**江南大学**

**课程设计报告**

**计科1603凌晓**

# 1开发环境与开发工具

学校医务室已有局域网，网络中有若干配置较高的台式机可以用作服务器，服务器上安装SQL server2014，其中有一台服务器能以ADSL方式带宽上网并装有IIS Web服务器。服务器或各医务室部门内部的客户机都安装了Windows操作系统，一般还安装了如Word、Excle等Office软件。

为此，开发设计的库存管理子系统，首先是基于局域网的客户机/服务器系统（C/S模式），支持企业信息集中存放在SQL Server数据库中，承担数据服务器功能，使用系统的客户机上安装有将开发设计出的库存管理子系统，多用户机同时共享使用服务器中的库存系统数据。

本子系统使用Java、SQL server2014、C/S等开发工具开发。系统能在企业内部局域网上共享使用。

# 2系统需求分析

经过调查，对医务室门诊信息管理和药房管理的业务流程进行分析，可知门诊信息管理主要是通过病人挂号、诊断、划价、收费等操作进行的，对药房管理的主要是通过药品入库、出库和门诊发药等操作进行的。

系统主要功能如下：

挂号：为每个病人建立一个包含病人基本信息（如病历号、姓名、性别、年龄等）的病历，其中病历号必须唯一，以供全系统共享调用，整个系统通过这个唯一病历号贯通一体，大夫和病人都可以藉此查询所有的就诊历史信息，并实现划价收费、药房取药等操作。

诊断：完成病历基本信息的填写，医生可以建立病情、查看病史，并将处方输入系统，供划价时使用。

划价：根据处方进行自动 计价，对药品及其他诊疗项目的价格进行严格控制。

收费：这套自动计价后，自动计算总金额，收银员据此收款，并有详尽的门诊收入统计资料。可按照收银员和病人姓名进行收费明细查询。

药房管理：包括供应商管理、药品入库、药品出库、药品盘点、药品报损、过期药品报警等，通过药品入库、出库和门诊发药等操作，可自动统计出药品现在的库存数量，根据药品的警戒高线和警戒底线进行判断，并能打印输出超线药品、底线药品。

在使用本系统之前，医务室通过手工维护病历、Excel表格来管理病人信息以及药品库存的数据。但是在使用中遇到很多问题。如1)文件级共享、共享性差,安全性低；2)实时性差，Excel表格只有及时保存后，其他电脑才能读到。3)查询、统计等操作很不方便，纸质病历的可读性差，病人难以看懂。

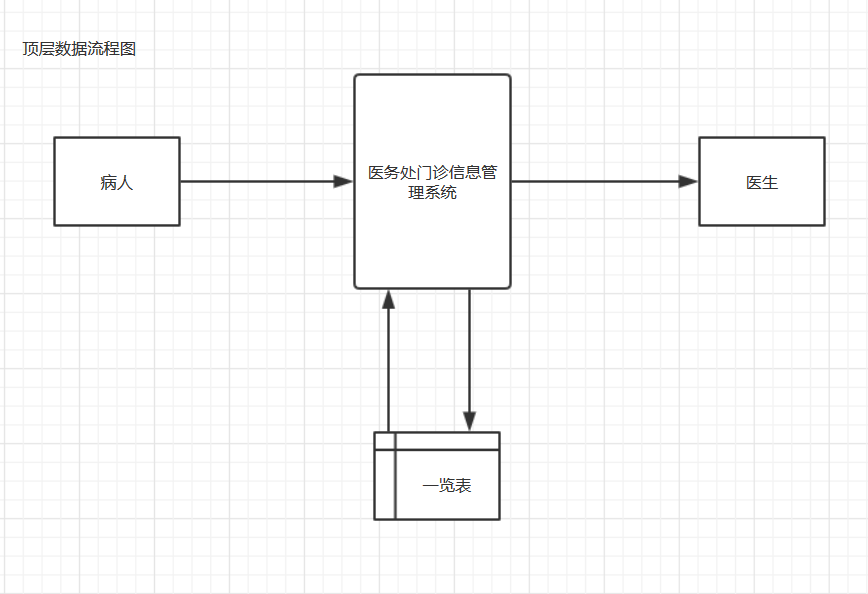
在充分了解到纸质病历，Excel等传统方式的工作模式、多次深入询问调研后，基本了解了学校医务室门诊信息管理对数据与处理的需求。

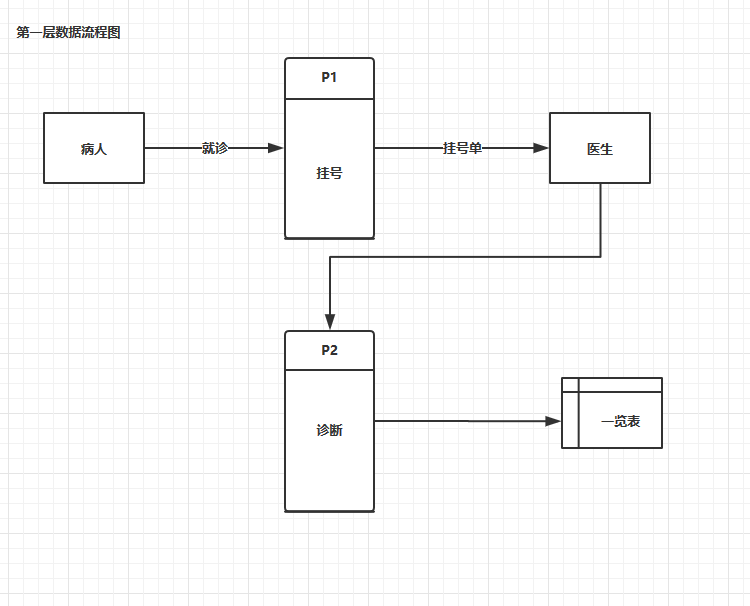
本系统主要处理的数据有，病人的挂号信息（如病历号、姓名、性别、年龄，病史等），医生对其的诊断信息（如病情，处方等等），病人看病抓药的收费信息，以及药房管理信息（包括供应商管理、药品入库、药品出库、药品盘点、药品报损、过期药品报警等)

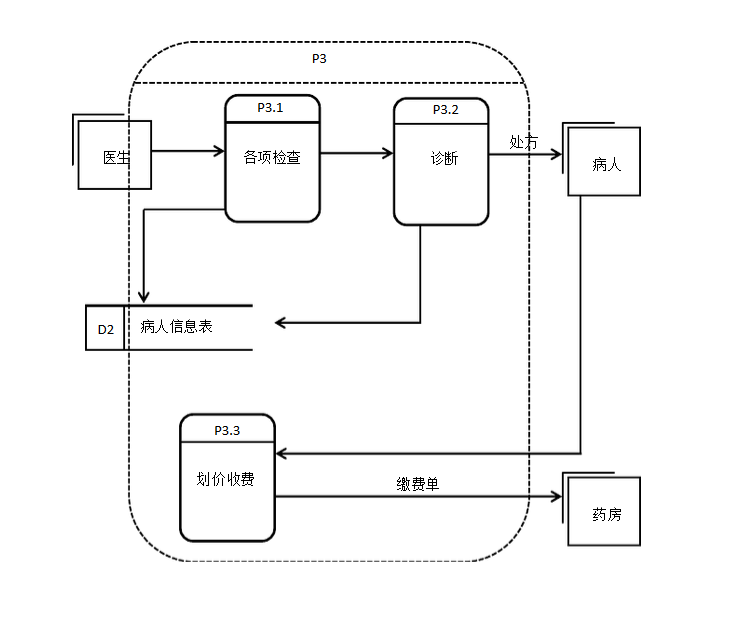
总而言之，输入病人病历信息以及药品入库信息后，可以得到病人收费信息，药品库存信息等。

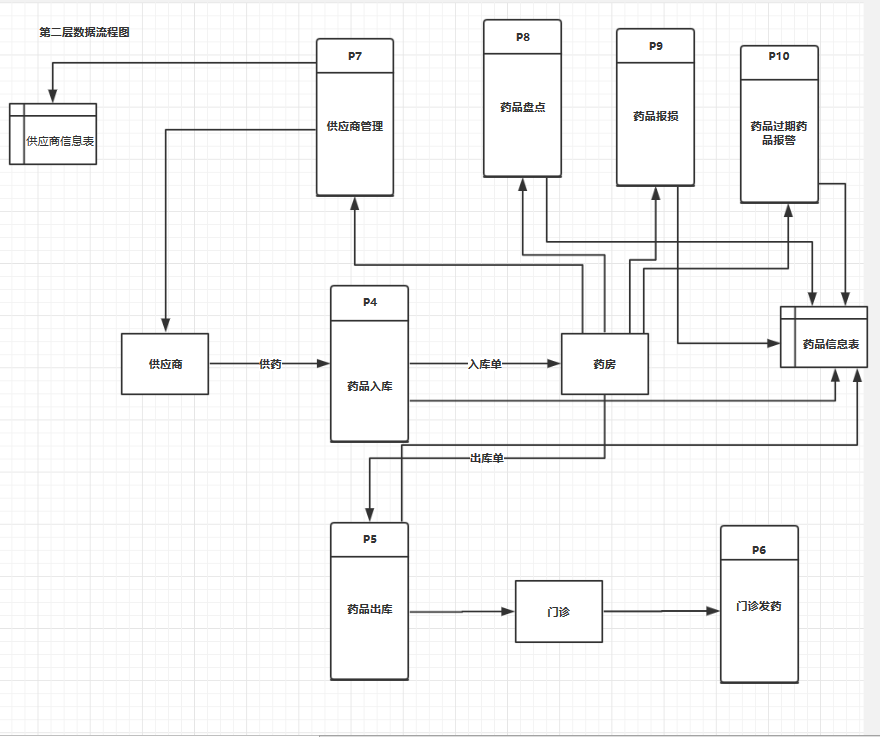
2.1系统数据流图

整个管理系统主要涉及四个事件：病人挂号，医生诊断，病人缴费，药房管理。









2.2系统数据字典

数据流图表达了数据和处理的关系，数据字典则是系统中各类数据描述的集合，是进行详细的数据收集和数据分析所获得的主要成果。数据字典通常包括数据项、数据结构、数据流、数据存储和处理过程5个部分。下面以数据字典卡片的形式来举例说明。

以“药品入库单”为例

（1）“药品入库单”数据结构

名字：药品入库单

描述：从供应商所得到的药品，以入库单的形式来记录，并登记入库

定义：药品入库单=入库单号+药品编号+药品数量+入库时间+经办人+供应商编号

位置：保存到入库表或打印保存

（2）“药品入库单”数据结构之数据项

以“入库单号”为例

名字：入库单号

描述：唯一标识某类药品入库的数字编号

定义：整型数

位置：药品入库表

（3）数据流

数据流是数据结构在系统内传输的路径。

数据流名：入库单数据流

数据流来源：管理员接收事务

数据流去向：库存处理事务

平均流量：每天几十次

高峰期流量：每天上百次

（4）数据转储

数据转储是数据结构停留或保存的地方，也是数据流的来源和去向之一。它可以是手工文档或手工凭单，也可以是计算机文档。对数据存储的描述通常包括以下内容(以入库表数据存储说明)

数据存储名：入库表

编号：入库单号为唯一标识

输入的数据流：来自供应商

输出的数据流：用于销售部门销售

数据结构：药品入库单

（5）处理过程

处理过程的具体逻辑一般用判定表或判定树来描述。数据字典中只需要描述处理过程的说明性信息。

处理过程名：实时产品库存计算

输入：入库单数据流；出库单数据流，来自销售部门销售

输出：计算出各产品当前库存

处理：产品库存计算的功能就是实时计算产品库存

2.3本系统需要管理的实体信息及实体联系信息

实体信息：  
病人：病历号、姓名、性别、年龄、联系方式、账户余额  
医生：医生编号、科室名称、姓名、联系方式、职称  
药品供应商：公司编号，公司名称，联系电话，所在地址；  
药品：药品编号、药品名称、药品单价、药品有效期、药品规格，库存数量；  
  
实体联系信息：  
  
挂号：单号、病历号、医生编号、挂号费、挂号时间  
诊断：病历号、医生编号、处方编号、诊断日期、病情、诊治费用、处方编号  
采购：药品编号、供应商编号、药品数量、采购时间、采购价格、采购员  
处方：处方编号、药品编号、药品数量

# 3功能需求分析

在数据库服务器中如SQL server2014中，要创建KCGL数据库，在数据库上建立个关系模式对应的库表信息，并确定主键、索引、参照完整性、用户自定义完整性等约束性条件。

①能对原始数据表实现输入、修改、删除、添加、查询、打印等基本操作

②能方便病人和医生登记信息、查询信息、方便药房管理员及时录入药品出入库数据

③能一键查询病人信息

④能一键查询药品

⑤分级别用户管理

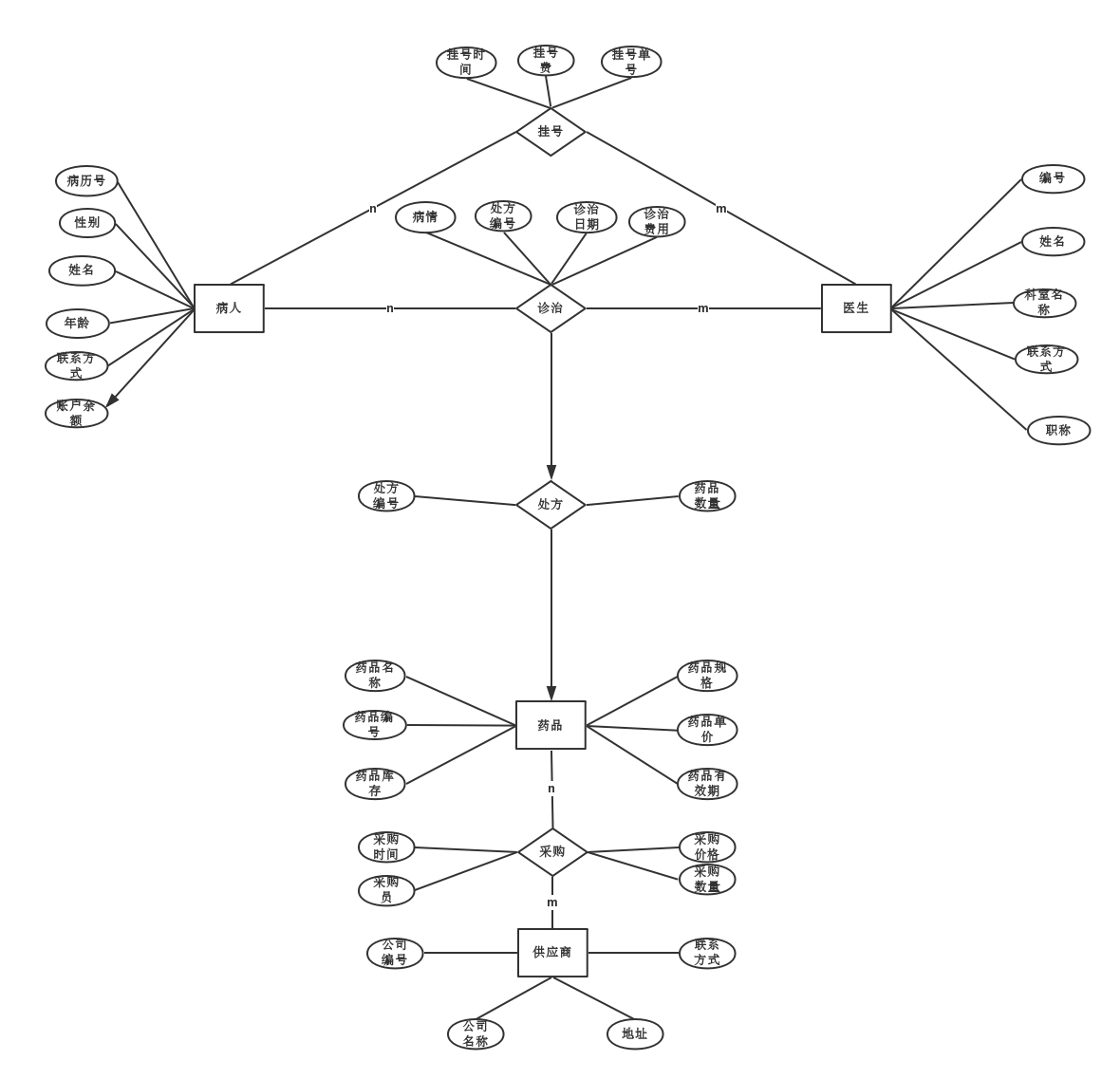
⑥高级管理员的管理操作如，系统数据库的备份与恢复、系统用户的维护、动态SQL命令操作、系统日志查阅等

⑧用户管理功能，包括用户登录、注册新用户、更改用户密码等功能

# 4系统设计

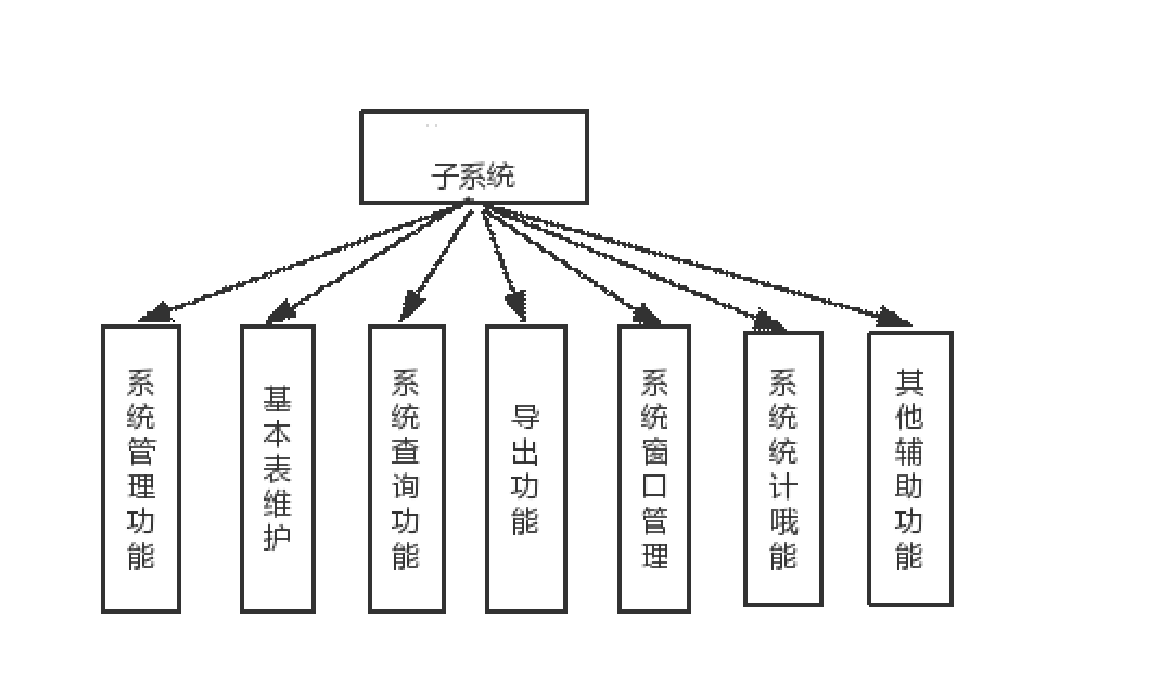
4.1数据库概念结构设计

在充分分析的基础上，经过逐步抽象、概括、分析研讨，可画出如下反映学校医务室门诊信息管理系统的整体E-R图。



4.2系统功能模块设计

对门诊信息管理子系统各项功能进行集中、分类，按照结构化程序设计的要求，可得出系统的功能模块图，如图所示



#### 4.3、数据库逻辑结构（关系模式）设计

按照E-R图到逻辑关系模式的转换规则，可得到系统如下6个关系。

1）病人信息（病历号、姓名、性别、年龄、联系方式,账户余额 ）  
2）医生信息（医生编号、姓名、科室名称、联系方式、职称）  
3）药品供应商信息（公司编号，公司名称，联系电话，所在地址）  
4）药品信息（药品编号、药品名称、药品单价、药品有效期、药品规格，库存数量）  
5）挂号信息（病历号、单号、医生编号、挂号费、挂号时间）  
6）诊治信息（病历号、医生编号、处方编号、诊断日期、病情、诊治费用）  
7）采购信息（药品编号、供应商编号、商品数量、采购时间、采购价格、采购员）  
8）处方信息（处方编号、药品编号、药品数量）

9）管理员信息（编号，姓名，密码，权限）

其中带下划线的为关系关键字（即主码）。

#### 4.4、数据库物理结构设计

**1.**本系统数据库表的物理设计通过创建表的SQL命令及数据库关系图来呈现，下面只列出Transact SQL创建命令（即T-SQL命令），针对其它数据库系统的创建命令略。

**（1）创建数据库表的T-SQL命令**

CREATE DATABASE Shoms -- 创建数据库

GO

-- 以下为创建各表的SQL命令

USE Shoms

--病人

CREATE TABLE [dbo].[Patient](

[P\_ID] [int] IDENTITY(1,1) ,

[P\_Name] [varchar](20) ,

[P\_Sex] [varchar](10) check (P\_Sex = '男'or P\_Sex = '女') ,

[P\_Age][int] ,

[P\_PN] [char](11),

[P\_Balance][money] NOT NULL,

CONSTRAINT [PK\_Patient] PRIMARY KEY CLUSTERED ([P\_ID] ASC ));

--医生

CREATE TABLE [dbo].[Doctor](

[D\_ID] [int] IDENTITY(1,1) ,

[D\_Name] [varchar](20) ,

[D\_Department] [varchar](10) ,

[D\_PN] [char](11) ,

[D\_Title][char](10) ,

CONSTRAINT [PK\_Doctor] PRIMARY KEY CLUSTERED ([D\_ID] ASC ));

--供应商

CREATE TABLE [dbo].[MSupplier](

[S\_ID] [int] IDENTITY(1,1) ,

[S\_Name] [varchar](20) ,

[S\_PN] [char](11) ,

[S\_Address][char](40) ,

CONSTRAINT [PK\_MSupplier] PRIMARY KEY CLUSTERED ([S\_ID] ASC ));

--药品

CREATE TABLE [dbo].[Medicine](

[M\_ID] [int] IDENTITY(1,1) ,

[M\_Name] [varchar](20) ,

[M\_Price] [money] ,

[M\_ExpiryDate][date] ,

[M\_Specification][char](20) ,

[M\_Stock][int] ,

CONSTRAINT [PK\_Medicine] PRIMARY KEY CLUSTERED ([M\_ID] ASC ));

--挂号

CREATE TABLE [dbo].[Registration](

[R\_ID] [int] IDENTITY(1,1) ,

[P\_ID] [int] FOREIGN KEY REFERENCES Patient(P\_ID),

[D\_ID] [int] FOREIGN KEY REFERENCES Doctor(D\_ID),

[R\_Price] [money] ,

[R\_Time][datetime] ,

CONSTRAINT [PK\_Registration] PRIMARY KEY CLUSTERED ([R\_ID] ASC ));

--处方

CREATE TABLE [dbo].[Recipt](

[R\_ID] [int] ,

[M\_ID] [int] FOREIGN KEY REFERENCES Medicine(M\_ID),

[R\_Quantity][int] ,

CONSTRAINT [PK\_Recipt] PRIMARY KEY CLUSTERED ([R\_ID],[M\_ID] ASC ));

--诊治

CREATE TABLE [dbo].[Diagnosis](

[P\_ID] [int] FOREIGN KEY REFERENCES Patient(P\_ID),

[D\_ID] [int] FOREIGN KEY REFERENCES Doctor(D\_ID),

[R\_ID] [int] ,

[D\_Date][date] ,

[D\_Condition] [text],

[D\_Price][money] ,

CONSTRAINT [PK\_Diagnosis] PRIMARY KEY CLUSTERED ([P\_ID],[D\_ID],[D\_Date] ASC ));

--采购

CREATE TABLE [dbo].[Purchase](

[M\_ID] [int] FOREIGN KEY REFERENCES Medicine(M\_ID),

[S\_ID] [int] FOREIGN KEY REFERENCES MSupplier(S\_ID),

[P\_Date][date] ,

[P\_Qulity] [int] ,

[P\_Price][money] ,

[P\_Buyer][char](20) ,

CONSTRAINT [PK\_Purchase] PRIMARY KEY CLUSTERED ([M\_ID],[S\_ID],[P\_Date] ASC ));

--管理员

CREATE TABLE [dbo].[Admin](

[A\_ID] [int] IDENTITY(1,1) ,

[A\_Name] [varchar](20),

[A\_Password][varchar](20) ,

[A\_Authority][char](20),

CONSTRAINT [PK\_Admin] PRIMARY KEY CLUSTERED ([A\_ID]ASC ));

**（2）数据库关系图**

数据库关系图如图15-4所示。



图15-4 数据库关系图

**（3）基于数据库表的视图与索引**

**1）视图：**基于该数据库库表关系图15-22，可定义出各种常用的用户视图。如：

--挂号视图

Create view [dbo].[PRD] AS

Select R\_ID,P\_Name,D\_Name

From Patient,Registration,Doctor

Where Registration.P\_ID = Patient.P\_ID AND Registration.D\_ID = Doctor.D\_ID

--药价统计视图

Create view [dbo].[DRM] AS

Select P\_ID,Sum(M\_Price) As 'Total Medicine Price '

From Diagnosis,Recipt,Medicine

Where Diagnosis.R\_ID = Recipt.R\_ID AND Recipt.M\_ID = Medicine.M\_ID

Group by P\_ID

按需可以定义出不同视图，这里不再一一列出。

1. **索引：**

注意：表索引的对性能的影响及是否采用，是需要通过实际系统的运行来比较而判定的。

--建立药品价格，名字，库存分别升序的非聚集索引

CREATE NONCLUSTERED INDEX [M\_Price]

ON [dbo].[Medicine] (M\_Price ASC,M\_Name ASC,M\_Stock ASC)

--建立医生编号，姓名，科室分别升序的非聚集索引

CREATE NONCLUSTERED INDEX [Doctor\_ID\_Name\_Department]

ON [dbo].[Doctor] (D\_ID ASC,D\_Name ASC,D\_Department ASC)

1. **表数据的初始化**

**Use** shoms

**insert** **into** **admin** **values(**'admin'**,**'123456'**,**'admin'**)**

**insert** **into** **admin** **values(**'madmin'**,**'123456'**,**'madmin'**)**

**select** **\***

**From** doctor

**insert** **into** doctor **values(**'xiaozhang'**,**'牙科'**,**'12345611234'**,**'主治医生'**)**

**insert** **into** doctor **values(**'xiaoduan'**,**'眼科'**,**'12345611235'**,**'住院医生'**)**

**insert** **into** doctor **values(**'xiaozhao'**,**'足科'**,**'14527896568'**,**'主任'**)**

**insert** **into** doctor **values(**'xiaoma'**,**'骨科'**,**'12845687856'**,**'高级医师'**)**

**insert** **into** doctor **values(**'xiaohu'**,**'口腔科'**,**'17845896589'**,**'住院医师'**)**

**Select** **\***

**From** Medicine

**insert** **into** Medicine **values** **(**'a'**,**'15'**,**'20180818'**,**'5'**,**'50'**)**

**insert** **into** Medicine **values** **(**'b'**,**'20'**,**'20170818'**,**'5'**,**'50'**)**

**insert** **into** Medicine **values** **(**'c'**,**'15'**,**'20190818'**,**'5'**,**'150'**)**

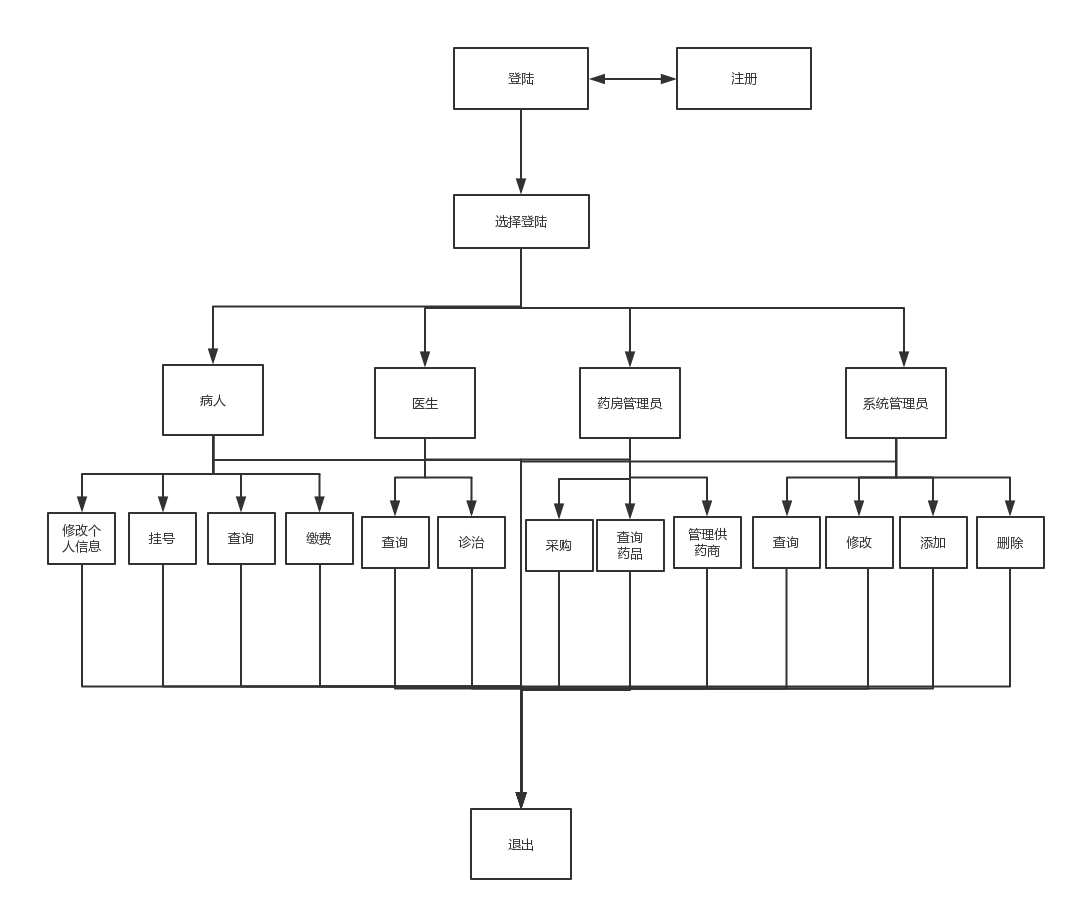
**insert** **into** Medicine **values** **(**'d'**,**'35'**,**'20180818'**,**'5'**,**'50'**)**

**insert** **into** Medicine **values** **(**'e'**,**'35'**,**'20180612'**,**'5'**,**'5'**)**

**insert** **into** Medicine **values** **(**'f'**,**'3'**,**'20180518'**,**'5'**,**'5'**)**

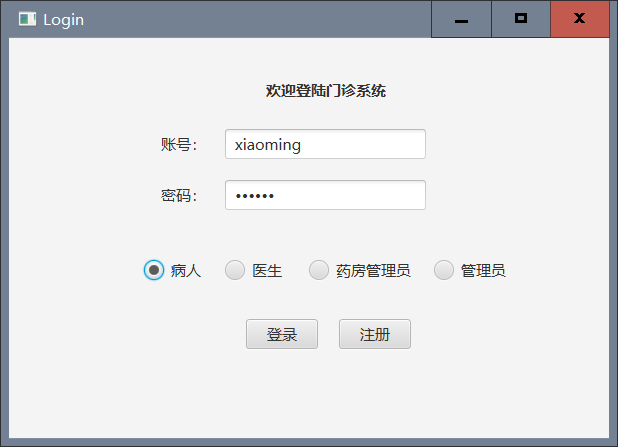
#### 系统实现

1. 程序流程图



1. 界面设计

登陆界面



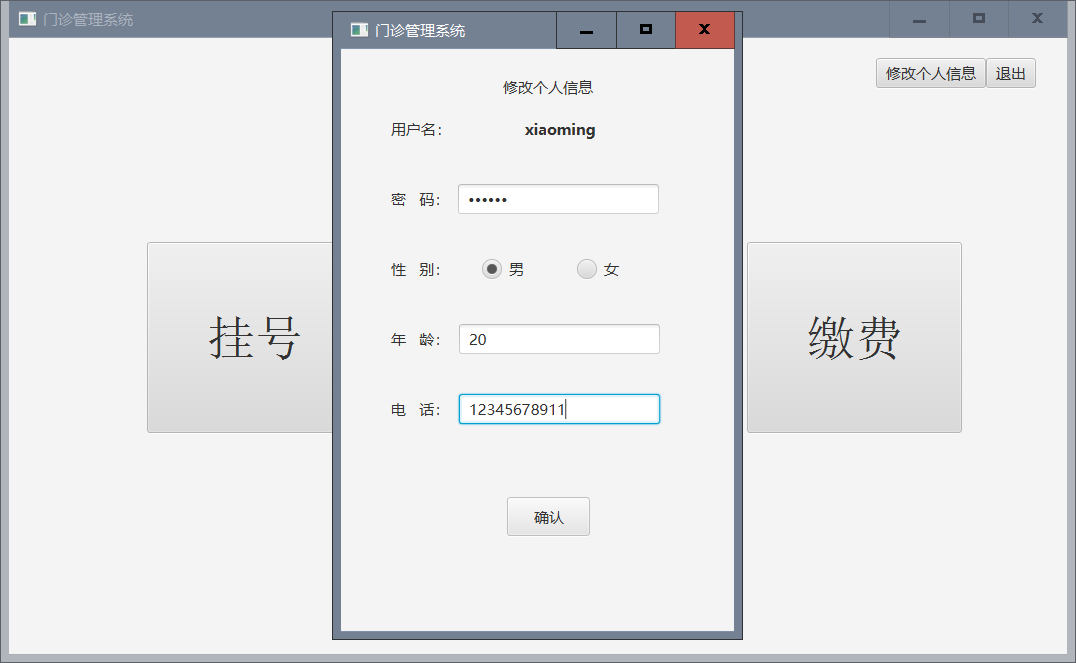
注册界面



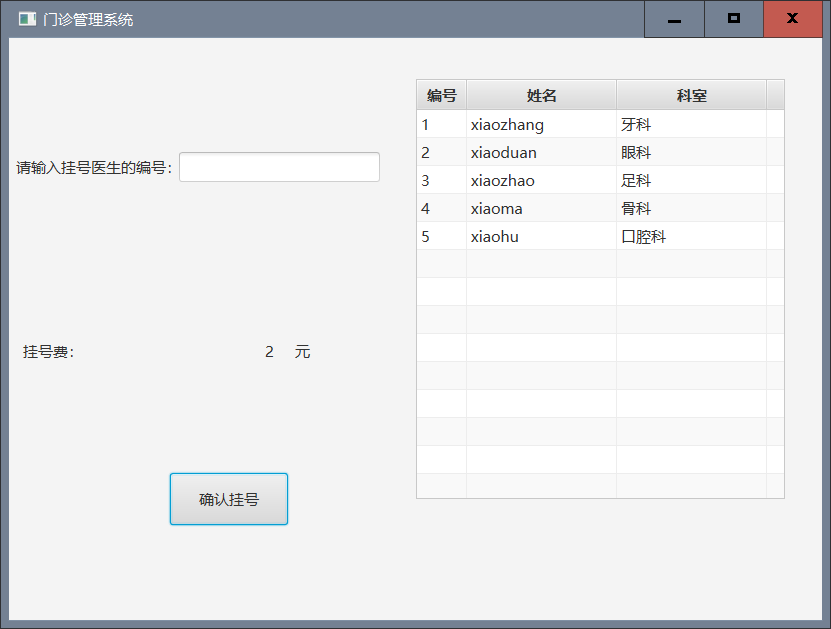
**病人端界面**



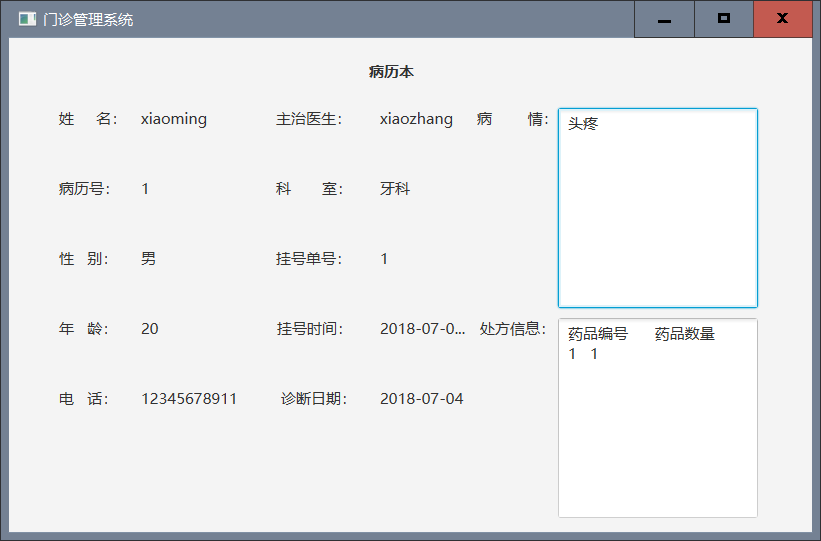
修改个人信息界面



挂号界面



查询界面



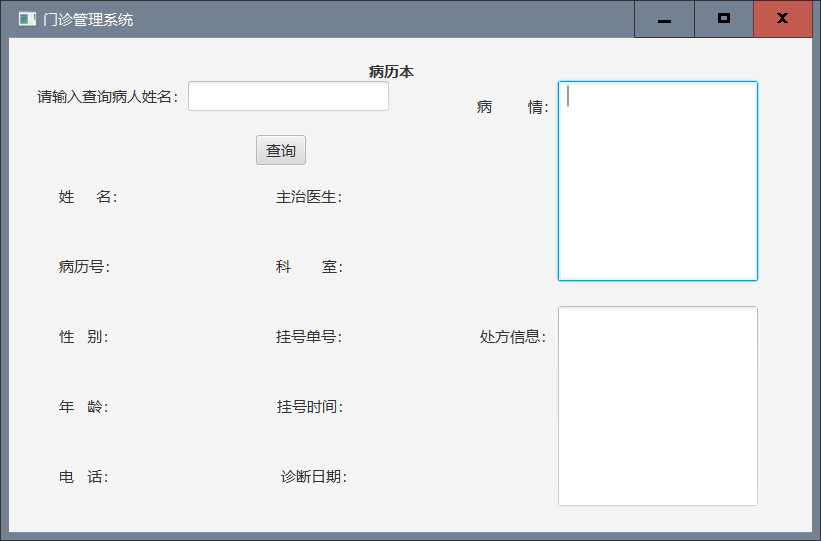
缴费界面



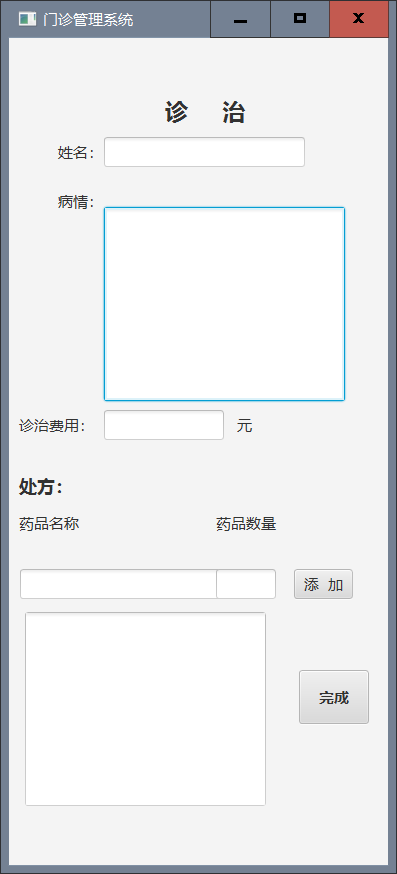
医生端界面



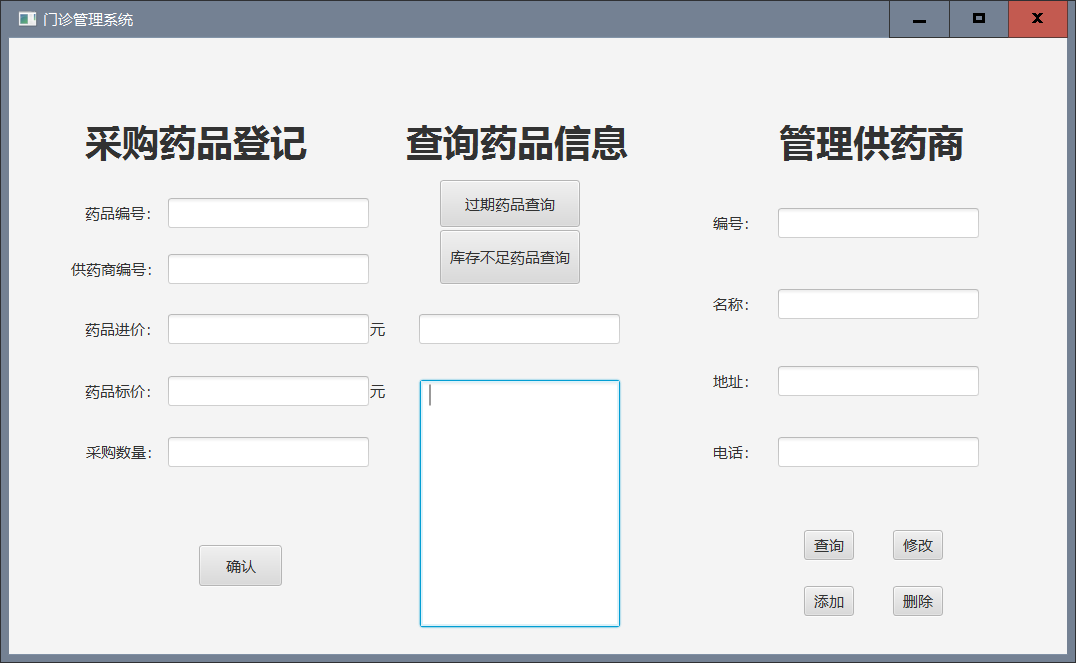
查询界面



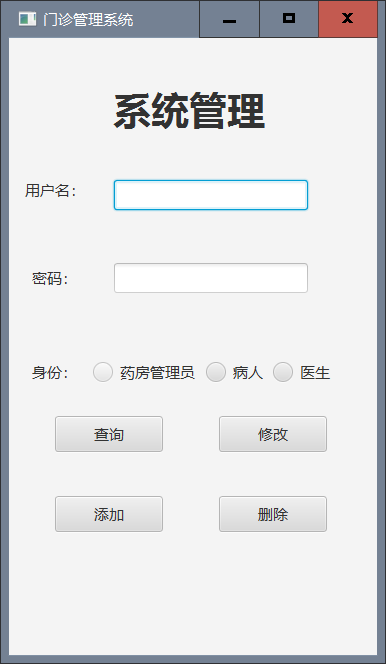
诊治界面



**药房管理员端界面**



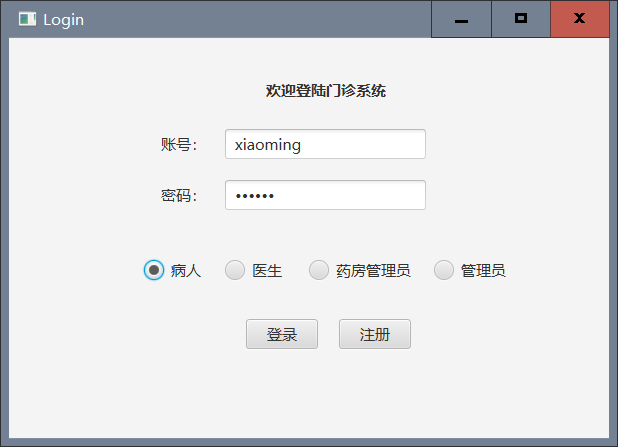
系统管理界面



1. **部分代码**

**使用JAVAFX编写**

**以登陆为例**



**登陆界面代码： Login.java**

package Login**;**

**import** javafx**.**application**.**Application**;**

**import** javafx**.**stage**.**Stage**;**

**import** javafx**.**scene**.**Scene**;**

**import** javafx**.**fxml**.**FXMLLoader**;**

**import** javafx**.**scene**.**Parent**;**

**;**

public class Login **extends** Application **{**

private static String username**;**

Stage stage**=new** Stage**();**

@Override

public void start**(**Stage primaryStage**)** **{**

**try** **{**

Parent root **=** FXMLLoader**.**load**(**getClass**()**

**.**getResource**(**"Login.fxml"**));**

primaryStage**.**setTitle**(**"Login"**);**

primaryStage**.**setScene**(new** Scene**(**root**));**

primaryStage**.**show**();**

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

e**.**printStackTrace**();**

**}**

**}**

public static void main**(**String**[]** args**)** **{**

launch**(**args**);**

**}**

public void showWindow**()** **throws** Exception **{**

start**(**stage**);**

**}**

public static void setusername**(**String s**)**

**{**

username **=** s**;**

**}**

public static String getusername**()**

**{**

**return** username**;**

**}**

**}**

**登陆界面设计代码：Login.FXML**

**<?**xml version**=**"1.0" encoding**=**"UTF-8"**?>**

**<?import** javafx**.**scene**.**text**.\*?>**

**<?import** javafx**.**scene**.**image**.\*?>**

**<?import** javafx**.**scene**.**chart**.\*?>**

**<?import** java**.**lang**.\*?>**

**<?import** javafx**.**scene**.**control**.\*?>**

**<?import** javafx**.**scene**.**layout**.\*?>**

**<**Pane fx**:**id**=**"Login" maxHeight**=**"-Infinity" maxWidth**=**"-Infinity" minHeight**=**"-Infinity" minWidth**=**"-Infinity" prefHeight**=**"400.0" prefWidth**=**"600.0" xmlns**=**"http://javafx.com/javafx/8" xmlns**:**fx**=**"http://javafx.com/fxml/1" fx**:**controller**=**"Login.LoginController"**>**

**<**children**>**

**<**Label layoutX**=**"152.0" layoutY**=**"96.0" prefHeight**=**"20.0" prefWidth**=**"53.0" text**=**"账号：" **/>**

**<**Label layoutX**=**"152.0" layoutY**=**"147.0" prefHeight**=**"20.0" prefWidth**=**"53.0" text**=**"密码：" **/>**

**<**TextField fx**:**id**=**"account" layoutX**=**"216.0" layoutY**=**"91.0" **/>**

**<**Label alignment**=**"CENTER" contentDisplay**=**"CENTER" layoutX**=**"248.0" layoutY**=**"36.0" prefHeight**=**"31.0" prefWidth**=**"137.0" text**=**"欢迎登陆门诊系统"**>**

**<**font**>**

**<**Font name**=**"System Bold" size**=**"15.0" **/>**

**</**font**></**Label**>**

**<**PasswordField fx**:**id**=**"password" layoutX**=**"216.0" layoutY**=**"142.0" **/>**

**<**RadioButton fx**:**id**=**"rbpatient" layoutX**=**"135.0" layoutY**=**"222.0" mnemonicParsing**=**"false" text**=**"病人"**>**

**<**toggleGroup**>**

**<**ToggleGroup fx**:**id**=**"group" **/>**

**</**toggleGroup**></**RadioButton**>**

**<**RadioButton fx**:**id**=**"rbdoctor" layoutX**=**"216.0" layoutY**=**"222.0" mnemonicParsing**=**"false" text**=**"医生" toggleGroup**=**"$group" **/>**

**<**RadioButton fx**:**id**=**"rbmadmin" layoutX**=**"300.0" layoutY**=**"222.0" mnemonicParsing**=**"false" prefHeight**=**"20.0" prefWidth**=**"108.0" text**=**"药房管理员" toggleGroup**=**"$group" **/>**

**<**RadioButton fx**:**id**=**"rbadmin" layoutX**=**"425.0" layoutY**=**"222.0" mnemonicParsing**=**"false" text**=**"管理员" toggleGroup**=**"$group" **/>**

**<**Button fx**:**id**=**"login" layoutX**=**"237.0" layoutY**=**"281.0" mnemonicParsing**=**"false" onAction**=**"#to\_Client" prefHeight**=**"30.0" prefWidth**=**"72.0" text**=**"登录" **/>**

**<**Button fx**:**id**=**"register" layoutX**=**"330.0" layoutY**=**"281.0" mnemonicParsing**=**"false" onAction**=**"#to\_Register" prefHeight**=**"30.0" prefWidth**=**"72.0" text**=**"注册" **/>**

**<**Accordion **/>**

**<**Label fx**:**id**=**"tips" alignment**=**"CENTER" layoutX**=**"223.0" layoutY**=**"339.0" prefHeight**=**"20.0" prefWidth**=**"201.0" textFill**=**"#f80505" **/>**

**</**children**>**

**</**Pane**>**

**登陆事件响应代码：LoginController**

package Login**;**

**import** javafx**.**fxml**.**FXML**;**

**import** Sql**.\*;**

**import** javafx**.**scene**.**control**.**Button**;**

**import** javafx**.**scene**.**control**.**Label**;**

**import** javafx**.**scene**.**control**.**PasswordField**;**

**import** javafx**.**scene**.**control**.**RadioButton**;**

**import** javafx**.**scene**.**control**.**TextField**;**

**import** javafx**.**scene**.**control**.**ToggleGroup**;**

**import** Register**.**Register**;**

**import** Client**.\*;**

public class LoginController**{**

public LoginController**()** **{**

// TODO Auto-generated constructor stub

**}**

@FXML

private Button login**,**register**;**

@FXML

private TextField account**;**

@FXML

private PasswordField password**;**

@FXML

private RadioButton rbpatient**,**rbdoctor**,**rbmadmin**,**rbadmin**;**

@FXML

private Label tips**;**

@FXML

private ToggleGroup group**;**

public void to\_Register**()** **throws** Exception **{**

Register r**=new** Register**();**

r**.**showWindow**();**

register**.**getScene**().**getWindow**().**hide**();**

**}**

@FXML

public void to\_Client**()** **throws** Exception**{**

**if(**account**.**getText**().**isEmpty**())**

**{**

tips**.**setText**(**"账号不能为空"**);**

**return** **;**

**}**

**else** **if(**password**.**getText**().**isEmpty**())**

**{**

tips**.**setText**(**"密码不能为空"**);**

**return** **;**

**}**

**else** **if(**group**.**getSelectedToggle**()** **==** rbpatient**)**

**{**

**if(!**Login\_sql**.**Login**(**account**.**getText**(),** password**.**getText**(),** "patient"**))**

**{**

tips**.**setText**(**"账号不存在或者密码错误"**);**

**return** **;**

**}**

Login**.**setusername**(**account**.**getText**());**

Patient p **=** **new** Patient**();**

p**.**showWindow**();**

register**.**getScene**().**getWindow**().**hide**();**

**}**

**else** **if(**group**.**getSelectedToggle**()** **==** rbdoctor**)**

**{**

**if(!**Login\_sql**.**Login**(**account**.**getText**(),** password**.**getText**(),** "doctor"**))**

**{**

tips**.**setText**(**"账号不存在或者密码错误"**);**

**return** **;**

**}**

Login**.**setusername**(**account**.**getText**());**

Doctor d **=** **new** Doctor**();**

d**.**showWindow**();**

register**.**getScene**().**getWindow**().**hide**();**

**}**

**else** **if(**group**.**getSelectedToggle**()** **==** rbmadmin**)**

**{**

**if(!**Login\_sql**.**Login**(**account**.**getText**(),** password**.**getText**(),** "madmin"**))**

**{**

tips**.**setText**(**"账号不存在或者密码错误"**);**

**return** **;**

**}**

Login**.**setusername**(**account**.**getText**());**

Madmin m **=** **new** Madmin**();**

m**.**showWindow**();**

register**.**getScene**().**getWindow**().**hide**();**

**return** **;**

**}**

**else** **if(**group**.**getSelectedToggle**()** **==** rbadmin**)**

**{**

**if(!**Login\_sql**.**Login**(**account**.**getText**(),** password**.**getText**(),** "admin"**))**

**{**

tips**.**setText**(**"账号不存在或者密码错误"**);**

**return** **;**

**}**

Login**.**setusername**(**account**.**getText**());**

Admin a **=** **new** Admin**();**

a**.**showWindow**();**

register**.**getScene**().**getWindow**().**hide**();**

**return** **;**

**}**

tips**.**setText**(**"请选择身份"**);**

**}**

**}**

**登陆数据库连接查询代码**：**Login\_sql.java**

package Sql**;**

**import** java**.**sql**.**Connection**;**

**import** java**.**sql**.**ResultSet**;**

**import** java**.**sql**.**SQLException**;**

**import** java**.**sql**.**Statement**;**

public class Login\_sql **{**

public static Boolean Login**(**String username**,**String password**,**String authority**)**

**{**

Connection con**=**DBUtil**.**getConnection**();**

String sql**=**"SELECT A\_Password\r\n" **+**

"FROM Admin\r\n" **+**

"WHERE A\_Name= '"**+** username **+** "' AND A\_Authority = '" **+** authority **+** "' "**;**

Statement st**;**

String Passwords **=** **null;**

**try** **{**

st **=** con**.**createStatement**();**

ResultSet rs **=** st**.**executeQuery**(**sql**);**

**if(**rs**.**next**())** Passwords **=** rs**.**getString**(**"A\_password"**);**

**else** **return** **false;**

rs**.**close**();**

st**.**close**();**

**}**

**catch(**SQLException e**)**

**{**

e**.**printStackTrace**();**

**}**

**if(**Passwords**.**equals**(**password**))** **return** **true;**

**else** **return** **false;**

**}**

**}**

**6.系统测试：**

**当堂测试成功**

**7.体会与自我评价**

历时3周的数据库课程设计，从数据库的概念设计到物理结构设计再到编写代码构建数据库，一步一步，循序渐进，让我熟练掌握了数据库的设计与实现，学会了如何构建以及实现，简化数据库等。在编写系统的过程中，不但让我对数据库的使用方式与方法更加清晰，同时让我对数据库的认知更加深入，让我学会了如何使用数据库搭建自己设计的系统，获益良多。