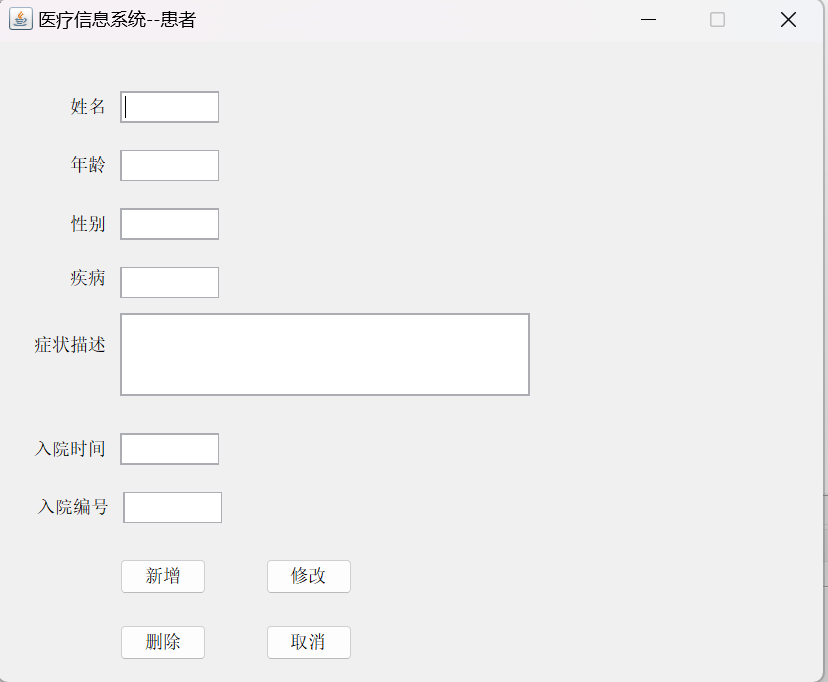


（5）药品查询界面

确定：用户输入药品编码，系统与数据库交互查看是否存在该药品，若有，跳转到药品信息页面。

添加：跳转到用户对药品信息实现修改、增加、修改操作页面

（6）患者管理页面



新增：用户添加患者信息，都需填满。

修改：用户根据患者入院编号可修改患者的相应信息。

删除：根据患者入院编号删除患者信息。

（7）药品管理信息界面

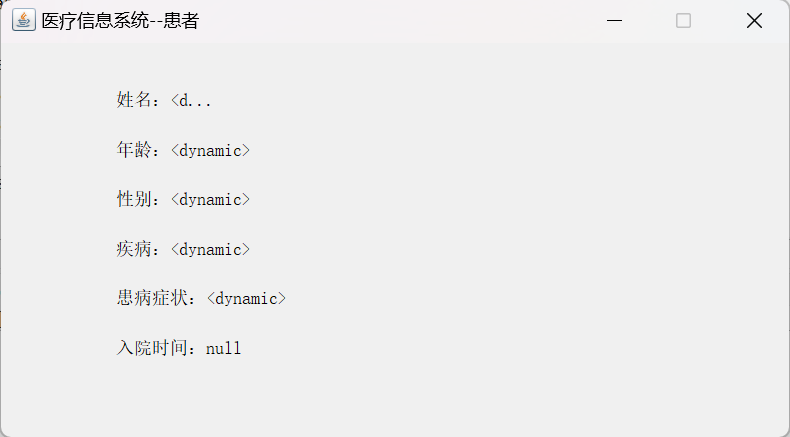


新增：用户添加药品信息

修改：用户根据药品编码可修改相应的药品信息

删除：根据主键药品编码删除药品信息

（8）患者信息界面



与数据库交互后可显示患者的信息

（9）药品信息界面



与数据库交互后可显示药品信息

## 5.2关键技术实现

1.技术亮点：

（1）Java Swing界面设计：使用Java Swing库来设计用户界面，包括文本框、标签、按钮等组件的布局和交互

（2）JDBC数据库连接：使用JDBC建立Java应用程序与数据库的连接，涉及到了加载数据库驱动、建立连接、创建和执行SQL语句等步骤。

（3）数据库的操作（新增、修改、查询、删除）：实现与数据库的交互。

（4）抽象类的设计：创建了人类抽象类，增加了代码的复用性和灵活性。

2.技术难点：

（1）UI界面和数据库的交互代码实现（eg.实现药品信息在界面中呈现）。

# 六、总结与展望

医疗信息系统在实际运用中存在着许多的不足，不足的着重在于系统的功能实现、安全性等方面。

功能实现：仅仅是护士登录系统查找、删除、修改用户患者、药品信息以及呈现具体的患者、药品信息的功能在实际需求和应用场景中时远远不够的，缺乏系统管理员、药房管理者参与和医疗记录、医疗设备记录管理功能等，而且功能较为固定模板，缺少灵活度。

数据处理：医疗信息系统面临的是海量的数据，该系统没有足够的数据处理和分析能力。

安全性：在连接数据库以及管理缺乏安全性的充分考虑。

解决这些问题还需提高对信息系统的开发能力和需求分析的充分考虑。

# 附录：主要算法或代码

**Entity:**

package Entity;

import java.text.SimpleDateFormat;

/\*\*

\* 人类抽象类

\*/

import java.util.Date;

public abstract class People\_Abstract {

private String id; //编号

private String name;//姓名

private String age;//年龄

private String sex;//性别

private String disease;//疾病

private String symptom;//症状

private Date EnrollDate;//入职时间

private Date AdmissionTime;//入院时间

/\*\*

\* 构造函数(nurser)

\* @param id

\* @param name

\* @param age

\* @param sex

\* @param EnrollDate

\*/

public People\_Abstract(String id,String name,String age,String sex,String EnrollDate) {

super();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

this.id=id;

this.name=name;

this.age=age;

this.sex=sex;

try {

if(EnrollDate!=null)

{

this.EnrollDate=sdf.parse(EnrollDate);

}

else {

this.EnrollDate=null;

}

} catch (Exception e) {

// TODO: handle exception

this.EnrollDate=null;

e.printStackTrace();

}

}

/\*\*

\* 构造函数（patient)

\* @param id

\* @param name

\* @param age

\* @param sex

\* @param disease

\* @param symptom

\* @param AdmissionTime

\*/

public People\_Abstract(String id,String name,String age,String sex,String disease,String symptom,String AdmissionTime) {

super();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

this.id=id;

this.name=name;

this.age=age;

this.sex=sex;

this.disease=disease;

this.symptom=symptom;

try {

if(AdmissionTime!=null)

{

this.AdmissionTime=sdf.parse(AdmissionTime);

}

else {

this.AdmissionTime=null;

}

} catch (Exception e) {

// TODO: handle exception

this.AdmissionTime=null;

e.printStackTrace();

}

}

public String getID() {

return id;

}

public void setID(String id) {

this.id=id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name=name;

}

public String getAge() {

return age;

}

public void setAge(String age) {

this.age=age;

}

public String getSex() {

return sex;

}

public void setSex(String sex) {

this.sex=sex;

}

public String getDisease() {

return disease;

}

public void setDisease(String disease) {

this.disease=disease;

}

public String getSymptom() {

return symptom;

}

public void setSymptom(String symptom) {

this.symptom=symptom;

}

public Date getEnrollDate() {

return EnrollDate;

}

public void setEnrollDate(Date EnrollDate) {

this.EnrollDate=EnrollDate;

}

public Date getAdmissionTime() {

return AdmissionTime;

}

public void setAdmissionTime(Date AdmissionTime) {

this.AdmissionTime=AdmissionTime;

}

public abstract String toString();

}

package Entity;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Nurser extends People\_Abstract {

public Nurser(String id,String name,String age,String sex,String EnrollDate) {

super(id,name,age,sex,EnrollDate);

}

public String toString() {

SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");

String nurserInfo;

nurserInfo="入职编号："+getID();

nurserInfo=nurserInfo+"\n姓名："+getName();

nurserInfo=nurserInfo+"\n入职日期："+sdf.format(getEnrollDate());

return nurserInfo;

}

}

package Entity;

import java.text.SimpleDateFormat;

public class Patient extends People\_Abstract {

public Patient(String id,String name,String age,String sex,String disease,String symptom,String AdmissionTime) {

super(id,name,age,sex,disease,symptom,AdmissionTime);

}

public String toString() {

SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");

String PatientInfo;

PatientInfo="入院编号："+getID();

PatientInfo=PatientInfo+"\n姓名："+getName();

PatientInfo=PatientInfo+"\n入院日期："+sdf.format(getEnrollDate());

PatientInfo=PatientInfo+"\n疾病："+getDisease();

PatientInfo=PatientInfo+"\n症状："+getSymptom();

return PatientInfo;

}

}

**DAO:**

package dao;

import dbUtils.IAccessDB;

import dbUtils.SqlServerJava;

/\*\*

\* 一个抽象类，所有DAO类都由此派生

\*

\*

\*/

public abstract class abstractDAO {

public String errorMessage=""; //错误信息

IAccessDB db=null; //数据库访问接口对象

//构造函数

public abstractDAO()

{

db=new SqlServerJava();//数据库访问类

errorMessage="";

}

//获得类内部发生的错误信息

public String getErrorMessage() {

return errorMessage;

}

//获得类内部发生的错误信息

public void setErrorMessage(String msg) {

errorMessage=msg;

}

}  
package dao;

import java.sql.ResultSet;

import Entity.NurUser;

import Entity.Nurser;

public class NurserDAO extends abstractDAO{

public NurserDAO() {

super();

}

/\*\*

\* 根据入职编号在数据库中查找该护士，查找成功返回护士对象，否则为null

\*@param nurid 入职编号

\*@return 查找成功返回护士对象，否则为null

\*/

public Nurser getNurserbyID(String nurid) {

errorMessage="";

Nurser nurser=null;//护士对象

String sql="select NurID,NurName,NurAge,NurSex,EnrollDate from Nurser";

sql=sql+"where NurID='"+nurid+"'";

ResultSet rs=db.getRsbySQL(sql);

try {

if(rs.next()) {

nurser=new Nurser(

rs.getString(1),

rs.getString(2),

rs.getString(3),

rs.getString(4),

rs.getString(5)

);

}

}catch(Exception e) {

errorMessage=e.getMessage();

}

return nurser;

}

/\*\*

\* 通过护士用户名获得用户对象

\* @userName 用户名

\* @return 如果用户名存在，则返回该用户的对象，否则返回null

\*/

public NurUser getNurUser(String userName) {

errorMessage="";

Nurser nurser=null;//护士对象

NurUser aNurUser=null;//护士用户对象

String sql="select UserID,UserName,UserPwd,NurID from NurUser";

sql=sql+"where UserName='"+userName+"'";

ResultSet rs=db.getRsbySQL(sql);

try {

if(rs.next()) {

String NurID=rs.getString(4);//获得NurID

nurser=this.getNurserbyID(NurID);

aNurUser=new NurUser(rs.getInt(1),rs.getString(2), rs.getString(3), nurser);

}

}catch(Exception e) {

errorMessage=e.getMessage();

}

return aNurUser;

}

/\*\*

\* 检查用户名是否存在

\* @param userName

\* @return

\*/

public boolean isUserExist(String userName) {

if(getNurUser(userName)!=null) {

return true;

}

else {

return false;

}

}

/\*\*

\* 获取用户ID最大号

\* @return

\*/

public int getMaxUserID() {

errorMessage="";

String sql="select MAX(userID)as maxID from NurUser";

return db.getMaxID(sql);

}

/\*\*插入一条护士记录

\* @param nurser

\* @return

\*

\*/

public boolean insertNurser(Nurser nurser) {

errorMessage="";

String sql="insert into Nurser(NurID,NurName,NurAge,NurSex,EnrollDate)values('";

sql=sql+nurser.getID()+"','";

sql=sql+nurser.getName()+"','";

sql=sql+nurser.getAge()+"','";

sql=sql+nurser.getSex()+"','";

sql=sql+nurser.getEnrollDate()+"')'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

errorMessage="写入护士信息数据不成功";

return false;

}

}

/\*\*插入一个护士用户记录

\* @param NurUser

\* @return

\*

\*/

public boolean insertNurUser(NurUser aNurUser) {

errorMessage="";

String sql="insert into NurUser(UserID,UserName,UserPwd,NurID)values('";

sql=sql+aNurUser.getUserID()+"','";

sql=sql+aNurUser.getUserName()+"','";

sql=sql+aNurUser.getUserPwd()+"','";

sql=sql+aNurUser.getNurInfo().getAge()+"')'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

return false;

}

}

//获得类内部发生的错误信息

public String getErrorMessage() {

return errorMessage;

}

}

package dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import Entity.Medicine;

/\*\*

\* 系统主界面药品操作 表名：Medicine

\*

\*

\*/

public class MedicalDao extends abstractDAO {

public MedicalDao() {

super();

}

/\*\*

\* 查询药品信息

\* @param id--药品编号

\*/

public Medicine getMedicinebyID(String id) {

errorMessage="";

Medicine medicine=null;//药品对象

String sql="select MedicineID,M\_name,M\_storeNumber,M\_price,M\_produceDate,M\_expriationDate,M\_address,M\_desc,M\_type from Medicine";

sql=sql+"where MedicineID='"+id+"'";

ResultSet rs=db.getRsbySQL(sql);

try {

if(rs.next()) {

medicine=new Medicine(rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getString(7),rs.getString(8),rs.getString(9));

}

}catch(Exception e) {

errorMessage=e.getMessage();

}

return medicine;

}

/\*\*

\* 删除药品信息

\* @param id

\* @return

\*/

public boolean deleteMedicinebyID(String id) {

errorMessage="";

Medicine medicine=null;

String sql="deleten from Medicine where MedicineID='"+id+"'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

errorMessage="删除药品信息失败!";

return false;

}

}

/\*\*

\* 药品信息操作之药品添加

\* @param medicine

\* @return

\*/

public boolean inserMedicine(Medicine medicine) {

errorMessage="";

String sql="inser into Medicine(MedicineID,M\_name,M\_storeNumber,M\_price,M\_produceDate,M\_expriationDate,M\_address,M\_desc,M\_type)values('";

sql=sql+medicine.getId()+"','";

sql=sql+medicine.getStoreNumber()+"','";

sql=sql+medicine.getPrice()+"','";

sql=sql+medicine.getProducedDate()+"','";

sql=sql+medicine.getExpirationDate()+"','";

sql=sql+medicine.getAddress()+"','";

sql=sql+medicine.getMedicinesDesc()+"','";

sql=sql+medicine.getMedicalType()+"')'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

errorMessage="写入药品信息数据不成功";

return false;

}

}

/\*\*

\* 更改药品信息

\*/

public boolean updateMedicine(Medicine medicine) {

errorMessage="";

String sql="update Medicine set M\_storeNumber='";

sql=sql+medicine.getStoreNumber()+"where MedicineID='"+medicine.getId()+"'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

errorMessage="更新药物信息失败！";

return false;

}

}

//获得类内部发生的错误信息

public String getErrorMessage() {

return errorMessage;

}

}

package dao;

import java.sql.ResultSet;

import Entity.Medicine;

import Entity.Patient;

public class PatientDAO extends abstractDAO{

public PatientDAO() {

super();

}

/\*\*

\* 根据入院编号在数据库查找该患者

\* @param patient\_id 入院编号

\* @return 查找成功返回患者对象，否则null

\*/

public Patient getPatientbyID(String patient\_id) {

errorMessage="";

Patient patient=null;//患者对象

String sql="select PatientID,P\_name,P\_age,P\_sex,P\_disease,P\_sympton,admission\_time from Patient";

sql=sql+"where PatientID"+patient\_id+"'";

ResultSet rs=db.getRsbySQL(sql);

try {

if(rs.next()) {

patient=new Patient(

rs.getString(1),

rs.getString(2),

rs.getString(3),

rs.getString(4),

rs.getString(5),

rs.getString(6),

rs.getString(7)

);

}

}catch(Exception e) {

errorMessage=e.getMessage();

}

return patient;

}

/\*\*

\* 添加新的患者

\* @param patient

\* @return

\*/

public boolean inserPatient(Patient patient) {

errorMessage="";

String sql="insert into Patinet(PatientID,P\_name,P\_age,P\_sex,P\_disease,P\_sympton,admission\_time)values('";

sql=sql+patient.getID()+"','";

sql=sql+patient.getName()+"','";

sql=sql+patient.getAge()+"','";

sql=sql+patient.getSex()+"','";

sql=sql+patient.getDisease()+"','";

sql=sql+patient.getSymptom()+"','";

sql=sql+patient.getAdmissionTime()+"')'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

return false;

}

}

/\*\*

\* 删除患者

\* @id --患者id

\* @return

\*/

public boolean deletePatientbyID(String patient\_id) {

errorMessage="";

Patient patient=null;

String sql="deleten from Patient where MedicineID='"+patient\_id+"'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

errorMessage="删除患者信息失败!";

return false;

}

}

/\*\*

\* 修改患者

\* @param medicine

\* @return

\*/

public boolean updatePatient(Patient patient) {

errorMessage="";

String sql="update Patient set M\_disease='";

sql=sql+patient.getDisease()+"where PatientID='"+patient.getID()+"'";

if(db.ExcuteSQL(sql)==1) {

return true;

}

else {

errorMessage="更新药物信息失败！";

return false;

}

}

public String getErrorMessage() {

return errorMessage;

}

}

**BLL:**

package BLL;

import Entity.NurUser;

import Entity.Nurser;

import dao.NurserDAO;

/\*\*

\* 护士服务类--逻辑层

\* @author Arashi

\*

\*/

public class NurserService {

private String message="";

NurserDAO dao;

public NurserService() {

dao=new NurserDAO();

}

/\*\*

\* 护士用户登录

\* @Parma userName--用户名

\* @param userPwd --密码

\* @return 登录成功，返回一个护士用户，否则null

\*/

public NurUser login(String userName,String userPwd) {

//从数据库中获得学生用户对象

NurUser curNurUser=dao.getNurUser(userName);

if(curNurUser==null) {

message="用户名不存在！";

return null;

}

else {

if(!curNurUser.getUserPwd().equals(userPwd)) {

message="密码不正确！";

return null;

}

}

return curNurUser;

}

/\*\*

\* 护士注册

\* @param nurID

\* @param nurName

\* @param userName

\* @param userPwd

\* @return

\*/

public boolean register(String nurID,String nurName,String userName,String userPwd) {

//入职编号和姓名一致检查

Nurser nurser=dao.getNurserbyID(nurID);

if(nurser==null) {

message="护士信息库中没有该护士！注册不成功！";

return false;

}

if(!nurser.getName().equals(nurName)) {

message="护士信息有错误！注册不成功！";

return false;

}

//用户名不能重名

if(dao.getNurUser(userName)!=null) {

message="用户名已存在！注册失败！";

return false;

}

//一个护士只能注册一次

if(dao.getNurserbyID(nurID)!=null) {

message="该护士已注册！注册不成功!";

return false;

}

//获得当前用户ID

int UserID=dao.getMaxUserID();

//构造护士用户

NurUser nuruser=new NurUser(UserID,userName,userPwd,nurser);

if(!dao.insertNurUser(nuruser)){

message="写入数据时出现问题，注册失败";

return false;

}

else {

return true;

}

}

public String getMessage() {

return message;

}

}

package BLL;

import Entity.Medicine;

import dao.MedicalDao;

/\*\*

\* 药品服务类--逻辑层

\* @author Arashi

\*

\*/

public class MedicineService {

private String message="";

MedicalDao dao;

public MedicineService() {

dao=new MedicalDao();

}

/\*\*

\* 药品系统登入

\* @param id--药品编码

\* @return 进入成功，返回一个药品信息，否则null

\*/

public Medicine login(String id) {

Medicine medicine=dao.getMedicinebyID(id);

if(medicine==null) {

message="该药品不存在！";

return null;

}

return medicine;

}

/\*\*

\* 药品系统录入新药

\* @param id

\* @param name

\*

\*/

public boolean register(String id, String name, String storeNumber, String price, String producedDate,

String expirationDate, String address, String medicinesDesc, String medicalType ) {

//不能有重复的药品录入

if(dao.getMedicinebyID(id)!=null) {

message="该药品已存在！注册失败！";

return false;

}

Medicine medicine=new Medicine(id,name,storeNumber,price,producedDate,expirationDate,address,medicinesDesc,medicalType);

if(!dao.inserMedicine(medicine)) {

message="在写入数据时出现问题，注册失败！";

return false;

}

else {

return true;

}

}

public String getMessage() {

return message;

}

}

package BLL;

/\*\*

\* 患者服务层

\*/

import Entity.Medicine;

import Entity.Patient;

import dao.PatientDAO;

public class PatientService {

private String message="";

PatientDAO dao;

public PatientService() {

dao=new PatientDAO();

}

/\*\*

\* 进入患者信息

\* @param id--患者入院编号

\* @return 进入成功，返回一个患者，否则null

\*/

public Patient login(String id) {

Patient patient=dao.getPatientbyID(id);

if(patient==null) {

message="系统中无该患者！";

return null;

}

return patient;

}

/\*\*

\* 录入新患者的信息

\*

\*/

public boolean register(String id,String name,String age,String sex,String disease,String symptom,String AdmissionTime) {

if(dao.getPatientbyID(id)!=null) {

message="该患者已录入系统！注册不成功！";

return false;

}

Patient patient=new Patient(id,name,age,sex,disease,symptom,AdmissionTime);

if(!dao.inserPatient(patient)) {

message="写入数据时出现错误！注册不成功！";

return false;

}

else {

return true;

}

}

public String getMessage() {

// TODO Auto-generated method stub

return message;

}

}

**UI:**

package View;

import java.awt.BorderLayout;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import BLL.NurserService;

import Entity.NurUser;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class loginUI extends JFrame {

private JPanel contentPane;

private JTextField txtUserName;

private JTextField txtUserPwd;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

loginUI frame = new loginUI();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public loginUI() {

setTitle("医疗信息系统登录页面");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 498, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("用户名");

txtUserName = new JTextField();

txtUserName.setColumns(10);

JLabel lblNewLabel\_1 = new JLabel("密码");

txtUserPwd = new JTextField();

txtUserPwd.setColumns(10);

JButton btnLogin = new JButton("登录");

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

login(e);

}

}

);

JButton btnRegister = new JButton("注册");

btnRegister.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

register(e);//护士注册

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(148)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel, GroupLayout.PREFERRED\_SIZE, 75, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_1))

.addPreferredGap(ComponentPlacement.RELATED)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addComponent(txtUserPwd)

.addComponent(txtUserName, GroupLayout.DEFAULT\_SIZE, 88, Short.MAX\_VALUE)))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(158)

.addComponent(btnLogin)

.addGap(79)

.addComponent(btnRegister)))

.addContainerGap(156, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(48)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel, GroupLayout.PREFERRED\_SIZE, 43, GroupLayout.PREFERRED\_SIZE)

.addComponent(txtUserName, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(28)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_1)

.addComponent(txtUserPwd, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(45)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnLogin)

.addComponent(btnRegister))

.addContainerGap(51, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

/\*\*

\* 注册

\*/

private void register(ActionEvent e) {

// TODO Auto-generated method stub

new registerUI(this).setVisible(true);//启动注册窗口

this.setVisible(false);

}

/\*\*

\* 登录

\* @param userName

\* @param userpwd

\*/

private void login(ActionEvent e) {

// TODO Auto-generated method stub

String userName=txtUserName.getText();

String userpwd=txtUserPwd.getText();

NurserService aService=new NurserService();

NurUser curNurUser=aService.login(userName, userpwd);

if(curNurUser==null) {

JOptionPane.showMessageDialog(null, "登录失败！"+aService.getMessage(),"提示信息",1);

return ;

}

//登录成功

new SelectUI().setVisible(true);;

this.dispose();

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import BLL.NurserService;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class registerUI extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private JTextField txtID;

private JTextField txtName;

private JTextField txtUsername;

private JTextField txtPwd;

private JTextField txtCheckPwd;

private JButton btnOK,btnCancel;

private static loginUI loginui=null;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

registerUI frame = new registerUI(loginui);

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public registerUI(loginUI loginui) {

this.loginui=loginui;

setTitle("医疗信息系统注册系统");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 461, 356);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("入职编号");

txtID = new JTextField();

txtID.setColumns(10);

txtName = new JTextField();

txtName.setColumns(10);

txtUsername = new JTextField();

txtUsername.setColumns(10);

txtPwd = new JTextField();

txtPwd.setColumns(10);

txtCheckPwd = new JTextField();

txtCheckPwd.setColumns(10);

JLabel lblNewLabel\_1 = new JLabel("姓名");

JLabel lblNewLabel\_2 = new JLabel("用户名");

JLabel lblNewLabel\_4 = new JLabel("密码");

JLabel lblNewLabel\_5 = new JLabel("再次输入密码");

btnOK = new JButton("提交注册");

btnOK.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

register(e);

}

});

btnCancel = new JButton("取消注册");

btnCancel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

cancel(e);

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(92)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addComponent(lblNewLabel)

.addComponent(lblNewLabel\_1)

.addComponent(lblNewLabel\_2)

.addComponent(lblNewLabel\_4)

.addComponent(lblNewLabel\_5))

.addPreferredGap(ComponentPlacement.UNRELATED)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(txtCheckPwd, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(txtPwd, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(txtUsername, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(txtName, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(txtID, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(120)

.addComponent(btnOK)

.addGap(39)

.addComponent(btnCancel)))

.addContainerGap(116, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(19)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel)

.addComponent(txtID, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(txtName, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_1))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(txtUsername, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_2))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_4)

.addComponent(txtPwd, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(ComponentPlacement.RELATED, 29, Short.MAX\_VALUE)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_5)

.addComponent(txtCheckPwd, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(29)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnOK)

.addComponent(btnCancel))

.addGap(98))

);

contentPane.setLayout(gl\_contentPane);

}

private void cancel(ActionEvent e) {

// TODO Auto-generated method stub

loginui.setVisible(true);

dispose();

}

private void register(ActionEvent e) {

// TODO Auto-generated method stub

String nurID=txtID.getText();

String nurName=txtName.getText();

String userName=txtUsername.getText();

String userPwd1=txtPwd.getText();

String userPwd2=txtCheckPwd.getText();

Register(nurID,nurName,userName,userPwd1,userPwd2);

}

//注册

private void Register(String nurID, String nurName, String userName, String userPwd1, String userPwd2) {

//检查数据不能为空

if(checkEmpty(nurID,nurName, userName,userPwd1, userPwd2)) {

JOptionPane.showMessageDialog(null,"注册不成功！","提示",1);

return;

}

if(!checktwoPwd(userPwd1,userPwd2))

{

JOptionPane.showMessageDialog(null,"注册不成功！","提示",1);

return;

}

NurserService op=new NurserService();

if(op.register(nurID, nurName, userName, userPwd1)) {

loginui.setVisible(true);

dispose();

}

else {

JOptionPane.showMessageDialog(null,op.getMessage(),"提示",1);

return;

}

}

/\*\*

\* 检查是否有空字段

\* @param nurID

\* @param nurName

\* @param userName

\* @param userPwd1

\* @param userPwd2

\* @return

\*/

private boolean checkEmpty(String nurID, String nurName, String userName, String userPwd1, String userPwd2) {

boolean flag=false;

if (nurID.isEmpty()) flag=true ;

if(nurName.isEmpty()) flag=true ;

if(userName.isEmpty()) flag=true;

if(userPwd1.isEmpty()) flag=true;

if(userPwd2.isEmpty()) flag=true;

if(flag)

{

JOptionPane.showMessageDialog(null,"有空字段没有填写","提示",1);

}

return flag;

}

/\*\*

\* 检查密码是否一致

\* @param userPwd1

\* @param userPwd2

\* @return

\*/

private boolean checktwoPwd(String userPwd1, String userPwd2) {

// TODO Auto-generated method stub

if(userPwd1.equals(userPwd2))

{

return true;

}

else

{

JOptionPane.showMessageDialog(null,"两次密码不相同","提示",1);

return false;

}

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class SelectUI extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

SelectUI frame = new SelectUI();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public SelectUI() {

setTitle("医疗信息系统--护士");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JButton btnMedicine = new JButton("药物信息");

btnMedicine.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Medicine(e);

}

});

JButton btnPatient = new JButton("患者信息");

btnPatient.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Patient(e);

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(49)

.addComponent(btnMedicine, GroupLayout.PREFERRED\_SIZE, 114, GroupLayout.PREFERRED\_SIZE)

.addGap(77)

.addComponent(btnPatient, GroupLayout.PREFERRED\_SIZE, 104, GroupLayout.PREFERRED\_SIZE)

.addContainerGap(82, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(106)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnMedicine, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnPatient, GroupLayout.PREFERRED\_SIZE, 30, GroupLayout.PREFERRED\_SIZE))

.addContainerGap(110, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

protected void Patient(ActionEvent e) {

// TODO Auto-generated method stub

new PatientQuery().setVisible(true);//打开新界面

this.setVisible(false);

}

protected void Medicine(ActionEvent e) {

// TODO Auto-generated method stub

new MedicineQuery().setVisible(true);//打开新界面

this.setVisible(false);

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import BLL.PatientService;

import Entity.Patient;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class PatientQuery extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private JTextField textID;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

PatientQuery frame = new PatientQuery();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public PatientQuery() {

setTitle("医疗信息系统--患者");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("请输入要查询的患者编号");

lblNewLabel.setFont(new Font("宋体", Font.PLAIN, 20));

textID = new JTextField();

textID.setColumns(10);

JButton btnOK = new JButton("确定");

btnOK.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

login(e);

}

});

btnOK.setFont(new Font("宋体", Font.PLAIN, 15));

JButton btnAdd = new JButton("添加...");

btnAdd.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

Add(e);

}

});

btnAdd.setFont(new Font("宋体", Font.PLAIN, 15));

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING, false)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(58)

.addComponent(lblNewLabel))

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addContainerGap(GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(btnOK, GroupLayout.PREFERRED\_SIZE, 71, GroupLayout.PREFERRED\_SIZE)

.addGap(10)))

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, 223, GroupLayout.PREFERRED\_SIZE))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(19)

.addComponent(btnAdd)))

.addContainerGap(81, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(99)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, 25, GroupLayout.PREFERRED\_SIZE))

.addGap(53)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnAdd, GroupLayout.PREFERRED\_SIZE, 37, GroupLayout.PREFERRED\_SIZE)

.addComponent(btnOK, GroupLayout.PREFERRED\_SIZE, 39, GroupLayout.PREFERRED\_SIZE))

.addContainerGap(51, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

private void login(ActionEvent e) {

String patient\_id=textID.getText();

PatientService aService=new PatientService();

Patient patient=aService.login(patient\_id);

if(patient==null) {

JOptionPane.showMessageDialog(null, "查看失败!"+aService.getMessage(),"提示信息",1);

return ;

}

//进入成功

new PatientUI(patient).setVisible(true);;

this.setVisible(false);

}

private void Add(ActionEvent e) {

new addPatient(this).setVisible(true);;

this.setVisible(false);

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import BLL.MedicineService;

import Entity.Medicine;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class MedicineQuery extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private JTextField textID;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

MedicineQuery frame = new MedicineQuery();

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public MedicineQuery() {

setTitle("医疗信息系统--药物");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("请输入药品的编号");

lblNewLabel.setFont(new Font("宋体", Font.PLAIN, 20));

textID = new JTextField();

textID.setColumns(10);

JButton btnOK = new JButton("确定");

btnOK.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

login(e);

}

});

btnOK.setFont(new Font("宋体", Font.PLAIN, 15));

JButton btnCancel = new JButton("添加...");

btnCancel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

register(e);

}

});

btnCancel.setFont(new Font("宋体", Font.PLAIN, 15));

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(88, Short.MAX\_VALUE)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addComponent(btnOK)

.addGap(68))

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addComponent(lblNewLabel)

.addGap(33)))

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(btnCancel)

.addGroup(gl\_contentPane.createSequentialGroup()

.addPreferredGap(ComponentPlacement.RELATED)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, 131, GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(132, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addContainerGap(125, Short.MAX\_VALUE)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(47)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnOK)

.addComponent(btnCancel))

.addGap(43))

);

contentPane.setLayout(gl\_contentPane);

}

private void login(ActionEvent e) {

String id=textID.getText();

MedicineService aService=new MedicineService();

Medicine medicine=aService.login(id);

if(medicine==null) {

JOptionPane.showMessageDialog(null, "查询失败!"+aService.getMessage(),"提示信息",1);

return ;

}

//查询成功

new MedicineUI(medicine).setVisible(true);;

this.setVisible(false);

}

private void register(ActionEvent e) {

new addMedicine(this).setVisible(true);;

this.setVisible(false);

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import BLL.PatientService;

import Entity.Patient;

import dao.PatientDAO;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JTextField;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class addPatient extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private JTextField textName;

private JTextField textAge;

private JTextField textSex;

private JTextField textDisease;

private JTextField textSymptom;

private static PatientQuery query;

private JLabel lblNewLabel\_6;

private JTextField textAdmissionTime;

private JLabel lblNewLabel\_7;

private JTextField textID;

private JButton btnModify;

private JButton btnDelete;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

addPatient frame = new addPatient(query);

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public addPatient(PatientQuery query) {

this.query=query;

setTitle("医疗信息系统--患者");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 565, 464);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("姓名");

textName = new JTextField();

textName.setColumns(10);

JLabel lblNewLabel\_1 = new JLabel("");

textAge = new JTextField();

textAge.setColumns(10);

JLabel lblNewLabel\_2 = new JLabel("年龄");

JLabel lblNewLabel\_3 = new JLabel("性别");

textSex = new JTextField();

textSex.setColumns(10);

JLabel lblNewLabel\_4 = new JLabel("疾病");

textDisease = new JTextField();

textDisease.setColumns(10);

JLabel lblNewLabel\_5 = new JLabel("症状描述");

textSymptom = new JTextField();

textSymptom.setColumns(10);

JButton btnOK = new JButton("新增");

btnOK.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

add(e);

}

});

JButton btnCancel = new JButton("取消");

btnCancel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

cancel(e);

}

});

lblNewLabel\_6 = new JLabel("入院时间");

textAdmissionTime = new JTextField();

textAdmissionTime.setColumns(10);

lblNewLabel\_7 = new JLabel("入院编号");

textID = new JTextField();

textID.setColumns(10);

btnModify = new JButton("修改");

btnModify.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

modify(e);

}

});

btnDelete = new JButton("删除");

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

delete(e);

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(Alignment.TRAILING, gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addContainerGap(19, Short.MAX\_VALUE)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addComponent(lblNewLabel\_1)

.addGap(27))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addComponent(lblNewLabel\_3)

.addComponent(lblNewLabel\_2)

.addComponent(lblNewLabel)

.addComponent(lblNewLabel\_4)

.addComponent(lblNewLabel\_5)

.addComponent(lblNewLabel\_6))

.addPreferredGap(ComponentPlacement.UNRELATED)))

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(textAge, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textName, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textSex, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textDisease, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textSymptom, GroupLayout.PREFERRED\_SIZE, 273, GroupLayout.PREFERRED\_SIZE)

.addComponent(textAdmissionTime, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(btnOK)

.addComponent(btnDelete))

.addGap(40)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(btnCancel)

.addComponent(btnModify)))))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(21)

.addComponent(lblNewLabel\_7)

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)))

.addGap(191))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(28)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(textName, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(textAge, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_1)

.addComponent(lblNewLabel\_2))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_3)

.addComponent(textSex, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel\_4)

.addComponent(textDisease, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(24)

.addComponent(lblNewLabel\_5))

.addGroup(gl\_contentPane.createSequentialGroup()

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(textSymptom, GroupLayout.PREFERRED\_SIZE, 55, GroupLayout.PREFERRED\_SIZE)))

.addGap(25)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_6)

.addComponent(textAdmissionTime, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_7))

.addGap(24)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnOK)

.addComponent(btnModify))

.addPreferredGap(ComponentPlacement.RELATED, 17, Short.MAX\_VALUE)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnDelete)

.addComponent(btnCancel))

.addContainerGap())

);

contentPane.setLayout(gl\_contentPane);

}

protected void delete(ActionEvent e) {

// TODO Auto-generated method stub

String id=textID.getText();

Delete(id);

}

private void Delete(String id) {

// TODO Auto-generated method stub

PatientDAO dao=new PatientDAO();

if(dao.deletePatientbyID(id)) {

JOptionPane.showMessageDialog(null, "删除该患者成功！","提示信息",1);

return ;

}

}

protected void modify(ActionEvent e) {

// TODO Auto-generated method stub

String id=textID.getText();

Modify(id);

}

private void Modify(String id) {

// TODO Auto-generated method stub

PatientService op=new PatientService();

if(op.login(id)==null) {

JOptionPane.showMessageDialog(null,"系统中无该患者信息","提示",1);

return;

}

else {

PatientDAO dao=new PatientDAO();

Patient patient=op.login(id);

if(dao.updatePatient(patient)) {

JOptionPane.showMessageDialog(null,"修改患者信息成功！","提示",1);

return;

}

}

}

protected void cancel(ActionEvent e) {

// TODO Auto-generated method stub

query.setVisible(true);

dispose();

}

protected void add(ActionEvent e) {

// TODO Auto-generated method stub

String id=textID.getText();

String name=textName.getText();

String age=textAge.getText();

String sex=textSex.getText();

String disease=textDisease.getText();

String symptom=textSymptom.getText();

String admission\_time=textAdmissionTime.getText();

Add(id,name,age,sex,disease,symptom,admission\_time);

}

private void Add(String id, String name, String age, String sex, String disease, String symptom,

String admission\_time) {

// TODO Auto-generated method stub

if(checkEmpty(id,name,age,sex,disease,symptom,admission\_time)) {

JOptionPane.showMessageDialog(null,"新增不成功！","提示",1);

return;

}

PatientService op=new PatientService();

if(op.register(id, name, age, sex, disease, symptom, admission\_time)) {

query.setVisible(true);

dispose();

}

}

private boolean checkEmpty(String id, String name, String age, String sex, String disease, String symptom,

String admission\_time) {

// TODO Auto-generated method stub

boolean flag=false;

if(id.isEmpty()) flag=true;

if(name.isEmpty()) flag=true;

if(age.isEmpty()) flag=true;

if(sex.isEmpty()) flag=true;

if(disease.isEmpty()) flag=true;

if(symptom.isEmpty()) flag=true;

if(admission\_time.isEmpty()) flag=true;

if(flag) {

JOptionPane.showMessageDialog(null,"有空字段没有填写","提示",1);

}

return flag;

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import BLL.MedicineService;

import Entity.Medicine;

import dao.MedicalDao;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import javax.swing.LayoutStyle.ComponentPlacement;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class addMedicine extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

JButton btnOK,btnCancel;

private JTextField textID;

private JTextField textName;

private JTextField textStoreNumber;

private JTextField textPrice;

private JTextField textProducedDate;

private JTextField textExpirationDate;

private JTextField textAdress;

private JTextField textDesc;

private JTextField textType;

private static MedicineQuery query=null;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

addMedicine frame = new addMedicine(query);

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public addMedicine(MedicineQuery query) {

this.query=query;

setTitle("医疗信息系统--药品");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 542, 521);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("药品编号");

JLabel lblNewLabel\_1 = new JLabel("药品名");

textID = new JTextField();

textID.setColumns(10);

textName = new JTextField();

textName.setColumns(10);

JLabel lblNewLabel\_2 = new JLabel("存库量");

textStoreNumber = new JTextField();

textStoreNumber.setColumns(10);

JLabel lblNewLabel\_3 = new JLabel("价格");

textPrice = new JTextField();

textPrice.setColumns(10);

JLabel lblNewLabel\_4 = new JLabel("生产日期");

textProducedDate = new JTextField();

textProducedDate.setColumns(10);

JLabel lblNewLabel\_5 = new JLabel("保质期");

textExpirationDate = new JTextField();

textExpirationDate.setColumns(10);

JLabel lblNewLabel\_6 = new JLabel("");

JLabel lblNewLabel\_7 = new JLabel("生产地址");

textAdress = new JTextField();

textAdress.setColumns(10);

JLabel lblNewLabel\_8 = new JLabel("药品描述");

textDesc = new JTextField();

textDesc.setColumns(10);

JLabel lblNewLabel\_9 = new JLabel("药品类型");

textType = new JTextField();

textType.setColumns(10);

btnOK = new JButton("新增");

btnOK.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

add(e);

}

});

btnCancel= new JButton("取消");

btnCancel.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

cancel(e);

}

});

JButton btnModify = new JButton("修改");

btnModify.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

modify(e);

}

});

JButton btnDelete = new JButton("删除");

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

delete(e);

}

});

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(81)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(51)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addComponent(lblNewLabel\_6)

.addComponent(lblNewLabel\_5)

.addComponent(lblNewLabel\_4)

.addComponent(lblNewLabel\_3)

.addComponent(lblNewLabel\_2)

.addComponent(lblNewLabel\_1)

.addComponent(lblNewLabel)

.addComponent(lblNewLabel\_7)

.addComponent(lblNewLabel\_8)

.addComponent(lblNewLabel\_9)))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(81)

.addComponent(btnOK)))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textName, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textStoreNumber, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textPrice, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textProducedDate, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textExpirationDate, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textAdress, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(textDesc, GroupLayout.PREFERRED\_SIZE, 227, GroupLayout.PREFERRED\_SIZE)

.addComponent(textType, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addContainerGap(82, Short.MAX\_VALUE))

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(6)

.addComponent(btnModify)

.addPreferredGap(ComponentPlacement.RELATED, 13, Short.MAX\_VALUE)

.addComponent(btnDelete)

.addPreferredGap(ComponentPlacement.UNRELATED)

.addComponent(btnCancel)

.addGap(91))))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(41)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel)

.addComponent(textID, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_1)

.addComponent(textName, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_2)

.addComponent(textStoreNumber, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_3)

.addComponent(textPrice, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel\_4)

.addComponent(textProducedDate, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(23)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_5)

.addComponent(textExpirationDate, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel\_6)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(lblNewLabel\_7)

.addComponent(textAdress, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(ComponentPlacement.RELATED, 14, Short.MAX\_VALUE)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(textDesc, GroupLayout.PREFERRED\_SIZE, 48, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_8))

.addGap(18)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(textType, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_9))

.addGap(28)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.BASELINE)

.addComponent(btnCancel)

.addComponent(btnOK)

.addComponent(btnModify)

.addComponent(btnDelete))

.addGap(21))

);

contentPane.setLayout(gl\_contentPane);

}

protected void delete(ActionEvent e) {

// TODO Auto-generated method stub

String id=textID.getText();

Delete(id);

}

private void Delete(String id) {

// TODO Auto-generated method stub

MedicalDao dao=new MedicalDao();

if(dao.deleteMedicinebyID(id)) {

JOptionPane.showMessageDialog(null, "删除该药品成功！","提示信息",1);

return ;

}

}

protected void modify(ActionEvent e) {

// TODO Auto-generated method stub

String id=textID.getText();

Modify(id);

}

private void Modify(String id) {

// TODO Auto-generated method stub

MedicineService op=new MedicineService();

if(op.login(id)==null) {

JOptionPane.showMessageDialog(null,"系统中无该药信息","提示",1);

return;

}

else {

MedicalDao dao=new MedicalDao();

Medicine medicine=op.login(id);

if(dao.updateMedicine(medicine)) {

JOptionPane.showMessageDialog(null,"修改药物信息成功！","提示",1);

return;

}

}

}

protected void cancel(ActionEvent e) {

// TODO Auto-generated method stub

query.setVisible(true);

dispose();

}

protected void add(ActionEvent e) {

// TODO Auto-generated method stub

String id=textID.getText();

String name=textName.getText();

String store=textStoreNumber.getText();

String price=textPrice.getText();

String producedtime=textProducedDate.getText();

String expirationtime=textExpirationDate.getText();

String address=textAdress.getText();

String desc=textDesc.getText();

String type=textType.getText();

Add(id,name,store,price,producedtime,expirationtime,address,desc,type);

}

private void Add(String id, String name, String store, String price, String producedtime, String expirationtime,

String address, String desc, String type) {

// TODO Auto-generated method stub

if(checkEmpty(id,name,store,price,producedtime,expirationtime,address,desc,type)) {

JOptionPane.showMessageDialog(null,"注册不成功！","提示",1);

return;

}

MedicineService op=new MedicineService();

if(op.register(id, name, store, price, producedtime, expirationtime, address, desc, type)) {

query.setVisible(true);

dispose();

}

else {

JOptionPane.showMessageDialog(null,op.getMessage(),"提示",1);

return;

}

}

private boolean checkEmpty(String id, String name, String store, String price, String producedtime,

String expirationtime, String address, String desc, String type) {

// TODO Auto-generated method stub

boolean flag=false;

if(id.isEmpty()) flag=true;

if(name.isEmpty()) flag=true;

if(store.isEmpty()) flag=true;

if(price.isEmpty()) flag=true;

if(producedtime.isEmpty()) flag=true;

if(expirationtime.isEmpty()) flag=true;

if(address.isEmpty()) flag=true;

if(desc.isEmpty()) flag=true;

if(type.isEmpty()) flag=true;

if(flag) {

JOptionPane.showMessageDialog(null,"有空字段没有填写","提示",1);

}

return flag;

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import Entity.Medicine;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.LayoutStyle.ComponentPlacement;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class MedicineUI extends JFrame {

static Medicine medicine=null;

private static final long serialVersionUID = 1L;

private JPanel contentPane;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

MedicineUI frame = new MedicineUI(medicine);

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

public MedicineUI(Medicine medicine) {

setTitle("医疗信息系统--药品");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 430, 572);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("药品名称："+medicine.getName());

JLabel lblNewLabel\_1 = new JLabel("库存量："+medicine.getStoreNumber());

JLabel lblNewLabel\_2 = new JLabel("生产日期："+medicine.getProducedDate());

JLabel lblNewLabel\_3 = new JLabel("保质期："+medicine.getExpirationDate());

JLabel lblNewLabel\_4 = new JLabel("生产地址："+medicine.getAddress());

JLabel lblNewLabel\_5 = new JLabel("药品描述："+medicine.getMedicinesDesc());

JLabel lblNewLabel\_6 = new JLabel("药品类型："+medicine.getMedicalType());

JLabel lblNewLabel\_7 = new JLabel("价格："+medicine.getPrice());

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(121)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addComponent(lblNewLabel\_5)

.addComponent(lblNewLabel\_3)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.TRAILING)

.addComponent(lblNewLabel\_1)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel\_4)

.addComponent(lblNewLabel\_2))))

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel\_7)

.addComponent(lblNewLabel\_6)))

.addContainerGap(251, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(62)

.addComponent(lblNewLabel)

.addGap(26)

.addComponent(lblNewLabel\_1)

.addGap(18)

.addComponent(lblNewLabel\_2)

.addGap(18)

.addComponent(lblNewLabel\_3)

.addGap(18)

.addComponent(lblNewLabel\_4)

.addGap(18)

.addComponent(lblNewLabel\_5)

.addGap(18)

.addComponent(lblNewLabel\_6)

.addGap(18)

.addComponent(lblNewLabel\_7)

.addContainerGap(209, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

}

package View;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import Entity.Patient;

import javax.swing.GroupLayout;

import javax.swing.GroupLayout.Alignment;

import javax.swing.JLabel;

public class PatientUI extends JFrame {

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private static Patient patient=null;

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

PatientUI frame = new PatientUI(patient);

frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\* @param patient

\*/

public PatientUI(Patient patient) {

setTitle("医疗信息系统--患者");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 450, 300);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

JLabel lblNewLabel = new JLabel("姓名："+patient.getName());

JLabel lblNewLabel\_1 = new JLabel("年龄："+patient.getAge());

JLabel lblNewLabel\_2 = new JLabel("性别："+patient.getSex());

JLabel lblNewLabel\_3 = new JLabel("疾病："+patient.getDisease());

JLabel lblNewLabel\_4 = new JLabel("患病症状："+patient.getSymptom());

JLabel lblNewLabel\_5 = new JLabel("入院时间："+patient.getAdmissionTime());

GroupLayout gl\_contentPane = new GroupLayout(contentPane);

gl\_contentPane.setHorizontalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(72)

.addGroup(gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addComponent(lblNewLabel\_5)

.addComponent(lblNewLabel\_4)

.addComponent(lblNewLabel, GroupLayout.PREFERRED\_SIZE, 66, GroupLayout.PREFERRED\_SIZE)

.addComponent(lblNewLabel\_1)

.addComponent(lblNewLabel\_2)

.addComponent(lblNewLabel\_3))

.addContainerGap(330, Short.MAX\_VALUE))

);

gl\_contentPane.setVerticalGroup(

gl\_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl\_contentPane.createSequentialGroup()

.addGap(26)

.addComponent(lblNewLabel)

.addGap(18)

.addComponent(lblNewLabel\_1)

.addGap(18)

.addComponent(lblNewLabel\_2)

.addGap(18)

.addComponent(lblNewLabel\_3)

.addGap(18)

.addComponent(lblNewLabel\_4)

.addGap(18)

.addComponent(lblNewLabel\_5)

.addContainerGap(47, Short.MAX\_VALUE))

);

contentPane.setLayout(gl\_contentPane);

}

public Patient getPatient() {

return patient;

}

public void setPatient(Patient patient) {

this.patient=patient;

}

}