**牙周帮需求分析规约**

1652805 徐伟喆

1652803 赵 伟

1652752 袁文皓

1652875 冯玉山

指导老师：杜庆峰

学 院：软件学院

**同济大学**

**Tongji University**

# 目　录

[目　录 2](#_Toc29444)

[修订历史记录 3](#_Toc24308)

[1. 引言 4](#_Toc9585)

[1.1. 背景 4](#_Toc28738)

[1.2. 参考资料 4](#_Toc15721)

[1.3. 假定和约束 4](#_Toc29653)

[1.4. 用户的特点 5](#_Toc17113)

[2. 功能需求 5](#_Toc23330)

[2.1. 系统范围 5](#_Toc11706)

[2.2. 系统体系结构 6](#_Toc8518)

[2.3. 系统总体流程 7](#_Toc2964)

[2.4. 需求分析 8](#_Toc10502)

[3. 非功能需求 47](#_Toc31242)

[3.1. 安全性 47](#_Toc4002)

[3.2. 可移植性 47](#_Toc26335)

[3.3. 易用性 47](#_Toc17225)

[3.4. 可维护性 48](#_Toc18713)

[3.5性能要求 48](#_Toc25143)

[4. 运行环境规定 49](#_Toc25632)

[4.1. 设备 49](#_Toc20424)

[4.2. 支持软件 50](#_Toc27929)

[4.3. 接口 50](#_Toc5882)

[4.4. 控制 50](#_Toc27593)

[5. 需求跟踪 50](#_Toc11835)

修订历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编写日期** | **版本** | **说明** | **作者** | **。。。。** |
| 2018.12.25 | 版本1 | 详细需求分析 | 袁文皓 |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 1. 引言

### 1.1. 背景

牙周病是一种需要患者进行长期牙周检查的口腔疾病，传统模式中，检查时需要两名医生相互配合，一人检查一人用纸笔记录，效率不高而且医生工作量大容易出错；使用纸质表格进行保存，容易出现丢失或损坏等情况，影响治疗的进行；了解牙周病的发展趋势多是通过对牙周组织的情况对比得到，传统纸质表格，对比起来相当复杂，不够直观；由于病人的个人原因，有时会需要转诊，纸质表格记录使得转诊相当麻烦，影响治疗进程。根据医生和病人的这些痛点，我们提出了牙周帮这个项目，牙周帮包括网页端和微信小程序端。通过牙周帮，医生能够通过语音输入用手机独立完成一次牙周检查，还能够查询历次检查信息进行对比，便于诊断，患者也能够及时掌握自己的治疗情况。本系统希望能够访问医院的预约系统以及药房系统以便于患者预约就诊和医生开处方。

### 1.2. 参考资料

* Software Engineering A PRACTITIONER'S APPROACH
* 需求规约文档
* 需求（分析）规约（说明书）

### 1.3. 假定和约束

* 开发期限至2019/1/12
* 经费限制1000以内
* 设备限于个人电脑及智能手机
* 用户资料准备齐全，无交流上的问题

### 1.4. 用户的特点

用户分为两类，一类是医生，他们精于专业知识，也能够熟练的使用电脑进行操作，他们回频繁使用本软件进行工作；另一类是患者，他们的电脑使用水平不定，但应都能熟练使用微信或者在电脑上浏览网页，使用频率较医生低。

## 2. 功能需求

### 2.1. 系统范围

**系统开发意图**：

**传统的模式缺点**：

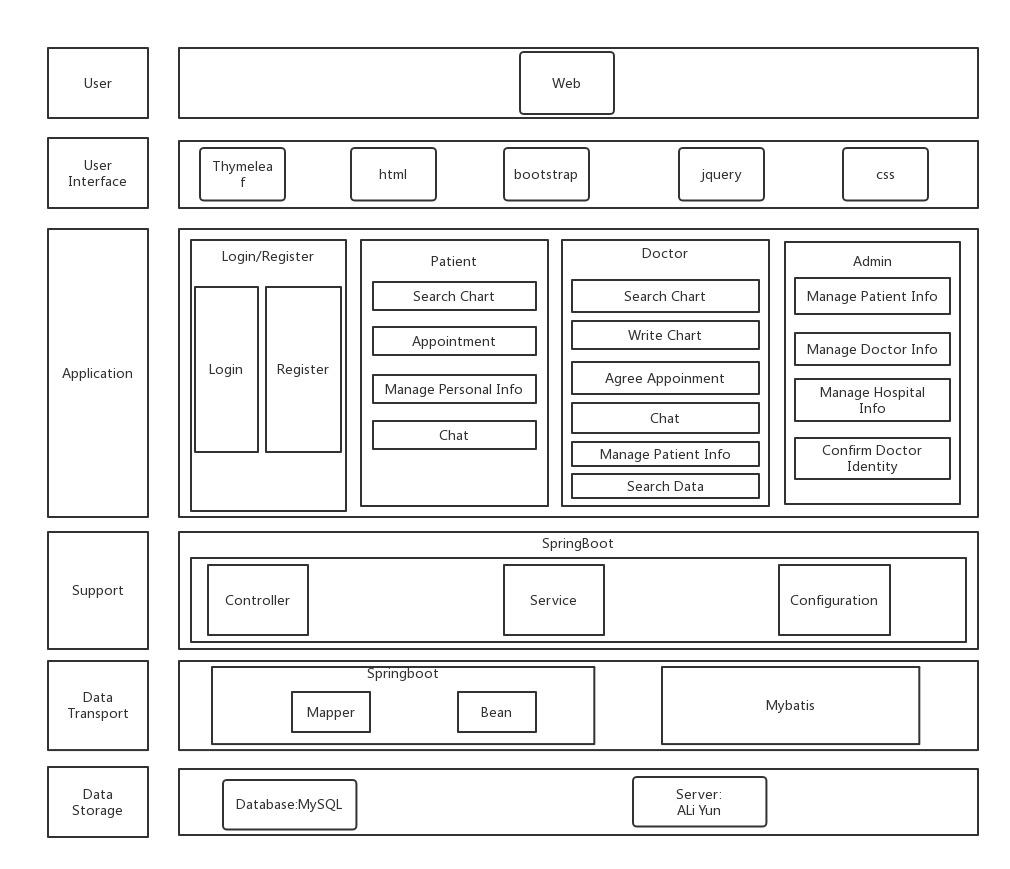
1. 需要两名牙医相互配合，一人检查一人用笔记录，造成人力资源浪费
2. 传统纸质表格数据对比起来比较复杂不够直观
3. 纸质表格保存不易
4. 患者转诊不易

因此开发这个系统的意图就在于解决这些传统模式下的不足之处，使病人与用户都得到最好的体验，提高传统模式的效率。

**应用目标**：针对具有合格资质的牙医以及他们的病人，对传统的看牙病的模式利用现有技术进行改进，将信息的保存，计算，分析交由系统，医生只需微信小程序或是网页就能完成数据的输入与查看，减轻医生的负担。

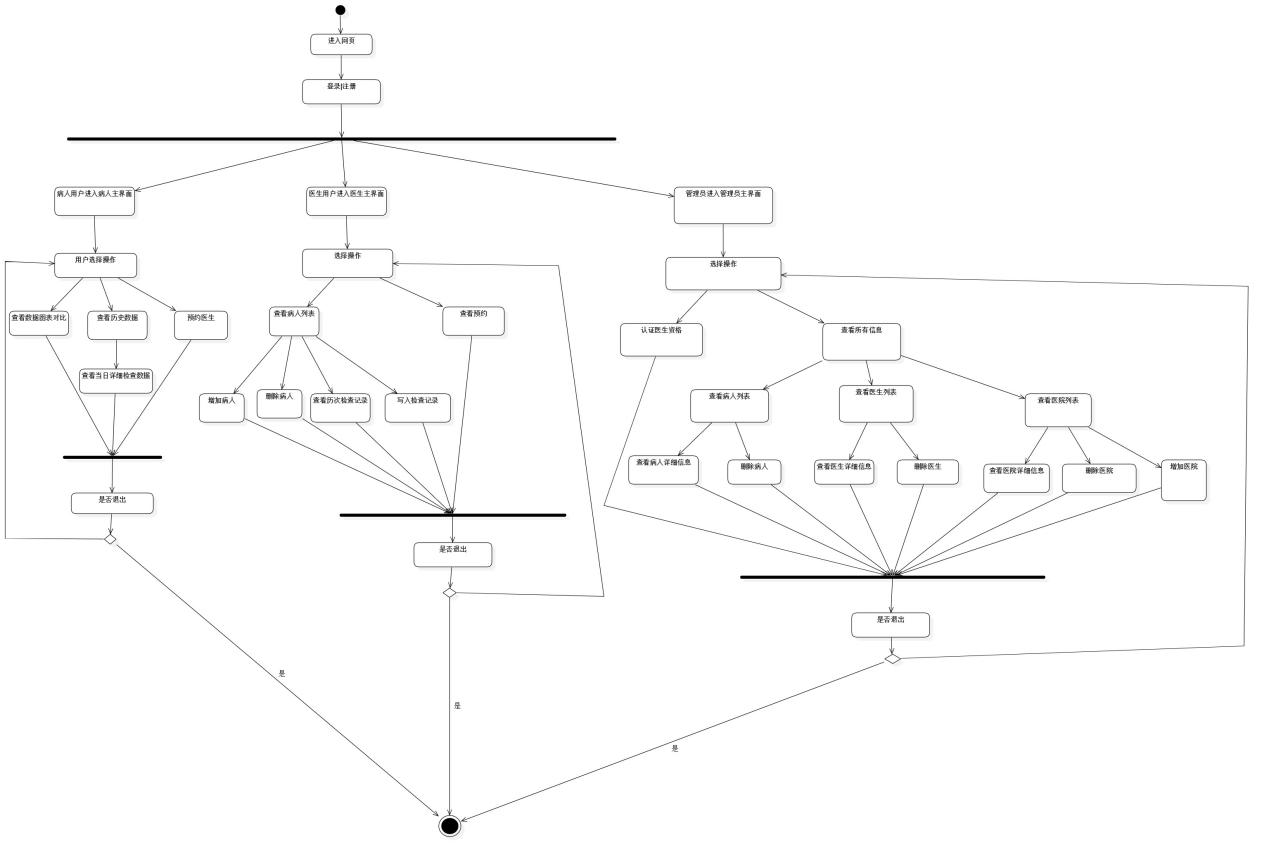
**作用范围**：身份经过认证的合格医生以及注册进入该系统的广大牙病患者。

### 2.2. 系统体系结构



本系统总体使用了springboot框架，使得开发人员不再需要样板化的配置，节省了开发时间，前端主要使用jquery+bootstrap+thymeleaf框架结构，对用户有良好的界面展示，用阿里云作为服务器，在阿里云服务器上配置了MySQL数据库，项目运行在一个稳定的环境。

### 2.3. 系统总体流程



系统总体流程如上所示，在用户选择进入主界面之后会要求用户首先进行登陆与注册，根据用户选择的身份然后进入相应的主界面。如果是病人则进入病人主界面，然后可以选择相应的三个操作查看数据图表对比，查看历史详细数据以及预约医生的操作，而医生用户则进入医生主界面，可以选择查看病人以及查看预约，在查看病人列表下又可以增加病人，删除病人，查看历次数据记录以及记录检查数据操作，而管理员则进入管理员界面，可以选择认证医生资格及查看所有信息操作，认证医生资格就是在医生注册时由管理员确认医生的身份是否合法，是否具有医生资格，选择查看所有信息可以查看到病人所有信息并且可以选择删除病人用户，查看医生所有信息可以查看医生详细个人信息以及删除医生信息，还有查看医院详细信息以及删除医院信息还有增加医院操作。三种类型的用户在结束操作之后都可以选择是否退出系统，是则退出，否则可以继续选择下一步操作。

### 2.4. 需求分析

#### 2.4.1.功能建模

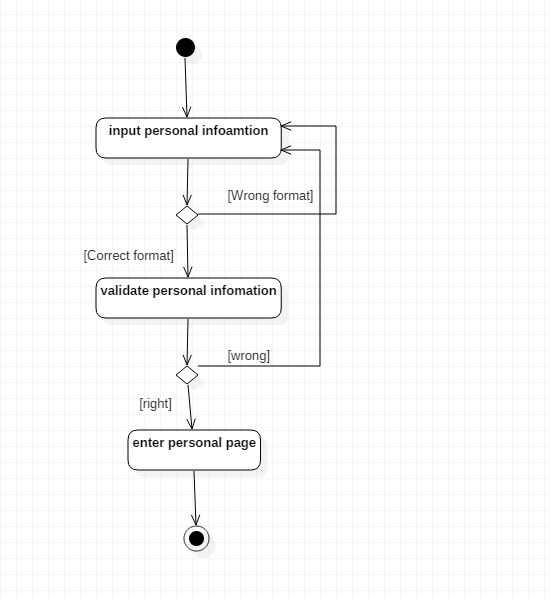
##### 2.4.1.1 SignIn

**用例名称：**SignIn

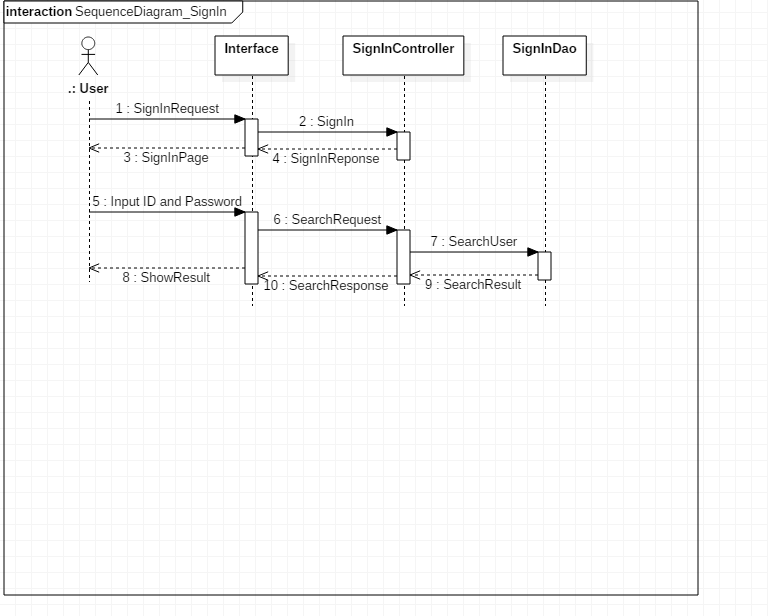
**用例说明：**用户在注册界面填写个人信息，选择用户角色（患者/医生），填写密码，点击注册按钮完成用户注册。

**简要描述：**用户填写完个人信息后，系统检查用户填写信息的格式，若格式不正确或者用户的账号密码不匹配则要求用户重新填写，若登陆成功则进入个人页面。

**活动图**



**时序图**



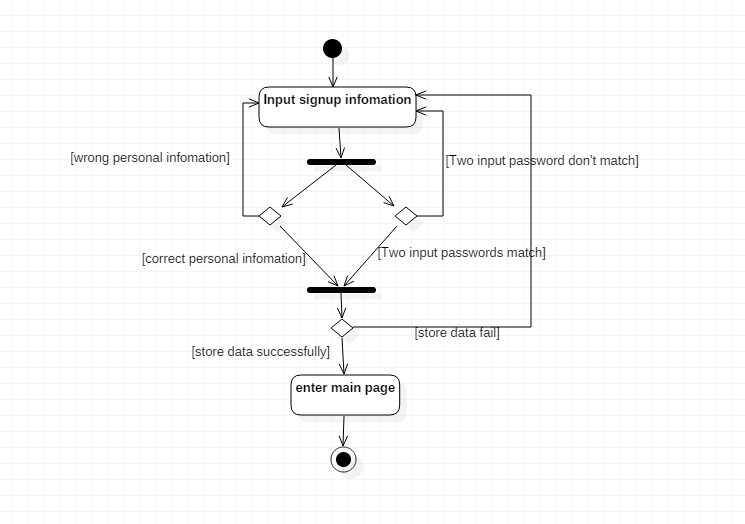
##### 2.4.1.2 SignUp

**用例名称：**SignUp

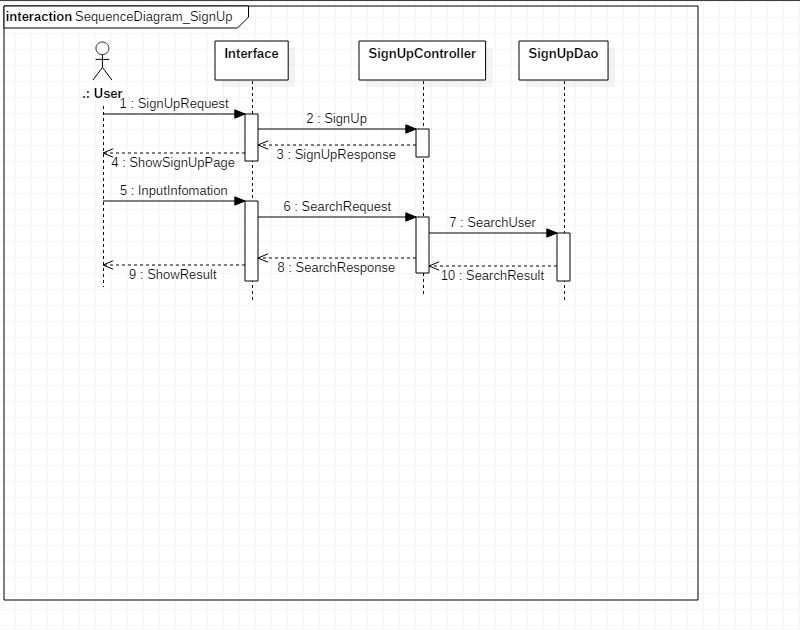
**用例说明：**用户在注册界面填写个人信息，选择用户角色（患者/医生），填写密码，点击注册按钮完成用户注册。

**简要描述：**用户在注册页面填写注册信息，如果用户个人信息填写错误/两次密码不符/存储数据错误，系统要求用户重新填写注册信息，注册成功则进入相应的主页面。

**活动图**



**时序图**



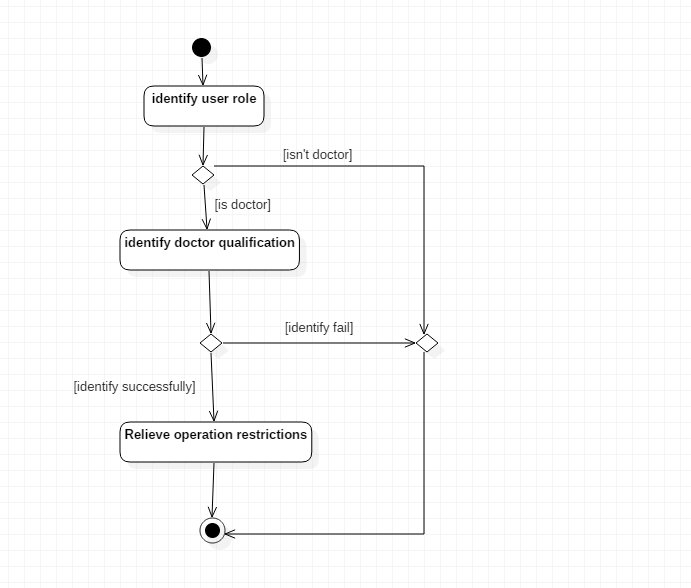
##### 2.4.1.3 Identify

**用例名称：**identify

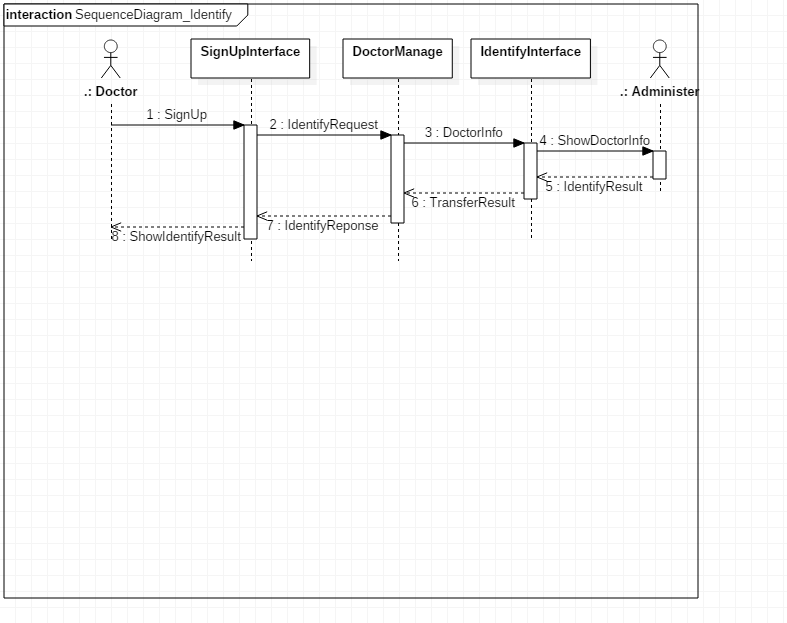
**用例说明：**管理员在医生注册成功后对其进行身份验证

**简要描述：**管理员界面根据显示医生的信息来进行资质认证，如果用户不是医生或者认证失败则无法进行医生的相关操作，如果是医生且认证成功则解除权限限制，医生可以进行诊断等操作。

**活动图**



**时序图**



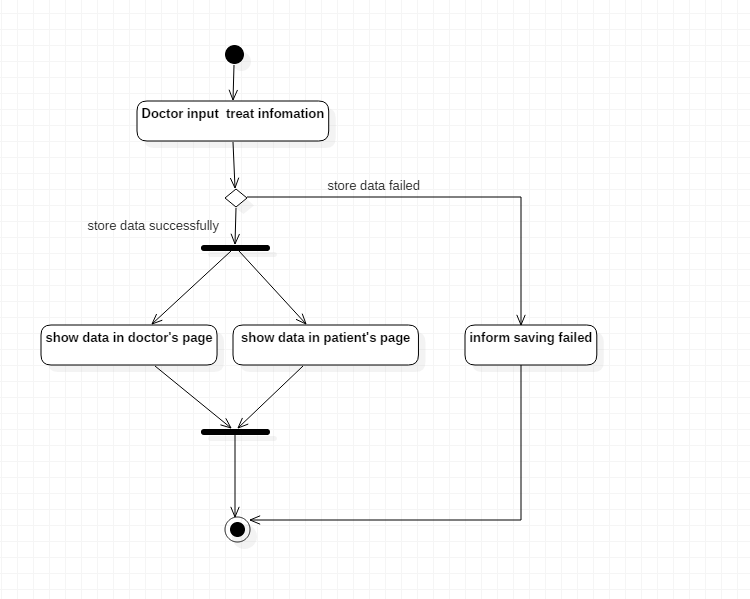
##### 2.4.1.4 Process data

**用例名称：**process data

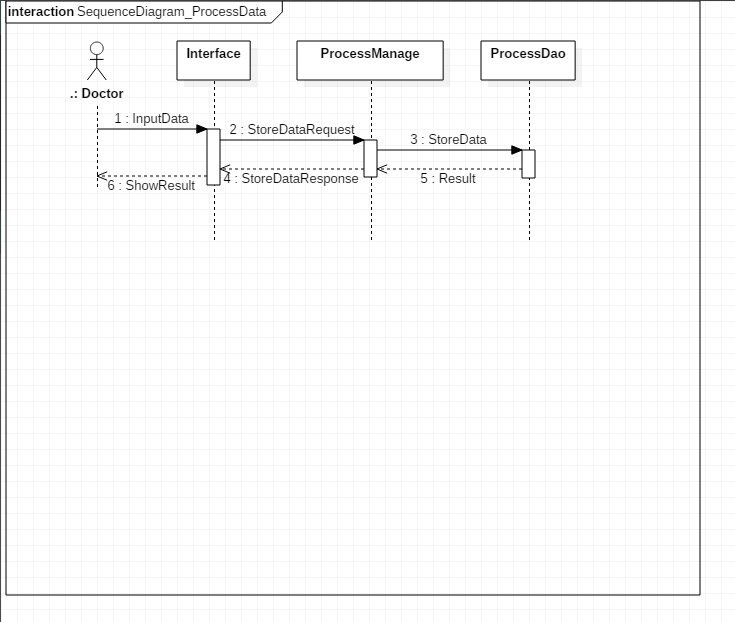
**用例说明：**医生完成诊断后提交诊断表，系统根据医生提交的信息进行数据的计算统计，将结果显示在展示界面。

**简要描述：**医生填写完诊断表的信息后，若数据存储成功则在医生和用户界面显示相关的数据信息，反之系统提示保存数据失败。

**活动图**



**时序图**

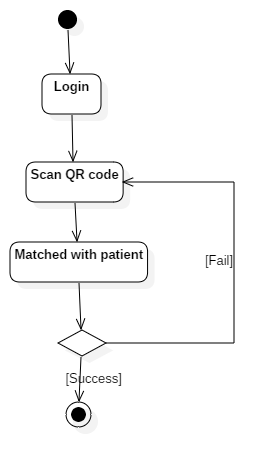


##### 2.4.1.5 Scan QR code

**用例名称**：scan QR code

**用例说明**：医生先登录系统，在病人出示二维码后扫描二维码，若是匹配成功，则完成，若是匹配失败，回到扫描二维码的步骤。

**活动图**

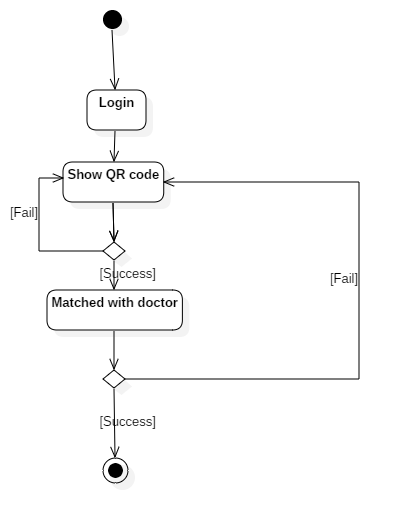


##### 2.4.1.6 Show QR code

**用例名称**：show QR code

**用例说明**：病人先登录系统，进入二维码界面。若是失败，则重新进入该界面；若是成功，则出示二维码给医生等待扫描。医生扫描后，若是匹配成功，则完成；若是匹配失败，则回到二维码界面等待再次扫描。

**活动图**

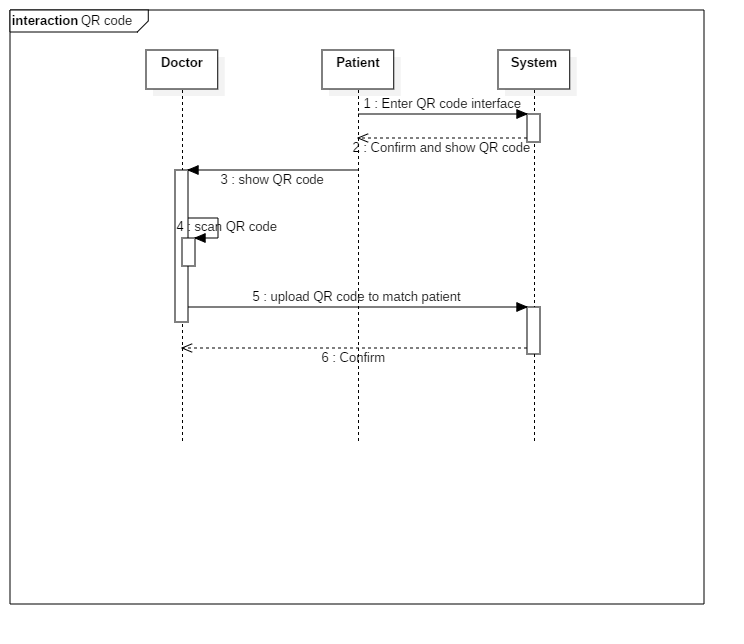


##### 2.4.1.7 QR code

**用例名称**：QR code

**用例说明**：对于二维码用例，病人先进入二维码界面并向系统发送信息请求二维码，系统收到后确认并返回二维码。然后，病人出示二维码给医生，医生扫描二维码后，将二维码信息提交到系统以匹配病人，系统收到医生提交的信息后，返回匹配结果。

**序列图**



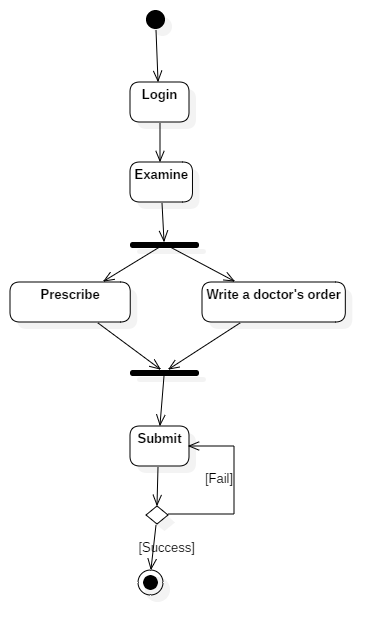
##### 2.4.1.8 Diagnose

**用例名称**：Diagnose

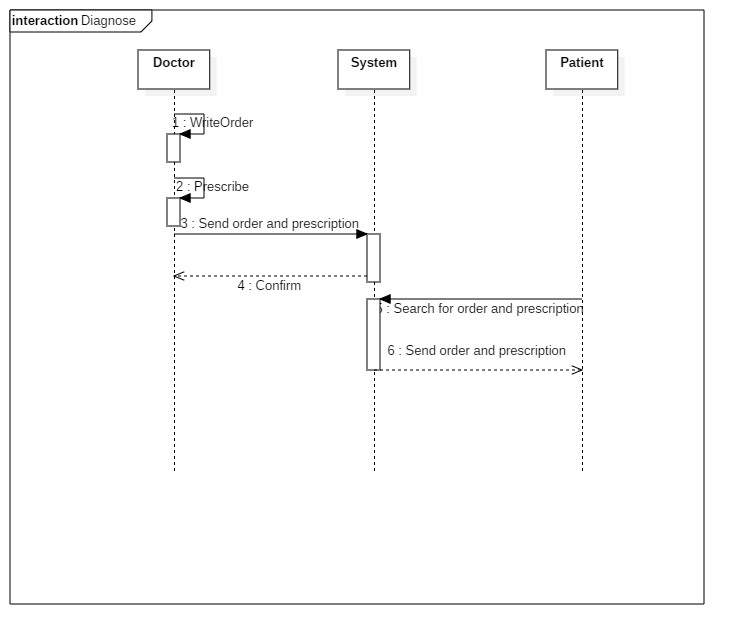
**用例说明**：医生先登录系统，在完成给病人的检查后，可开出处方并且写下医嘱，完成后提交给系统，若是提交成功，则完成；若是提交失败，则回到提交前的界面。然后病人可以向系统查询医嘱和药方，系统收到病人的请求后返回相应信息。

**简要描述：**对于开处方用例，医生在检查后先写好医嘱和处方，然后提交到系统等待确认，系统收到后返回确认信息。然后病人可以向系统查询医嘱和药方，系统收到病人的请求后返回相应信息。

**活动图**



**序列图**



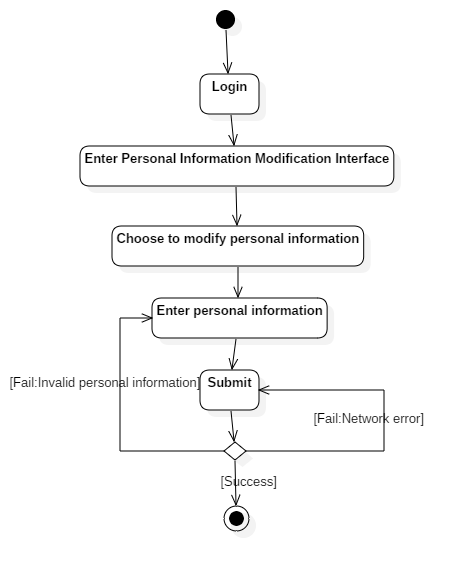
##### 2.4.1.9 Modify personal information

**用例名称**：Modify personal information：

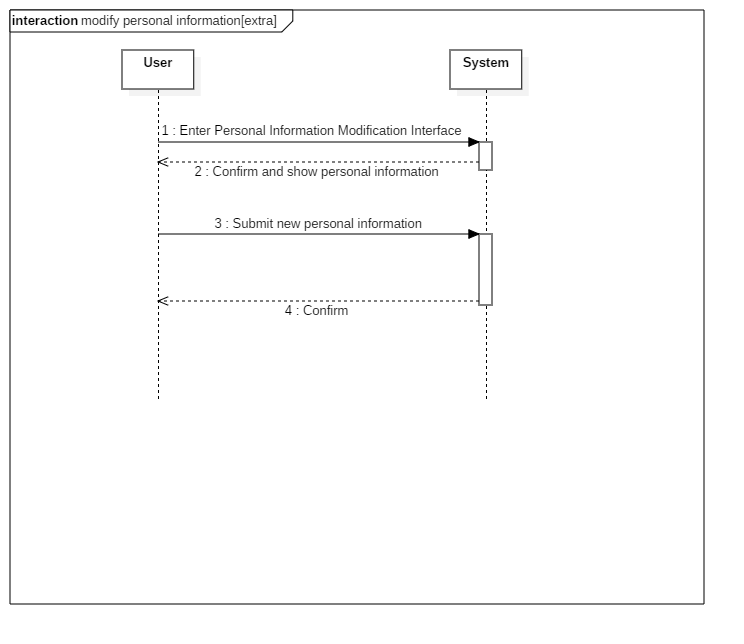
**用例说明**：用户登录系统后，进入个人信息修改界面，选择个人信息修改按钮，输入新的个人信息并提交，等待系统审核，若是网络错误，返回提交界面；若是个人信息不符合要求未通过审核，回到个人信息修改界面；若审核成功，则完成修改。

**简要描述：**对于修改个人信息用例，用户首先进入修改界面，系统收到请求后返回确认信息以及用户已存储的信息。用户修改信息后提交到系统，系统收到请求后返回确认信息。

**活动图**



**序列图**



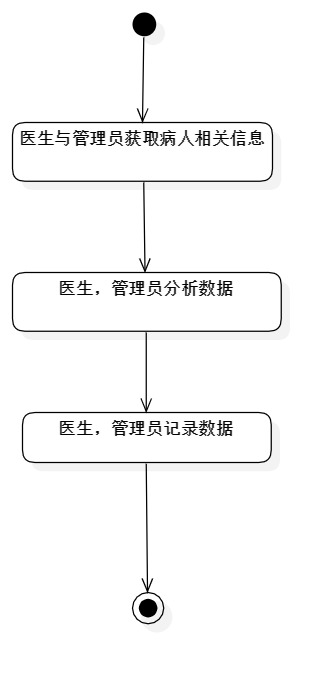
##### 2.4.1.10 Analyse data

**用例名称**：analyse\_data

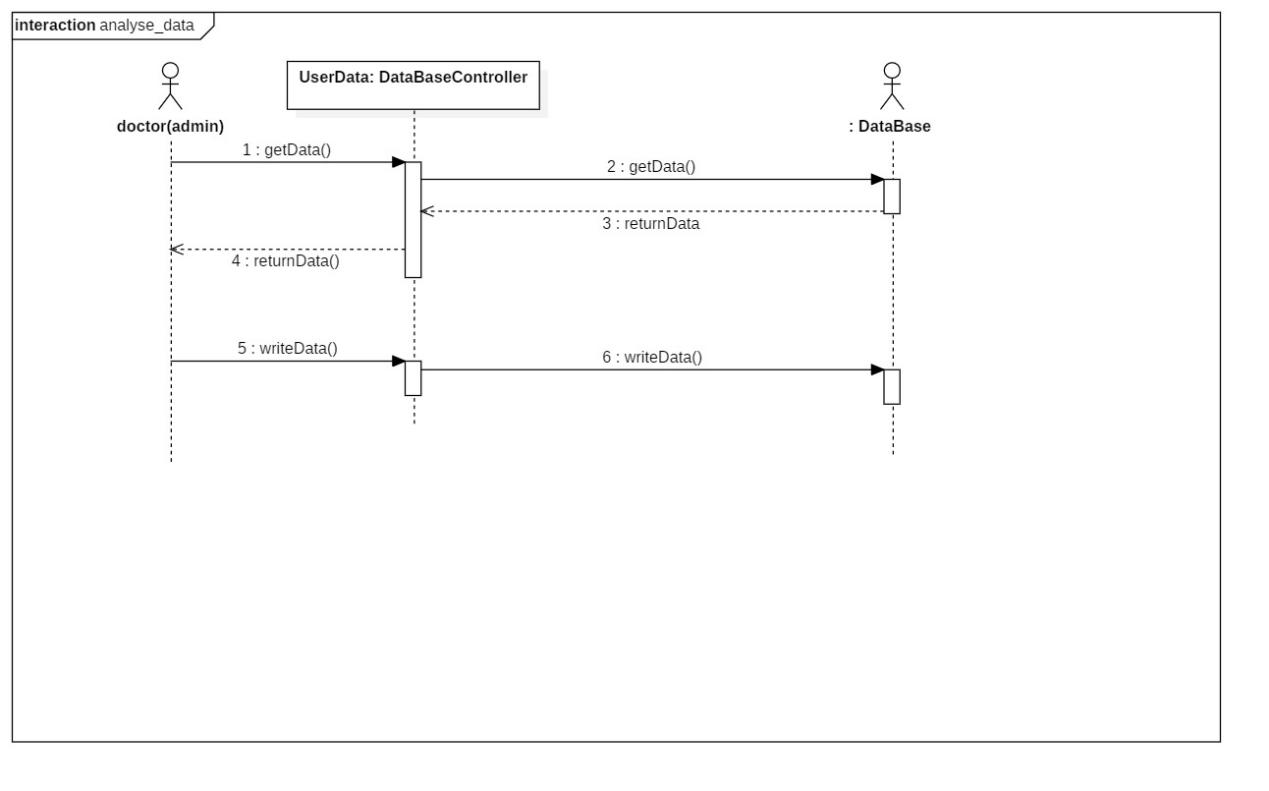
**用例说明**：医生与管理员登陆系统之后，可以获取到病人的相关数据（比如说病人的年龄性别），然后医生与管理员可以根据这些数据进行分析然后将分析结果记录到数据库之中。

**简要说明：**医生，管理员登陆系统申请得到用户数据，系统后端Controller类会从数据库获取信息然后返回给医生以及管理员，然后管理员医生将分析结果写好交由系统通过Controller类上传至数据库。

**活动图**



**序列图**



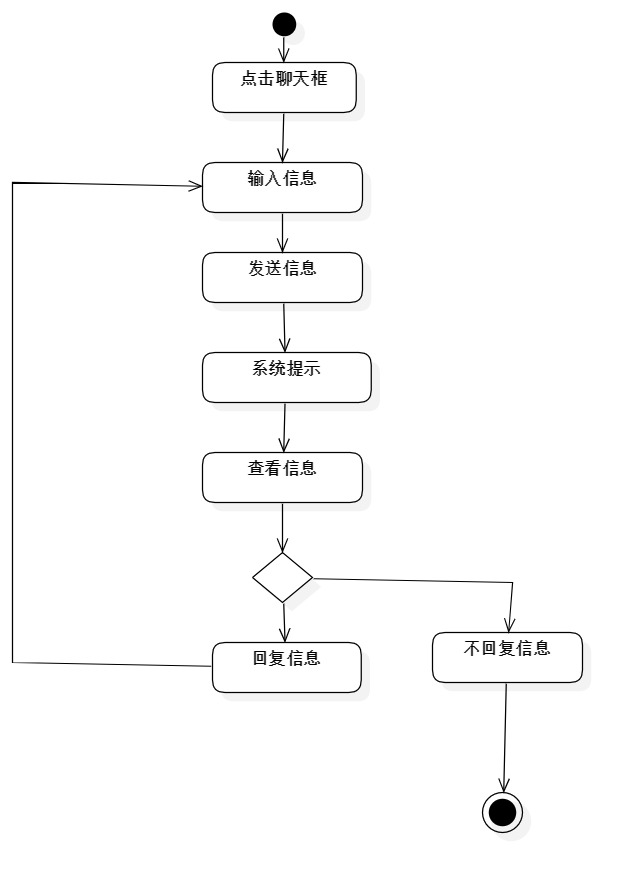
##### 2.4.1.11 Chat

**用例名称**：chat

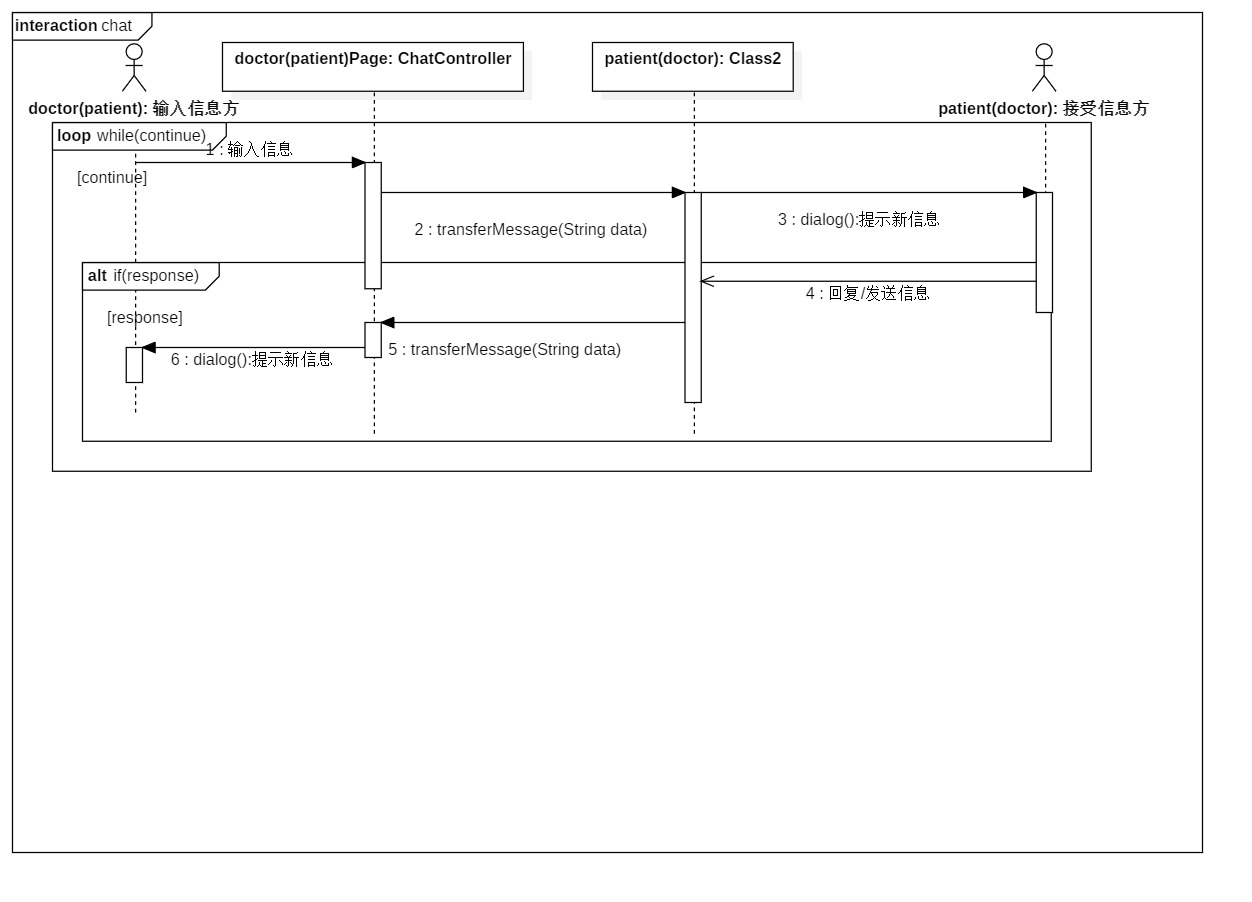
**用例说明**：此用例仅限于病人与医生之间进行交流。医生与病人登陆系统之后，一方点击聊天框，输入信息，选择发送信息，然后系统会提示另一方接受到信息，然后另一方选择查看信息，接着另一方可以选择回复信息就循环这个过程，或者不回复。

**简要描述：**医生病人其中用户一方选择在界面输入信息，然后发送信息，然后由相应的controller类发送信息到对方页面，提示对方接受新信息，选择回复或不回复。

**活动图**



**序列图**

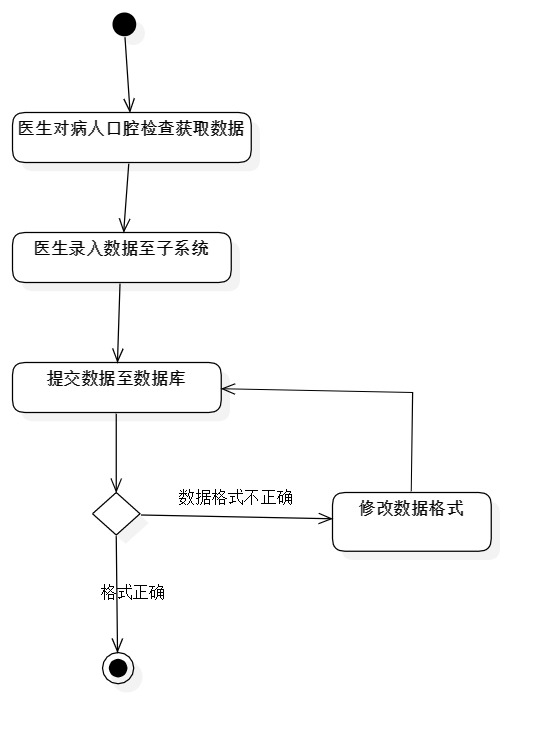


##### 2.4.1.12 Examine

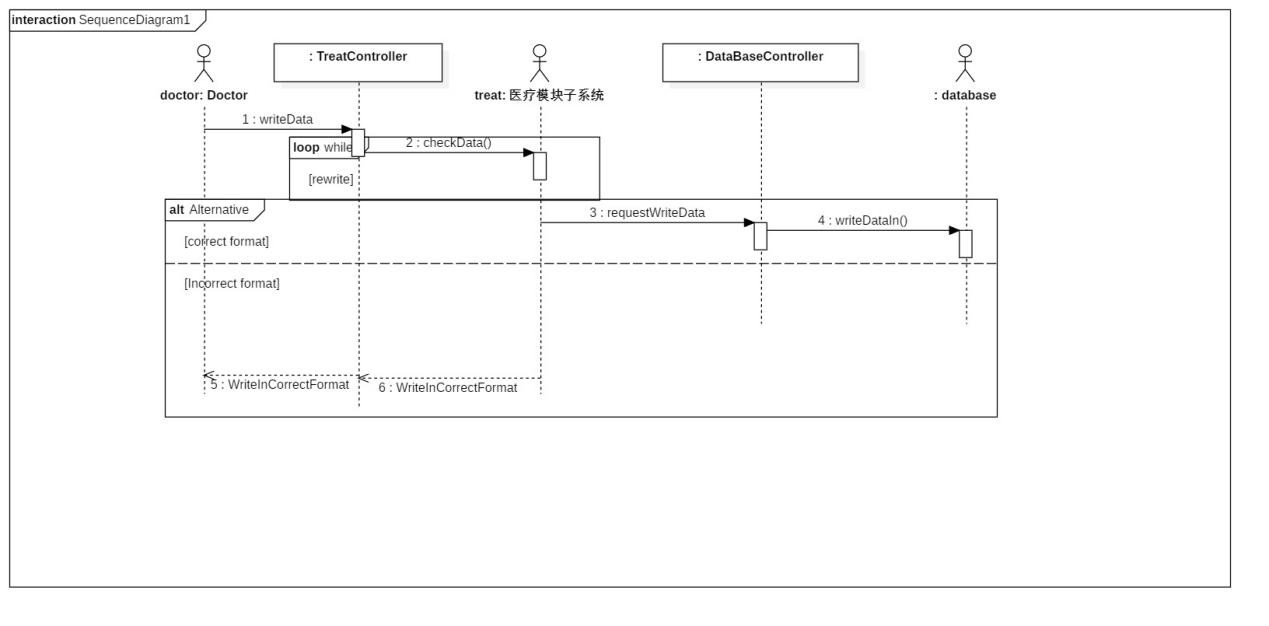
**用例名称**：examine

**用例说明**：医生登陆至医生界面，在对病人结束检查后获取到相关数据，然后将数据录入子系统，选择上传至数据库，可能被检测出数据格式错误，需要不断修改数据格式直至数据格式正确。

**简要描述：**医生将数据写好上传数据，Controller类会检测数据格式的正确性，错误就要重写至数据格式正确。接着子系统会通过调用数据库相关的Controller类将数据上传至数据库。



**序列图**



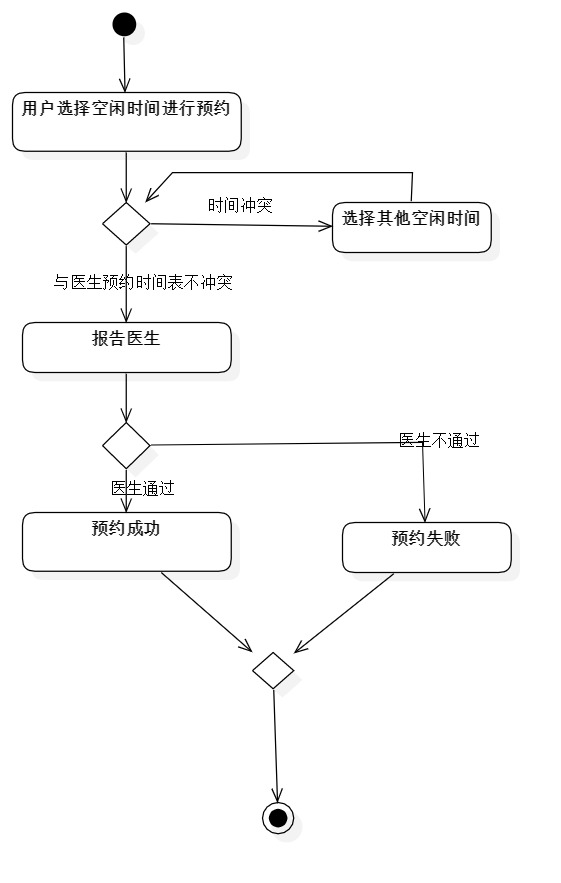
##### 2.4.1.13 Make an appointment

**用例说明**：make an appointment

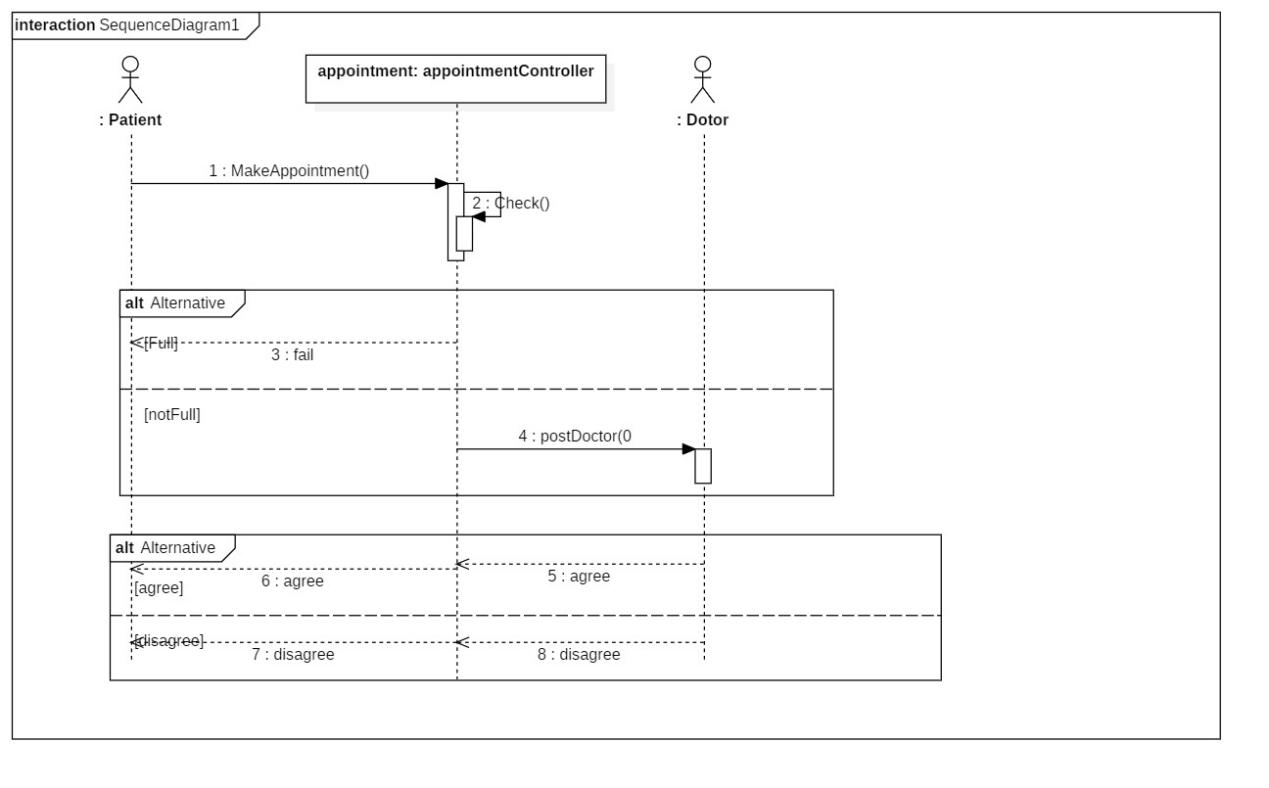
**用例说明**：用户选择时间与医生预约，系统检测医生在此时刻是否已有预约，有则重新选择预约时间，无则将预约请求上传至医生界面，医生选择是否通过，是则预约成功记录至数据库，否则失败。

**简要描述：**病人预约时间，appointmentController类检查时间是否冲突，不冲突之后上报医生，医生选择是否同意预约，同意或不同意反馈信息至病人。

**活动图**



**序列图**



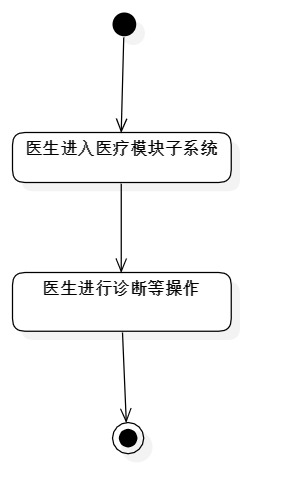
##### 2.4.1.14 Treat

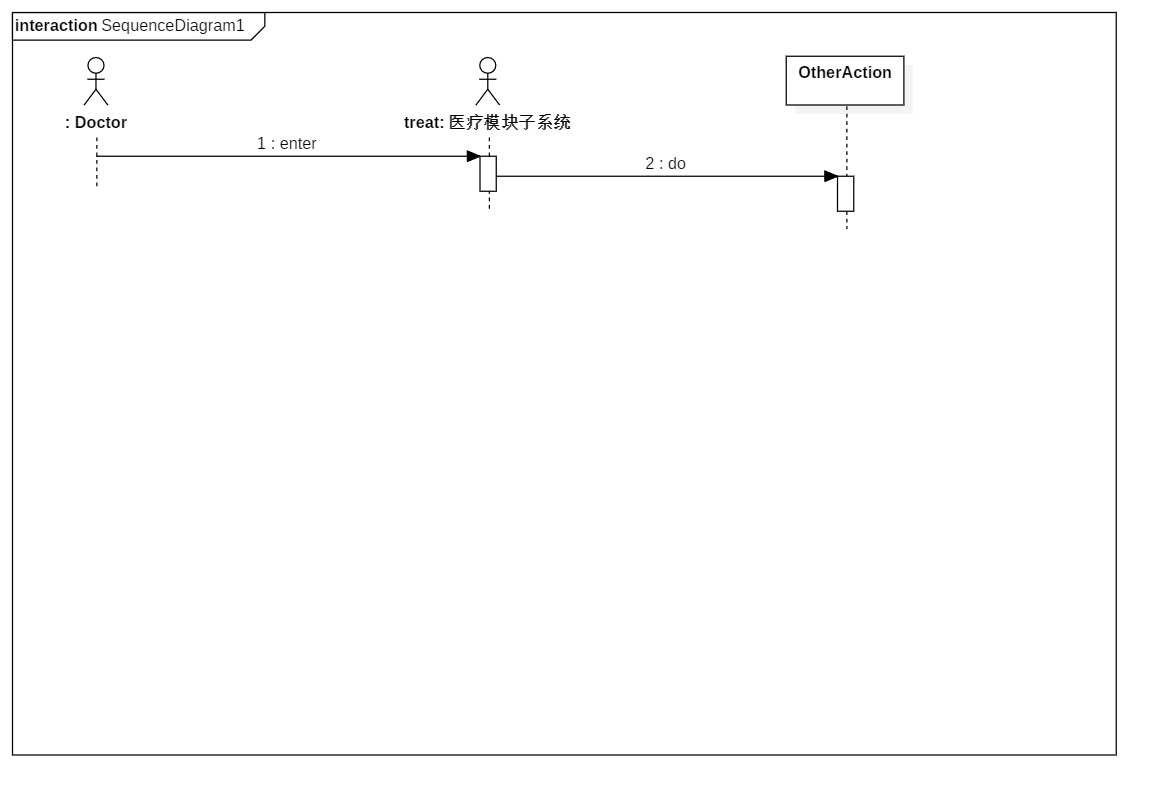
**用例名称**：treat

**用例说明**：医生登陆到系统后，可选择进入医疗模块子系统，接着医生可进行诊断等操作。

**简要描述：**医生进入医疗模块子系统，然后可以进行诊断等操作。

**活动图**



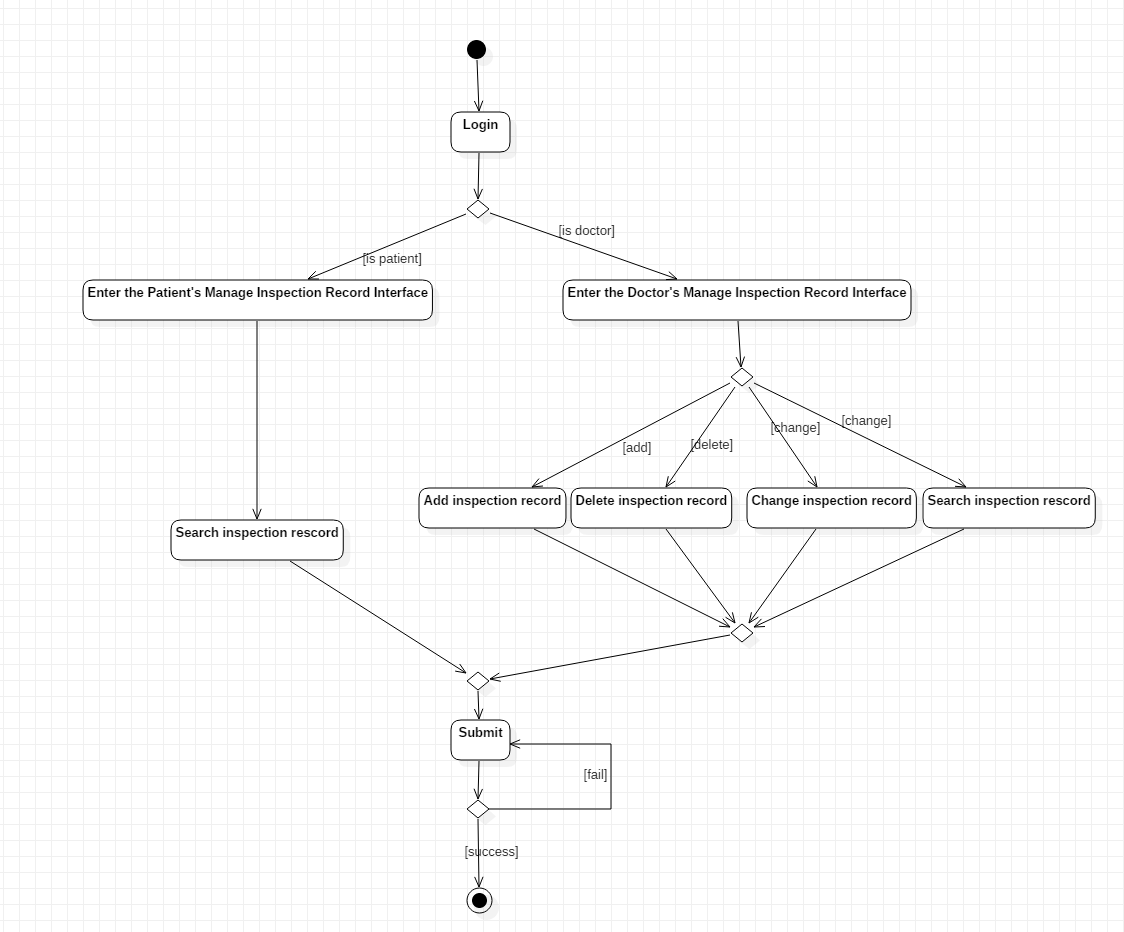
**序列图**

##### 2.4.1.15 Manage inspection record

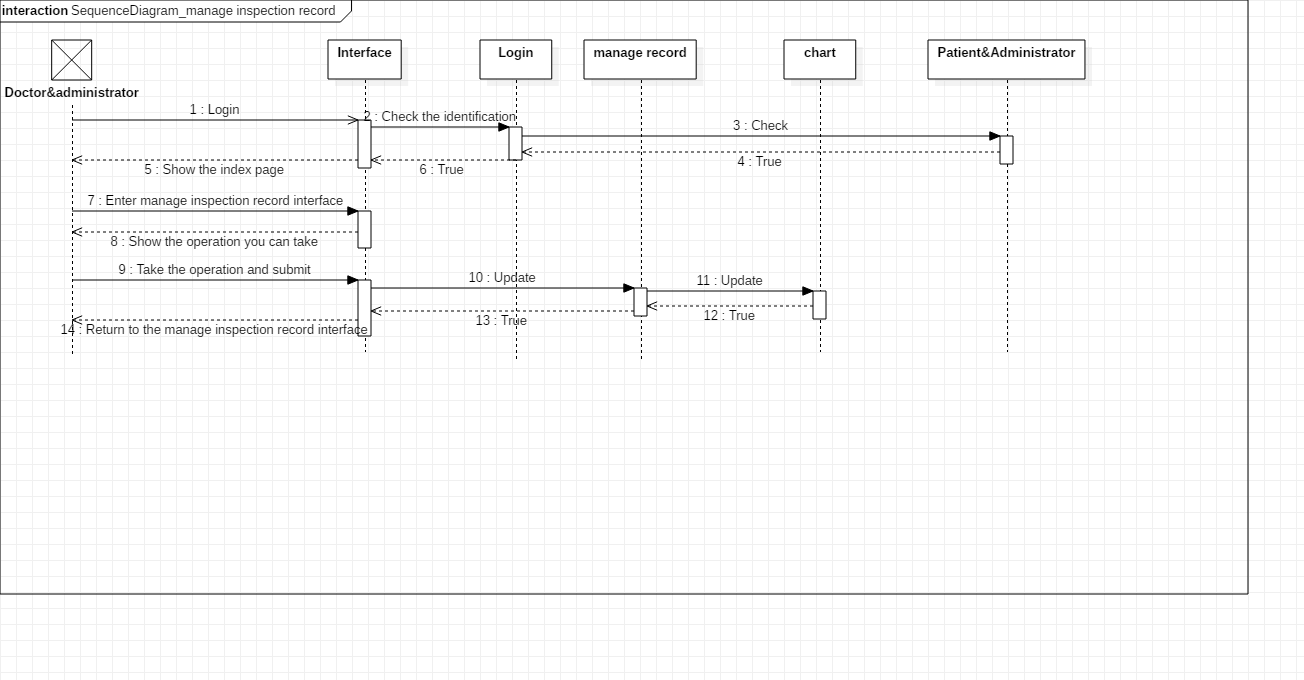
**用例名称**：manage inspection record

**用例说明**：用户先登录，根据用户的类型显示不同的管理检查记录的界面，病人可以查询记录，医生和管理员能够添加，删除，更改和查询记录，操作完毕后返回管理记录的界面。

**活动图**



**序列图**

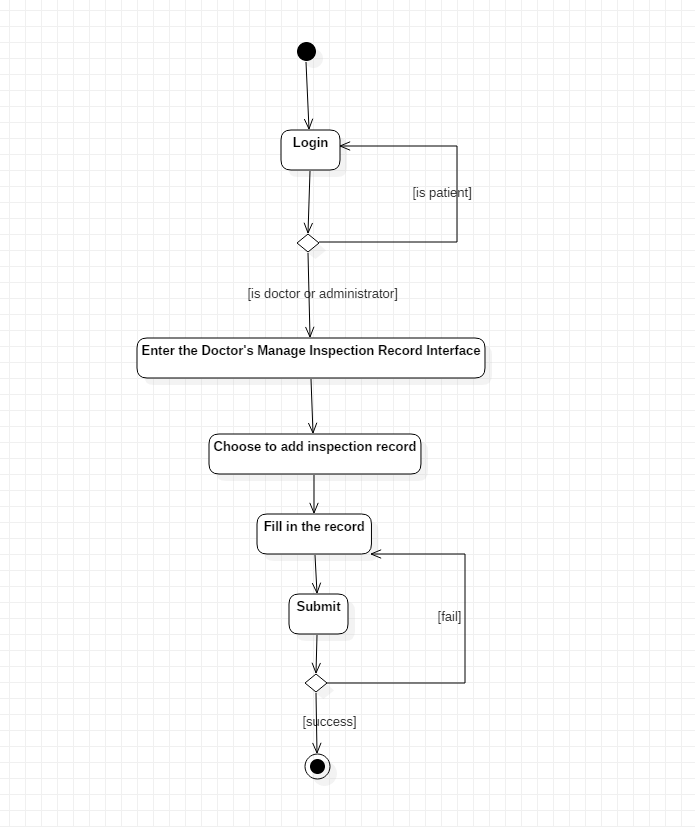


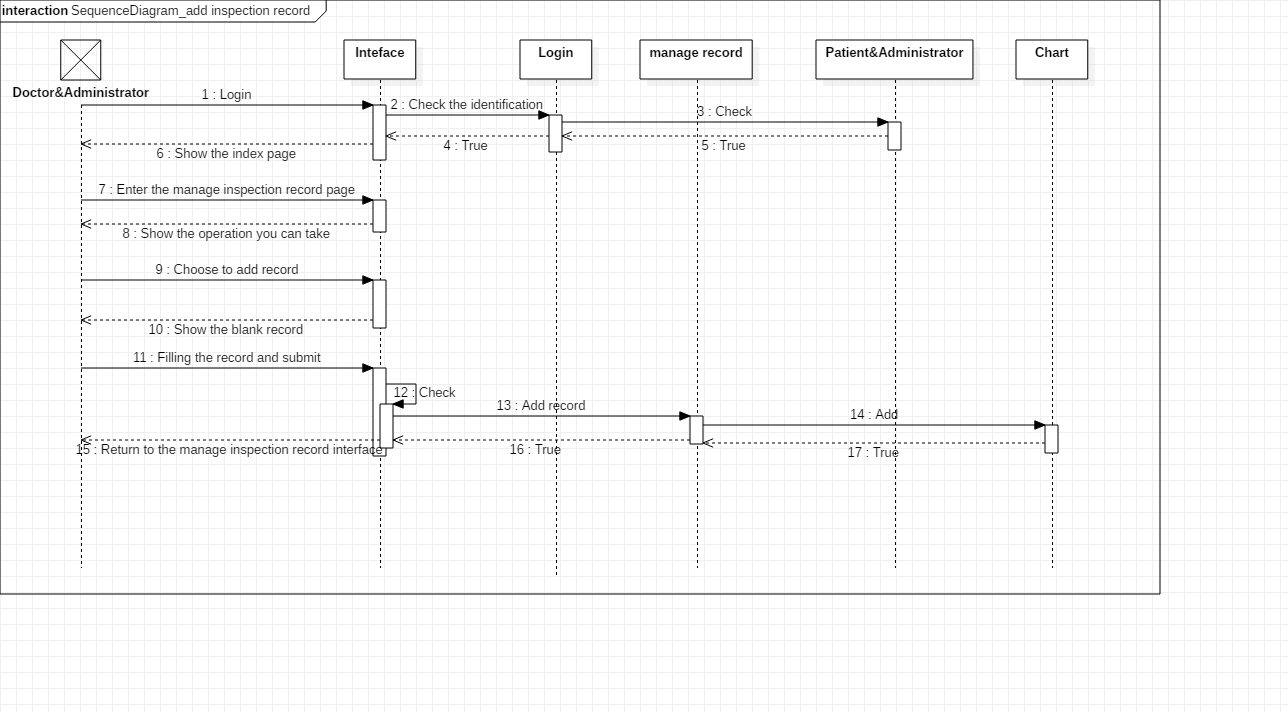
##### 2.4.1.16 Add inspection record

**用例名称**：add inspection record

**用例说明**：用户先登录，判断是否是医生和管理员，否则不能进行添加记录的操作，进入后在管理检查记录的界面点击添加检查记录，在弹出的空表格内完善信息并提交，系统判断输入内容是否违约，若违约则重新更改，若不违约将表格提交，更新数据库内容，用户则返回到管理检查记录的页面。

**活动图：**

**序列图**



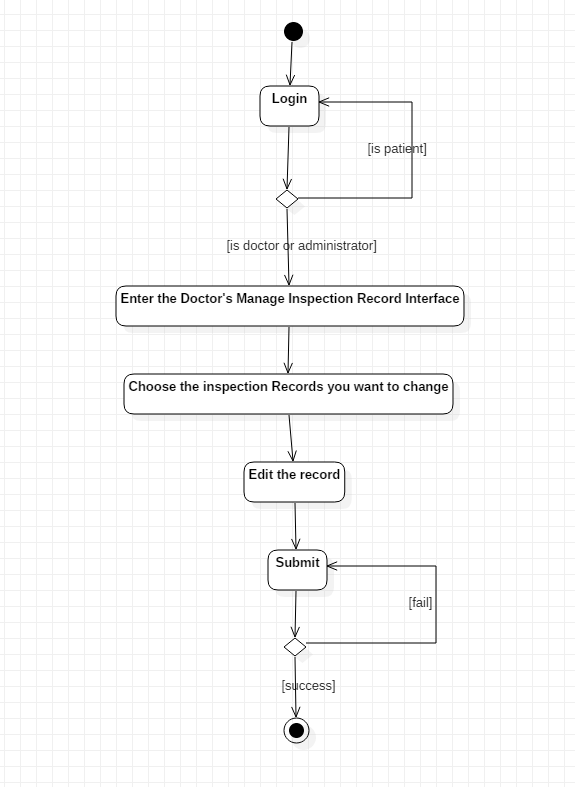
##### 2.4.1.17 Change inspection record

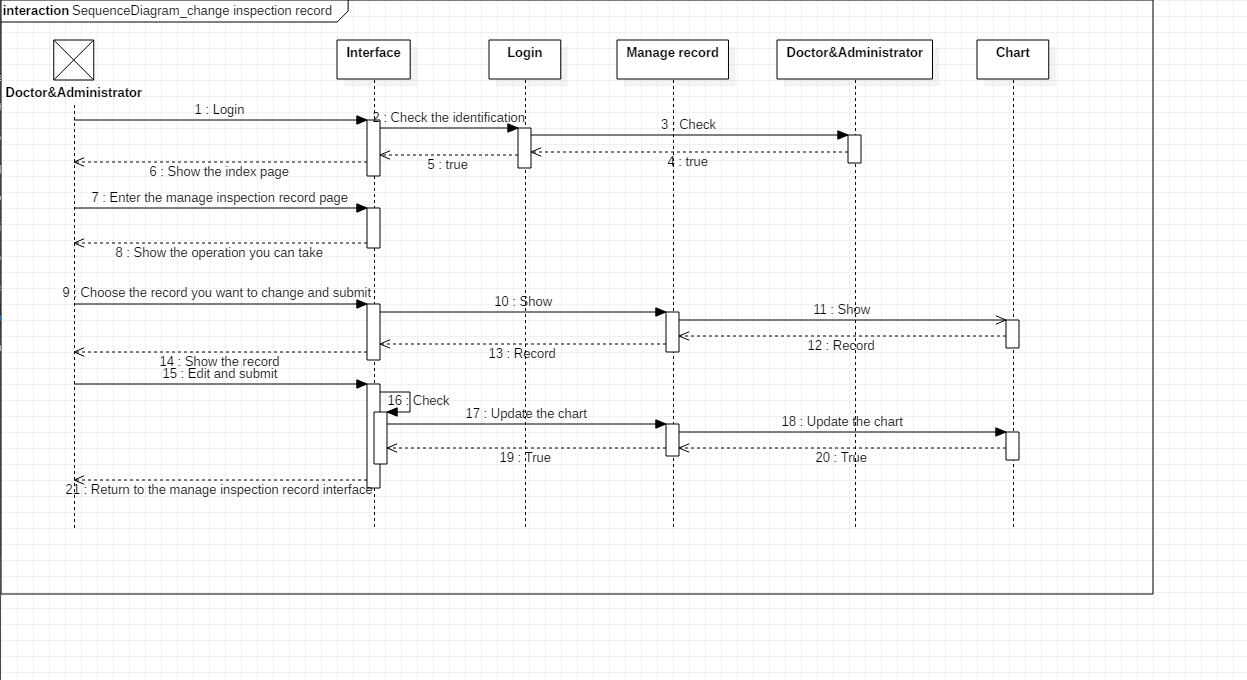
**用例名称**：change inspection record

**用例说明**：用户先登录，判断是否是医生和管理员，否则不能进行添加记录的操作，进入后在管理检查记录的界面

找到想要更改的记录，点击更改按钮，在弹出的可修改表格中修改信息并提交，系统判断输入内容是否违约，若违约则重新更改，若不违约将表格提交，更新数据库内容，用户则返回到管理检查记录的页面。

**活动图**



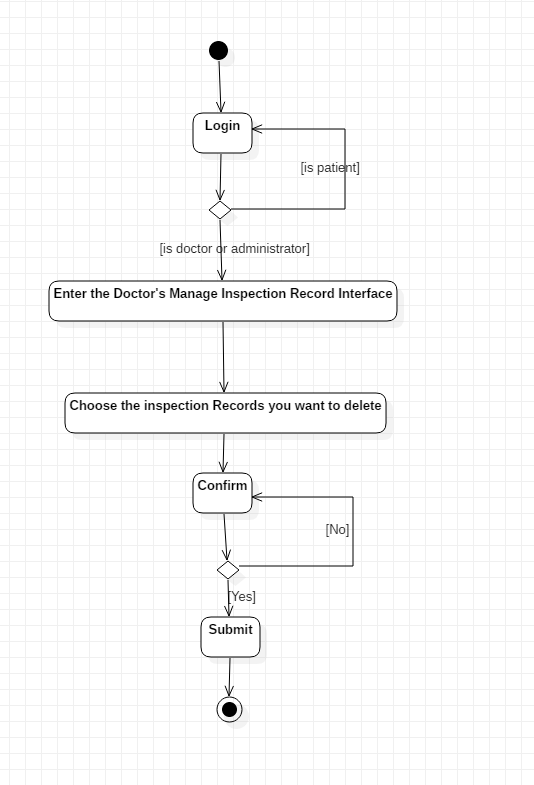
**序列图**

##### 2.4.1.18 Delete inspection record

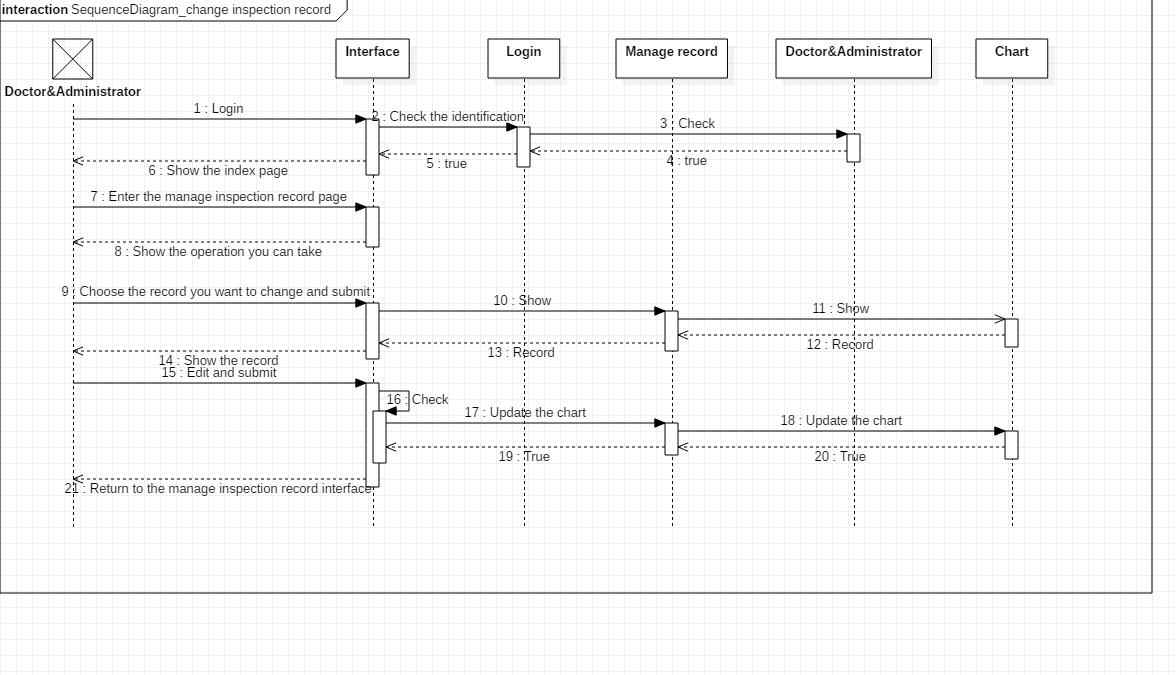
**用例名称**：delete inspection record

**用例说明**：用户先登录，判断是否是医生和管理员，否则不能进行添加记录的操作，进入后在管理检查记录的界面找到想要删除的记录，点击删除，弹出的警示框中点击确认，系统更新数据库内容，用户则返回到管理检查记录的页面。

**活动图**



**序列图**

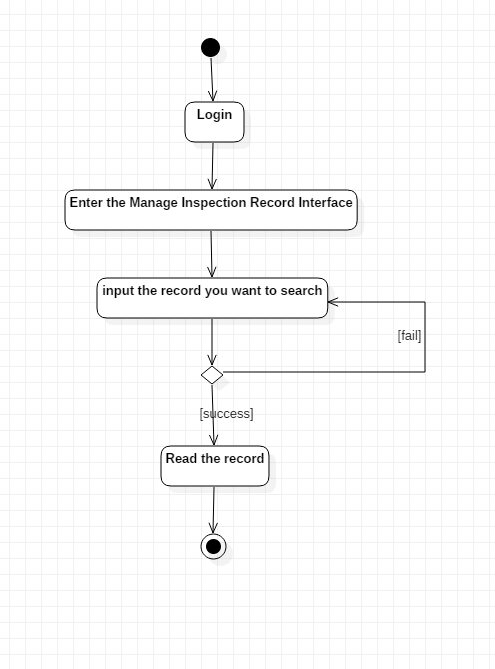


##### 2.4.1.19 Search inspection record

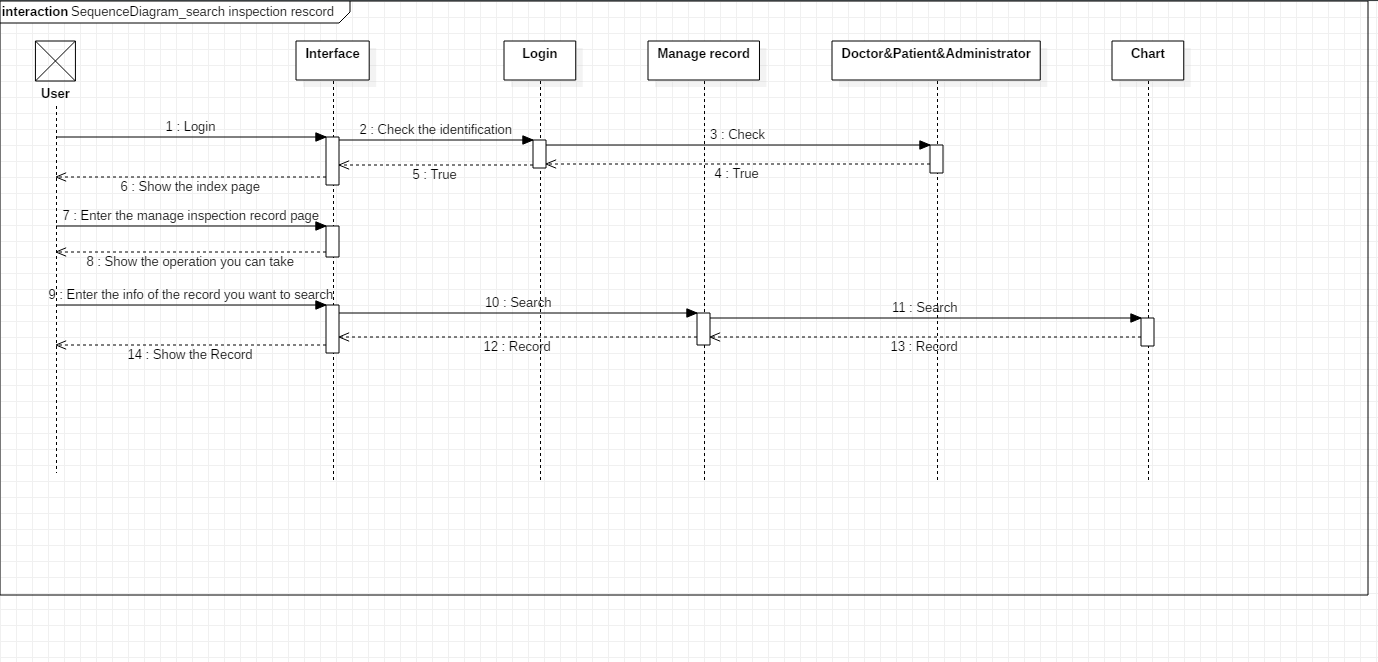
**用例名称**：search inspection record

**说明**：用户登陆后，进入管理检查记录的页面，在搜索框中输入与要查找记录有关的信息，在出现的列表中找到相应的检查记录，点击可进行查看。

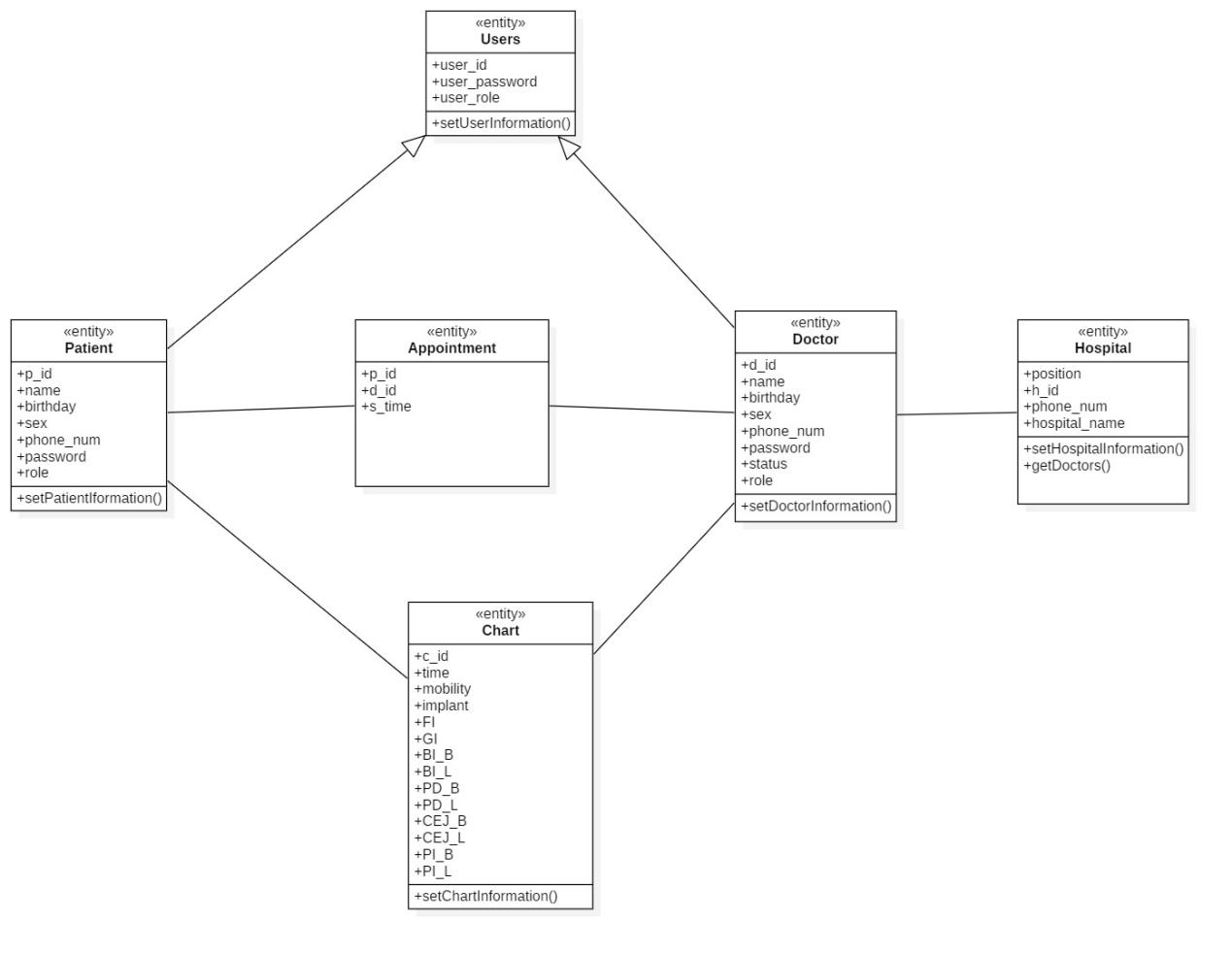
**活动图：**



**序列图**



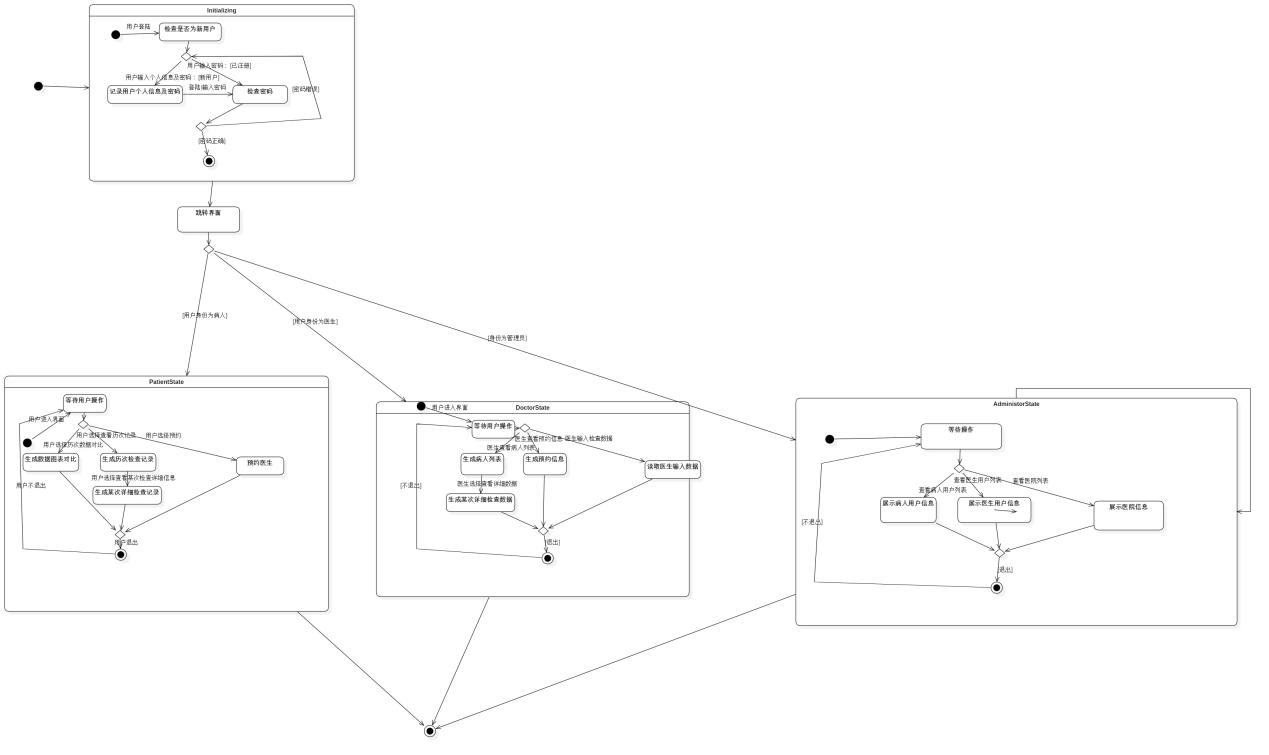
#### 2.4.2.数据建模



数据建模如图所示，有Patient类，主要功能是设置并展示病人用户的个人信息，Users类用于记录所有注册过的用户的id,以及一些其他的用户信息（诸如用户的身份：病人或是医生），而Doctor类则用来设置及展示医生的个人相关信息，Doctor及Patient都继承自User类。Appointment类则用来记录预约的相关信息，其属性主要有医生的id及病人id以及预约时间。Chart类用来记录检查数据，医生要填写这张表，病人可以选择查看这张表。Hospital类则负责记录以及设置医生所属医院的相关信息。

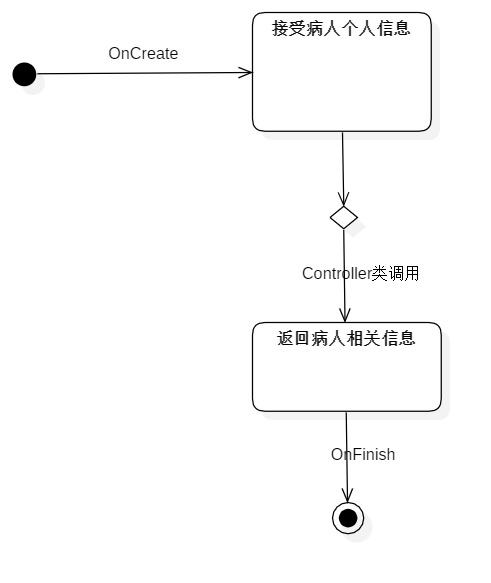
#### 2.4.3.行为建模

##### 2.4.3.1系统状态图：



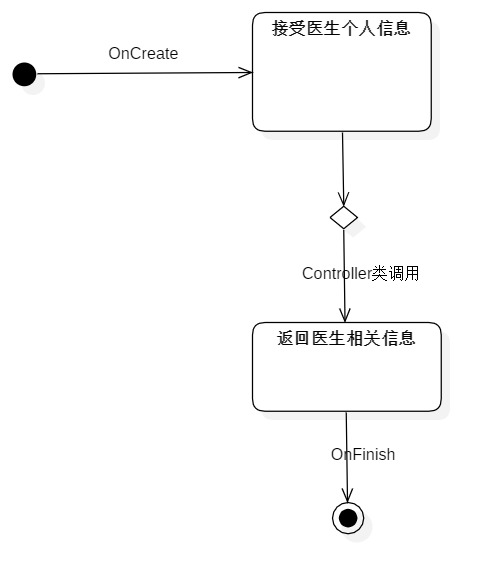
系统总体状态图如图所示，在任意用户进入界面之后选择登录，系统检测并提示是否为新用户，用户若为新用户选择注册，填写个人信息以及密码，若不是新用户则登录，输入密码，系统检查密码是否正确，若不正确系统返回用户继续输入密码。在成功登录之后进入系统，系统根据用户身份跳转。若用户身份为病人跳转进入病人主界面，然后系统等待用户操作，用户选择查看数据图表对比，预约医生或是检查记录功能，则系统根据相应选择生成相应界面给用户，用户选择退出则结束，否则系统继续处于等待用户操作状态。若用户身份为医生，则系统跳转进入医生主界面，等待用户操作，接着系统根据用户的选择（查看病人列表，查看预约信息或是选择输入检查数据）生成对应的界面，若是医生选择输入数据，系统要读取数据并保存。若医生选择退出则结束，否则系统继续处于等待操作状态。若是管理员状态，则进入管理员界面，系统等待用户操作，若用户选择操作（查看所有病人信息，查看医生信息，查看医院信息），则系统根据对应操作生成界面给用户，然后若用户退出，则结束，若用户不退出，则系统继续等待用户其他操作。

##### 2.4.3.2 Patient类状态图：



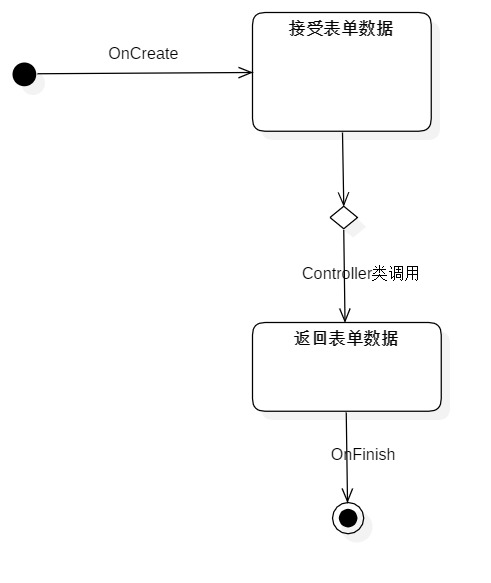
**说明：**Patient类被Controller调用创建后接受从数据库传来的用户个人信息，然后根据Controller类的要求返回一些信息给Controller类，Controller类调用结束之后，Patient类生命周期结束。

##### 2.4.3.3 Doctor类状态图：



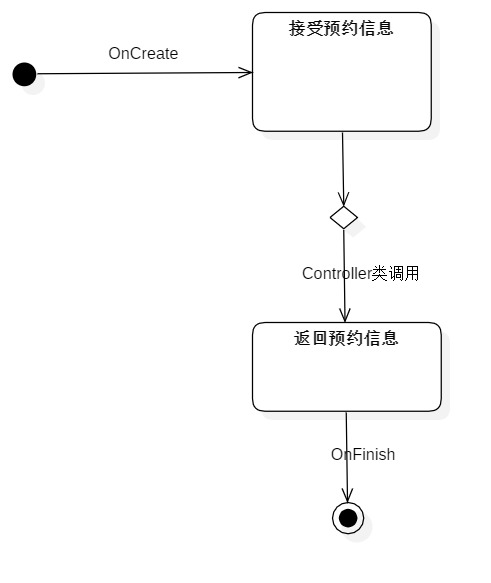
**说明：**Doctor类被Controller调用创建后接受从数据库传来的用户个人信息，然后根据Controller类的要求返回一些信息给Controller类，Controller类调用结束之后，Doctor类生命周期结束。

##### 2.4.3.4 Chart类状态图：



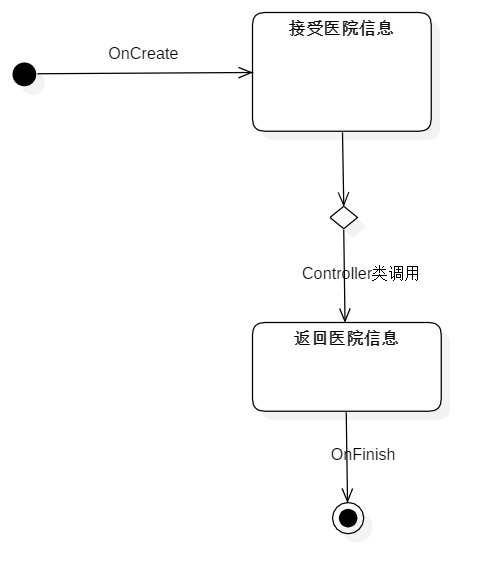
**说明：**Chart类被Controller调用创建后接受从数据库传来的表单数据，然后根据Controller类的要求返回数据给Controller类，Controller类调用结束之后，Chart类生命周期结束。

##### 2.4.3.5 Appointment类状态图：



**说明：**Appointment类被Controller调用创建后接受从数据库传来的预约信息，然后根据Controller类的要求返回数据给Controller类，Controller类调用结束之后，Appointment类生命周期结束。

##### 2.4.3.6 Hospital类状态图：



**说明：**Hospital类被Controller调用创建后接受从数据库传来的医院信息，然后根据Controller类的要求返回数据给Controller类，Controller类调用结束之后，Hospital类生命周期结束。

## 3. 非功能需求

### 3.1. 安全性

1. 严格权限访问控制，用户在经过身份认证后，只能访问其权限范围内的数据，只能进行其权限范围内的操作，如患者只能查询检查记录不能修改。不同的用户具有不同的身份和权限，需要在用户身份真实可信的前提下，提供可信的授权管理服务，保护数据不被非法/越权访问和篡改，要确保数据的机密性和完整性。

2. 提供运行日志管理及安全审计功能，可追踪系统的历史使用情况。

3. 医生的注册必须经过管理员的认证，即其医生身份合法性必须得到认证。

### 3.2. 可移植性

1. 网页端支持各大主流浏览器，如firefox、chrome等。
2. 网页端采用响应式布局，可在电脑平台与手机平台上使用。
3. 使用微信小程序，在可使用微信的Android、IOS、Windows Phone上均可使用。

### 3.3. 易用性

1. 界面设计对用户友好，功能方便。
2. 流程简单但是逻辑完整，用户不需要对电脑手机有过多了解也能熟练使用。
3. 采用了微信小程序的简易用户端，方便用户使用。

### 3.4. 可维护性

1. 由于本项目开发中涉及多次迭代，后期出现的需求变更可能导致现有设计的变更，因此本项目需充分考虑系统的可维护性能。
2. 系统采用前后端分离的架构，在开发过程中遵循“高内聚低耦合”的原则，保证各个模块间的松耦合性。
3. 代码加入详细的注释，为设计模式写好文档，方便维护。

### 3.5性能要求

#### 3.5.1时间特性要求

1. 数据库的更新涉及网络数据传输，在网速正常的条件下，数据库更新实践不应超过1s，网络状况较差的情况下，不应超过4s。
2. 界面更新的传送时间应低于1s。

#### 3.5.2输入输出要求

* 输入要求

牙周数据的输入应该符合要求：

|  |  |  |
| --- | --- | --- |
| 数据类型 | 数据输入要求 | 备注 |
| mobility（摇动度） | i ii iii | 一颗牙一个点 |
| implant(植体） | bool | 一颗牙一个点 |
| FI(根分叉） | i ii iii |  |
| GI（牙龈指数） | 0-3 | 一颗牙一个点 |
| BI（出血） | 0-5 | 一颗牙三个点 |
| PD | -1000~+1000 | 一颗牙三个点 |
| CEJ（龈缘） | -1000~+1000 | 一颗牙三个点 |
| PI（牙菌斑） | 0-3 | 一颗牙三个点 |

* 输出要求

在给病人展示详细数据表格时要严格按照顺序将数据呈现在表格之上，不可以有数据遗漏或是顺序的错误。

## 4. 运行环境规定

### 4.1. 设备

1. **用户端：**

**用户电脑：**设备应满足网络信号的硬件要求。

**安卓手机：**内存应在2G或以上，存储空间不小于100MB。设备应满足网络信号的硬件要求。

**苹果手机：**IPhone 3gs及其以后的所有苹果手机。

1. **服务端：**服务器内存应在2G或以上，硬盘空间不小于1GB，CPU运算能力应在Intel E5-Xeon Broadwell（v4）同等水平或之上。服务器应具有完全的互联网连接，带宽不低于1Mbps。

### 4.2. 支持软件

1. **用户端：**

**用户电脑：**windows 7/8/VISTA/10

**用户手机：**内核版本不低于4.0的Android操作系统或内核版本不低于6.1的IOS操作系统，完全的互联网连接。

1. **服务端：**阿里云服务器，Java Runtime Environment，MySQL数据库

### 4.3. 接口

微信小程序端的语音输入以来科大讯飞的语音输入接口，可直接调用此接口。

### 4.4. 控制

用户端直接使用浏览器启动软件，由浏览器发送请求以实现各类功能。

服务端程序部署后24h不间断运行，持续响应用户的请求。

## 5. 需求跟踪