

# 计算机科学系实验报告

课程名称 UML 与可视化建模 班级 14 网络 2 班  
实验名称 门诊挂号管理系统 教导教师 曾少宁  
姓名 陈燕珊 学号 1414080903238 日期 2017.4.18

---

## 一、实验目的

掌握基于 UML 2.0 的建模概念与方法，掌握各种 UML 图的概念与画法，其中包括用例图、活动图、类图、顺序图、组件图和状态图等。

## 二、实验设备与环境

操作系统：Windows 7；建模工具：StarUML。

## 四、实验要求

1. 实验及实验报告以增量方式完成，每次作业都在上一次作业的基础上完成，作业提交网站不提供报告下载，所以请同学们自行保管好自己的实验报告；
2. 请将实验报告中“占位符”信息替换为自己的实验相关信息；
3. 请认真撰写实验体会，**实验课结束时**立即上传实验报告：<http://zeng.shaoning.net/uml/>。

## 四、实验内容、程序清单及运行结果

门诊挂号管理系统

1. 预约挂号
2. 查询挂号信息
3. 退号

### 1、实验一：需求建模 - 用例模型

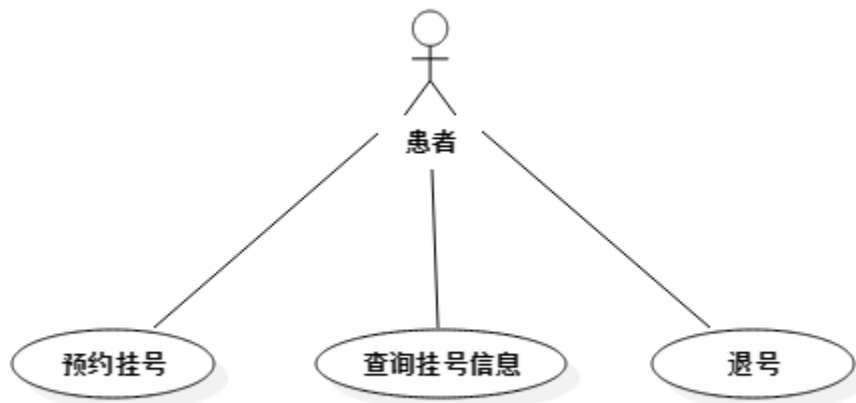


图 1：门诊挂号管理系统用例图

注：用例规约内容及项目可自行增加。

用例编号:	UC001
用例名称:	预约挂号
用例描述:	
前置条件:	患者已经登录
基本流程:	<ol style="list-style-type: none"> <li>1. 患者点击“预约挂号”按钮;</li> <li>2. 系统显示预约挂号登记界面;</li> <li>3. 患者选择挂号类别, 选择医生和预约时间段, 点击“预约”按钮;</li> <li>4. 系统检测选择医生和预约时间段无异常后, 显示确认预约登记界面;</li> <li>5. 患者检查信息无误, 点击“提交”按钮;</li> <li>6. 系统检测该患者当天预约次数无异常后, 产生该患者的预约号码, 将患者预约信息保存到数据库, 显示预约成功。</li> </ol>
扩展流程:	<ol style="list-style-type: none"> <li>4.1 系统检测到医生某时段或当天号源已满, 提示该医生预约已满;</li> <li>6.1 系统检测到该患者当天未预约次数超过 3 次, 显示预约失败</li> </ol>
后置条件:	

用例编号:	UC002
用例名称:	查询挂号信息
用例描述:	
前置条件:	患者已登录
基本流程:	<ol style="list-style-type: none"> <li>1. 患者点击“查询”按钮;</li> <li>2. 系统显示查询界面;</li> <li>3. 患者输入自己的身份证号, 点击“确定”按钮;</li> <li>4. 系统检测患者输入的身份证信息无误后, 显示患者的预约挂号信息。</li> </ol>
扩展流程:	4.1 系统检测到身份证格式有误或输入错误, 提示输入错误, 请重新输入;
后置条件:	

用例编号:	UC003
用例名称:	退号
用例描述:	
前置条件:	患者已经登录自己的账号
基本流程:	<ol style="list-style-type: none"> <li>1. 患者点击“预约挂号”按钮;</li> <li>2. 系统显示预约挂号界面;</li> <li>3. 患者点击“我的挂号”按钮;</li> <li>4. 系统显示待就诊界面;</li> <li>5. 患者点击“退号”按钮;</li> <li>6. 系统检测退号时间还未截止, 将患者挂号信息从数据库删除, 显示退号成功。</li> </ol>
扩展流程:	6.1 系统检测到退号时间已经截止, 显示退号失败
后置条件:	

## 2、实验二：过程建模 – 活动模型

使用活动图描述系统的业务过程。

方法：将用例规约中的基本流程与扩展流程抽象为过程步骤（Action），画出对应的活动图。

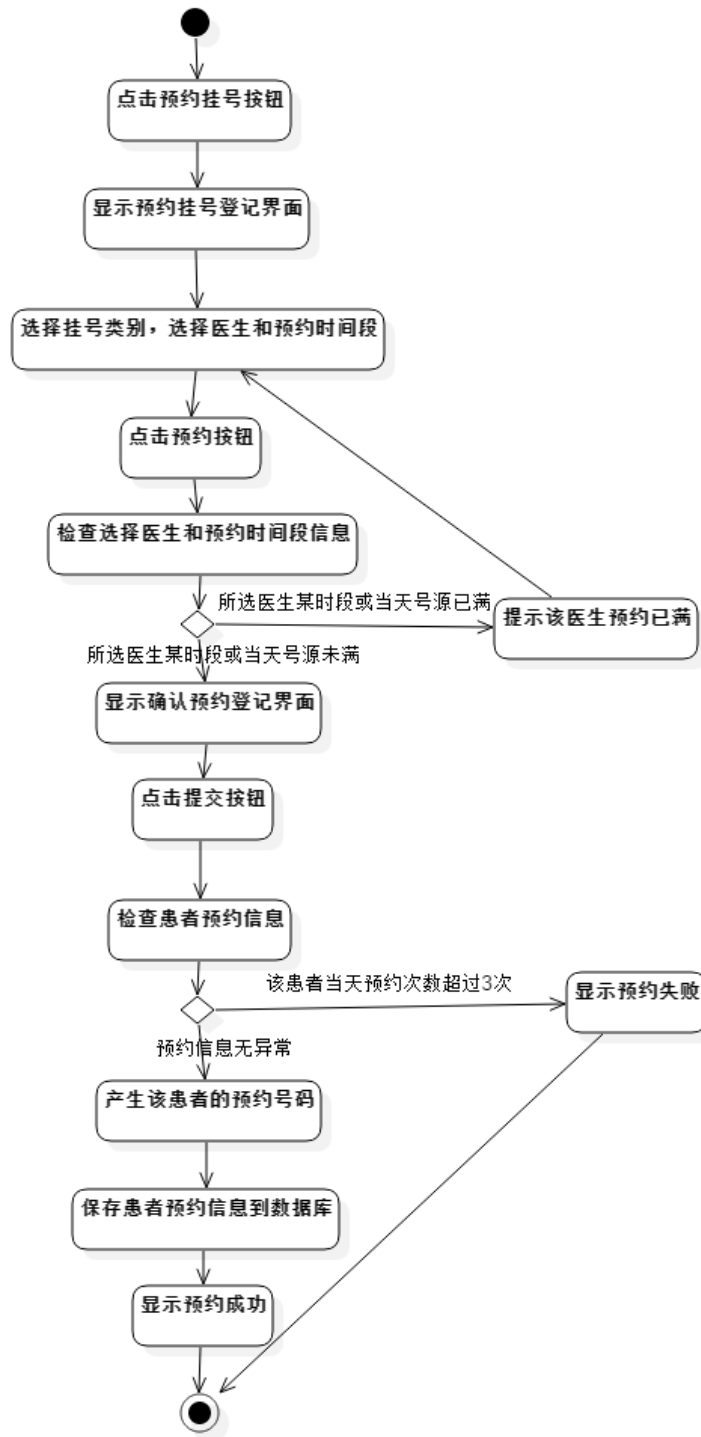


图 2：预约挂号活动图

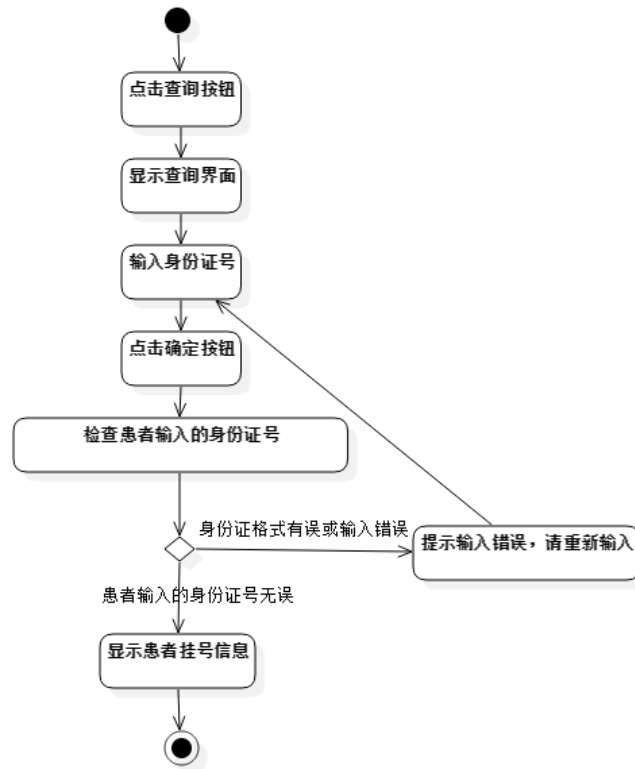


图 3: 查询挂号信息活动图

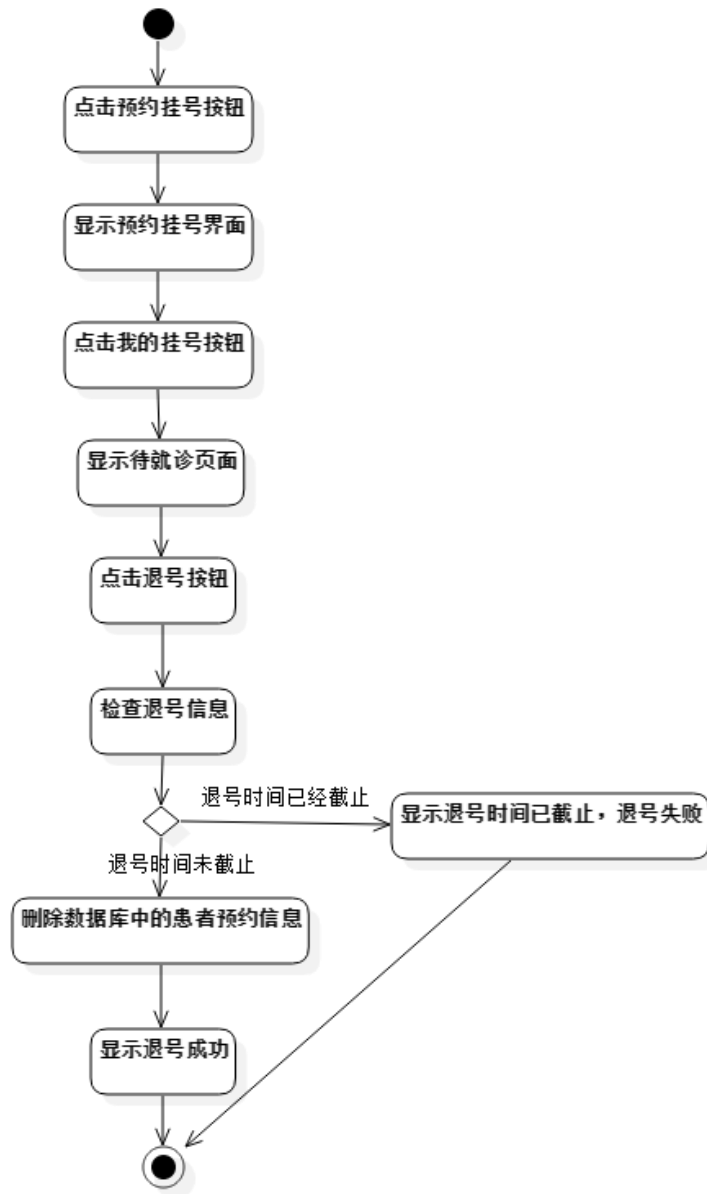


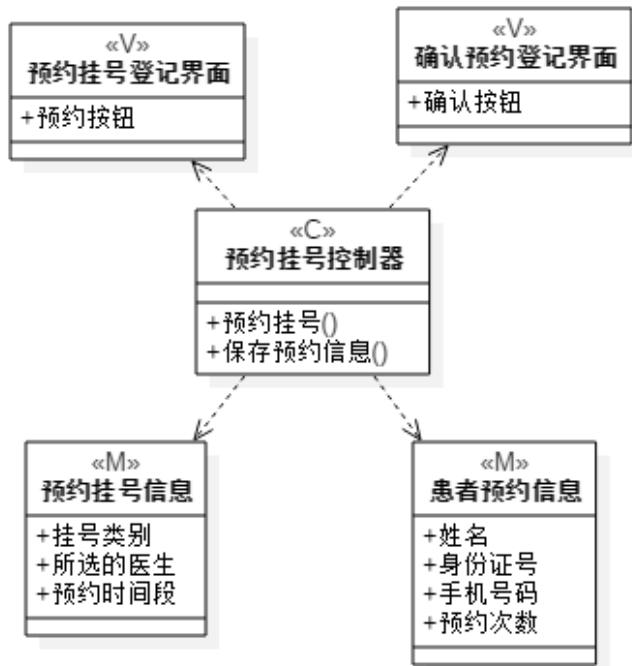
图 4：退号活动图

### 3、实验三：逻辑建模 – 类模型

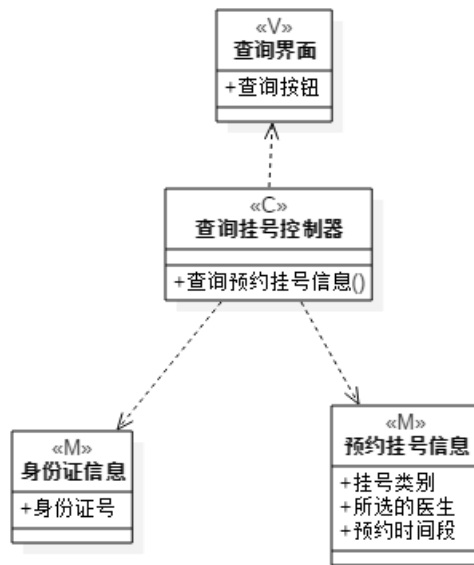
基于 MVC 设计模式找出实现用例的类。

方法：分别找出实现用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。

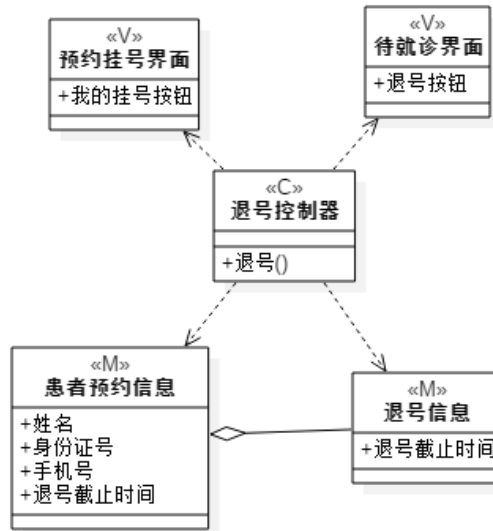
参考：讲义 P26 页。



预约挂号模型



查询挂号模型



退号模型

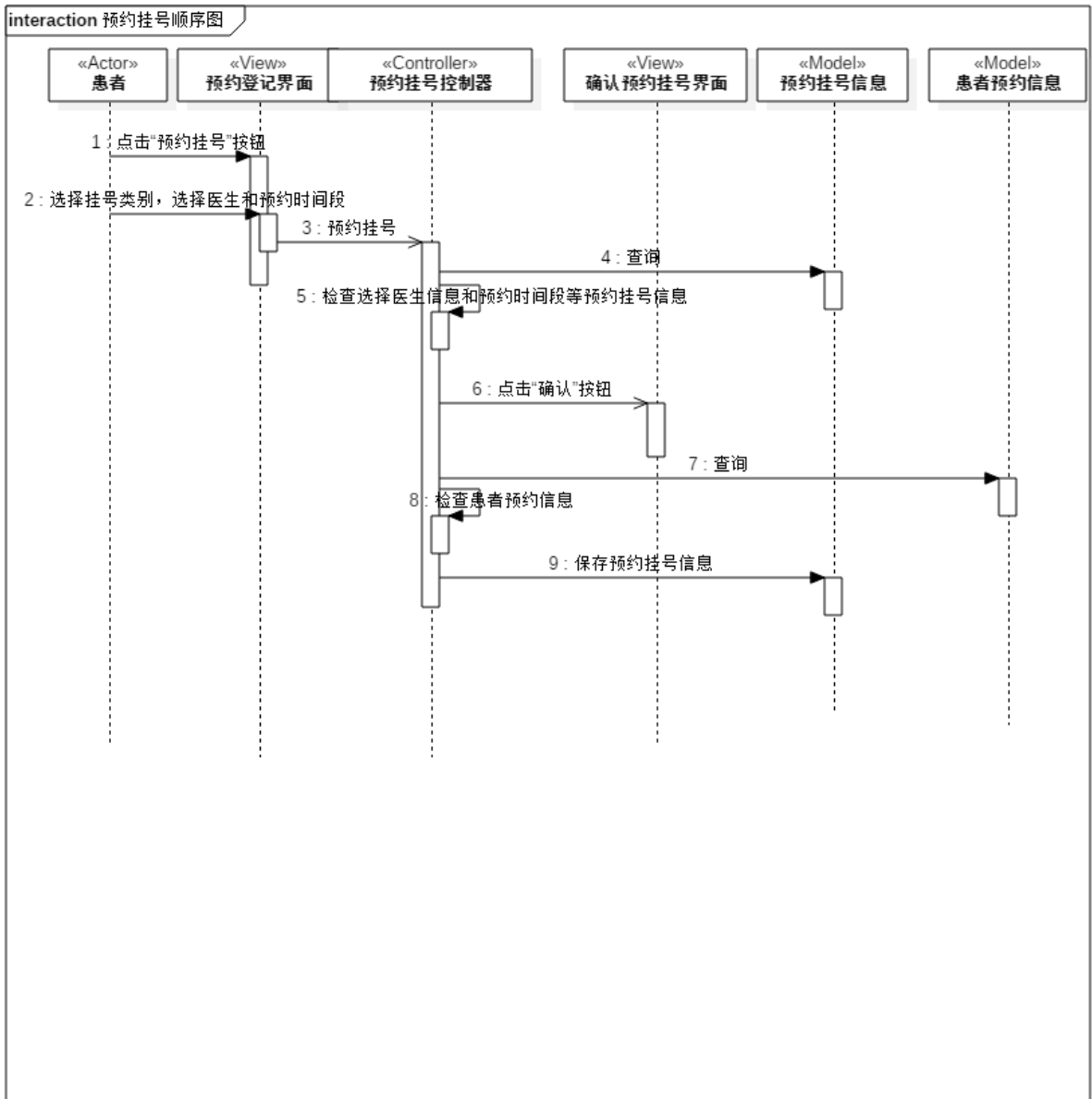
#### 4、实验四：交互建模 – 顺序模型

创建各个类（MVC 及 Actor）的对象，并描述对象之间的交互。

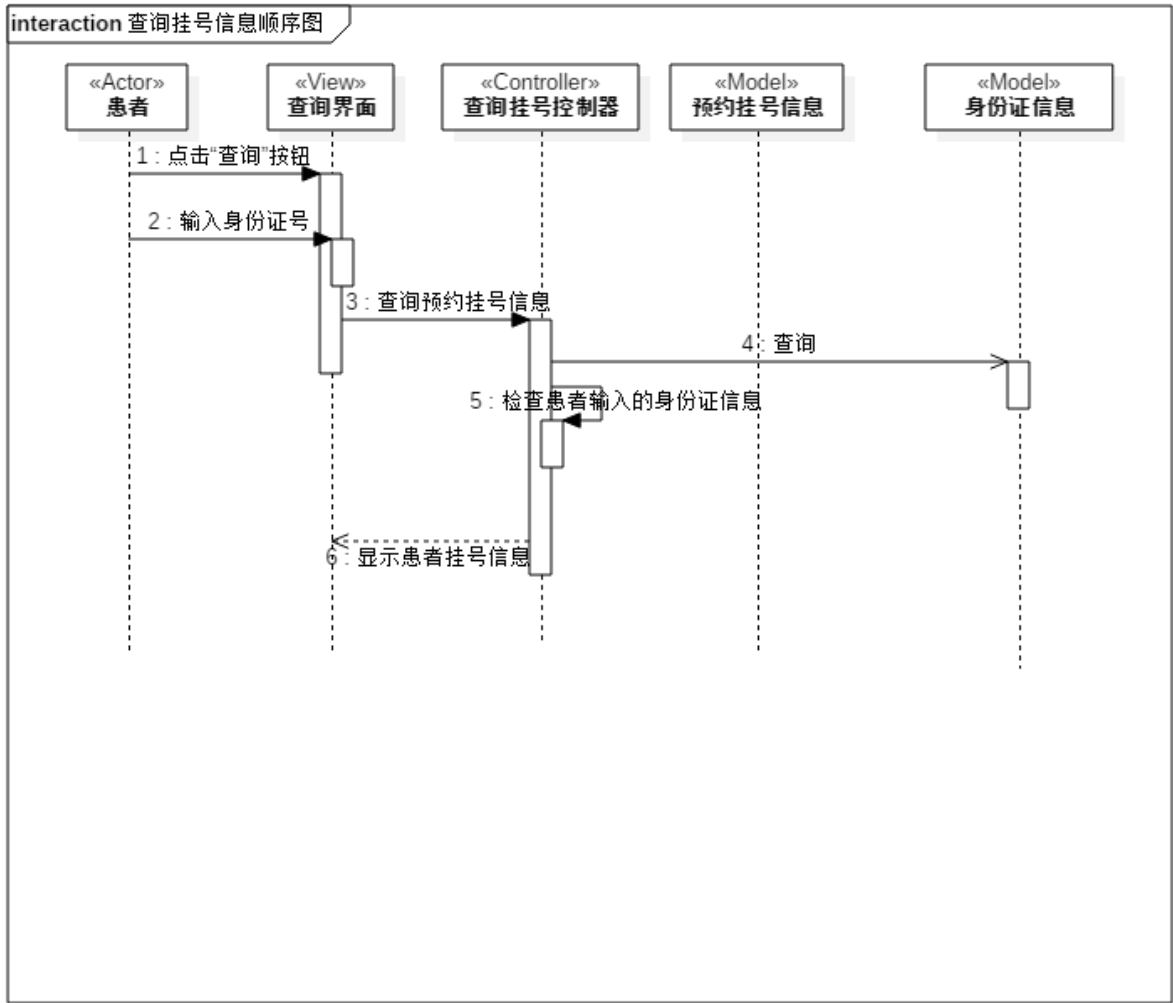
方法：分别创建参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）的对象，描述各个对象之间的消息及其顺序，画出顺序图。

参考：讲义 P33 页 8.7.2。

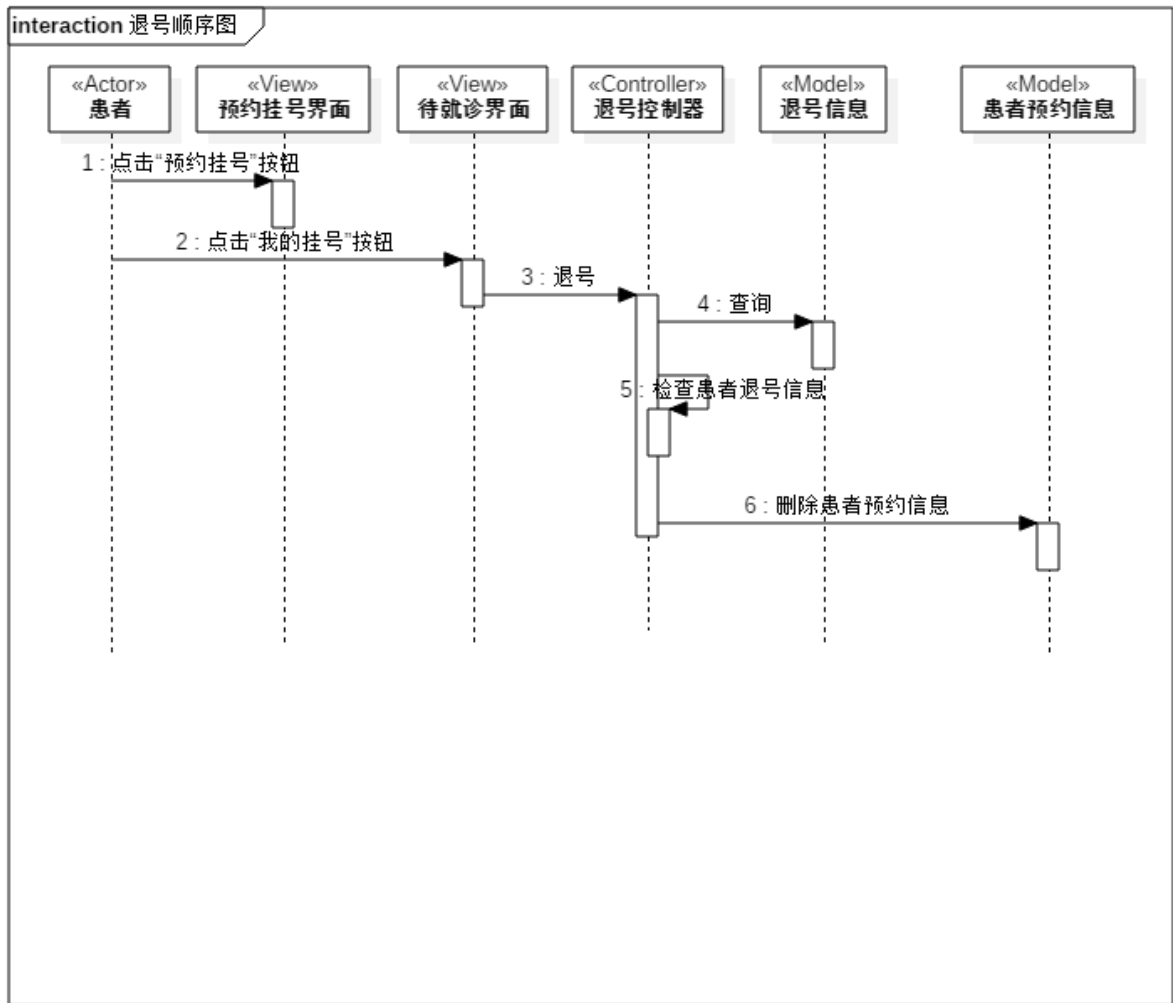




预约挂号顺序图



查询挂号信息顺序图



退号顺序图

## 5、实验五：状态建模 – 状态模型

对系统中最重要的对象进行状态建模。

方法：选择一种对象，定义该对象的状态，描述状态之间的切换及条件，画出状态图。

参考：讲义 P9 和 P10 页。

## 五、实验体会

实验一：本实验是用例模型。比较难的是要如何找出用例。根据上课老师讲的，用例是只做一件事，且有正确和错误两个结果，很快找出了 3 个用例。

实验二：本实验是活动模型。值得注意的是要保证用例规约的基本流程步骤与活动图的步骤一致；活动图无循环；流程图路径为 1+扩展流程数。按照上课老师所讲的进行，在实验一的基础上进行添加修改，最终完成了。

实验三：本实验是类模型，先分别找出实现用例的模型（Model）、视图（View）和控制器（Controller）

类；其中把找到的数据作为 Model；界面作为 View，把用例名作为控制器；在此实验中，不要因为要重用一种东西而使用继承。

实验四：本实验是顺序模型，先创建参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）的对象，再描述各个对象之间的消息及其顺序，画出顺序图；需要注意的是消息（类图中的方法）一定指向存活条，除了第一条消息，其余的消息都从存活条发出。消息在上面的先执行，不能把不同消息画在同一水平线上。按照老师所讲的进行，通过几次的修正，得出以上三个顺序图。

实验五：