**预约挂号业务管理信息系统**

**详细设计说明书**

**张腾飞、王赟鹏、杨润泽组**

**2017年11月**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档修改情况记录 | | | | |
| 版本 | 修改状态 | 修改日期 | 修改摘要 | 修改人 |
| 1.0 |  | 2017-10-09 | 创建 | 管艳华 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

[1 引言 1](#_Toc22051)

[1.1 目的 1](#_Toc3638)

[1.2 参考资料 1](#_Toc24949)

[1.3 名称解释 1](#_Toc9151)

[1.4 关键字 1](#_Toc23937)

[2 功能概述 1](#_Toc20555)

[2.1 系统管理员 1](#_Toc29163)

[2.1.1 医院管理 2](#_Toc20610)

# **引言**

## 编写目的

此设计说明书是预约挂号开发的主要依据，描述程序系统结构，并进行程序设计，为开发人员提供工作基准文件，并对后续阶段的工作起指导作用。

预期读者为设该系统的系统分析员、程序员、测试员、实施人员等。

## 参考资料

## 名称解释

|  |  |
| --- | --- |
| **名称** | **解释** |
| 预约挂号 | 网上预约专家 |
|  |  |

## 关键字

高级搜索，擅长诊治，提前预约时间，门诊日期，门诊时间，约满，停诊，赴约，失约。

# 总体设计

根据功能对系统进行划分，可分为预约挂号子系统、留言子系统、医院专家管理子系统、医院管理子系统以及系统管理子系统。

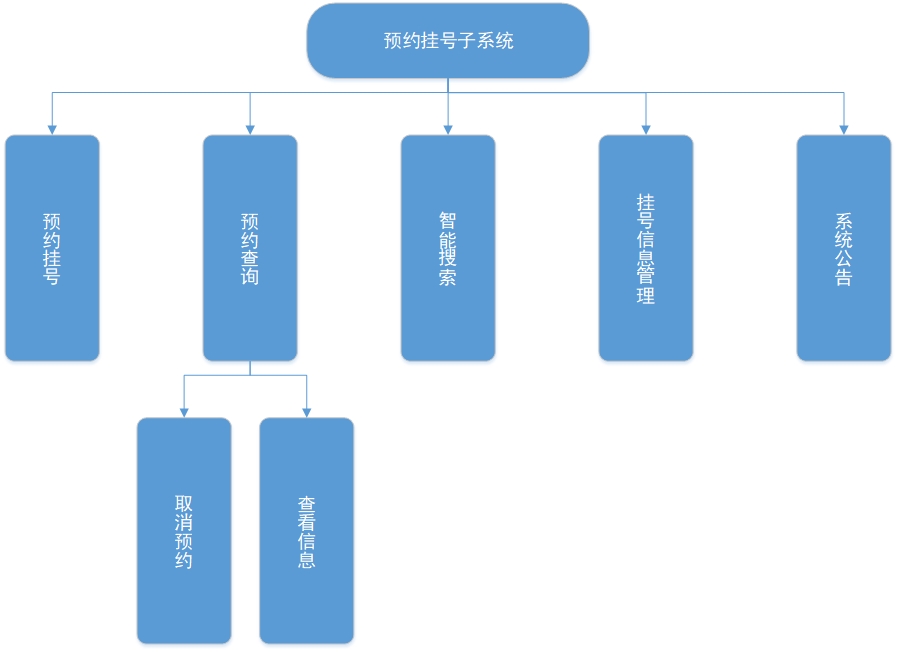
## 预约挂号子系统

负责对预约信息进行管理，主要内容有挂号信息管理、预约挂号、预约查询、智能搜索以及系统公告。

### **需求概述**

1. 预约挂号：就医患者可通过预约挂号平台向指定的专家进行预约挂号。预约挂号平台按科室分类，再细分专家。就医患者在进行预约挂号之前必须先确保自己的信息是否完善，再选择专家并且确定日期，实现网上预约挂号。
2. 预约查询：查看自己的预约信息，是否预约成功，是否过期，预约具体信息等。如果在预约日期前有其他特殊情况不能按时去医院就诊，可以登录系统查看自己的网上挂号预约情况，并提前取消预约。取消预约只能在预约日期之前进行，过期没有就诊的不能取消，并且用户查询不到过期或已就诊的预约信息。取消预约操作，需要填写就诊的姓名、身份证号以及预约识别码信息，信息填写完毕后执行提交操作完成取消预约操作。
3. 智能搜索：对于不清楚自己应该挂哪种科室的用户，可使用系统提供基于医疗行业专业术语描述的智能搜索，用户通过输入病症的一些关键词，系统自动给出推荐挂号科室。用户在搜索栏里输入自己的病情，点击搜索，则会显示相应病情的科室列表，点击符合自己需要的科室可浏览科室信息以及专家等。
4. 挂号信息管理：医院专家可通过预约挂号子系统对就医患者的挂号状态进行修改：若患者未能按期就诊，将其预约信息修改为过期；若患者按期就诊，将其预约信息修改为已就诊。

### **软件结构**



### **具体描述**

### **2.1.4 流程图**

### **2.1.5 数据库设计**

### 预约表

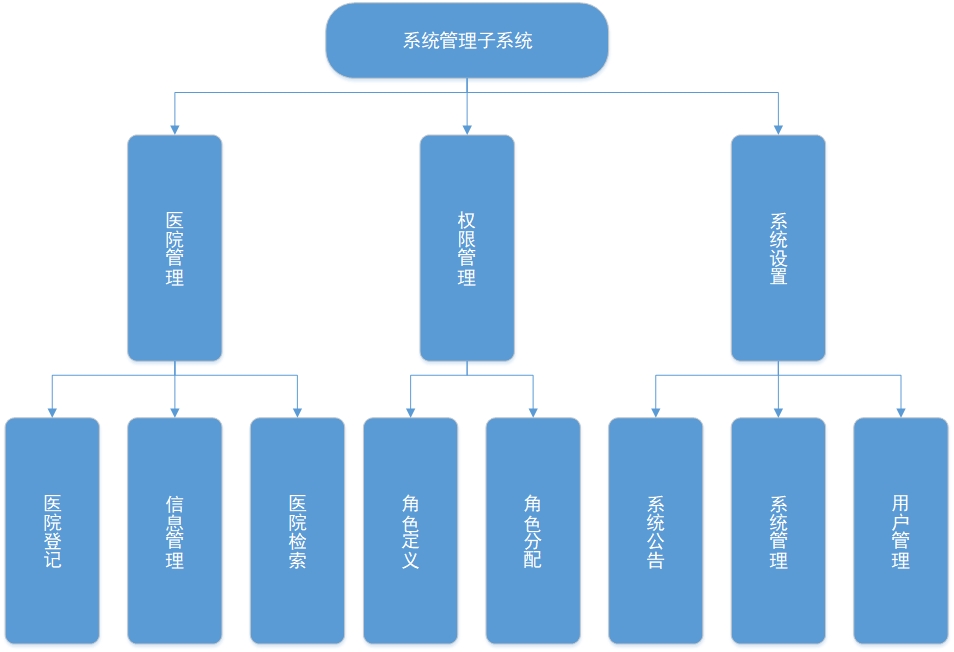
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据格式 | 是否为空 | 约束 | 说明 | 中文解释 |
| ap\_code | varchar(10) | 否 | 主键 | 唯一标识预约记录 | 预约号 |
| ap\_account | varchar(12) | 否 | 外键 | 用户表账号外键 | 用户账号 |
| ap\_name | varchar(20) | 否 |  |  | 用户姓名 |
| ap\_date | char(19) | 否 |  | 预约就诊的日期 | 预约日期 |
| ap\_operDate | char(19) | 否 |  | 预约操作的日期 | 操作日期 |
| ap\_lastModified | char(19) | 是 |  | 状态最后修改日期 | 修改日期 |
| ap\_status | char(1) | 否 | 默认1 | 1:已预约 2:已就诊 3:已过期 | 用户状态 |

### **2.2 系统管理子系统**

**2.2.1 需求概述**

1. 医院管理：设置相关医院信息，完成后就可以在前台显示出来,供患者预约。
2. 权限管理：管理员能够自定义角色，并将这些角色分配给每一个已经登记到预约平台的医院。拥有角色的医院专门负责人能够在管理系统执行该角色权限范围内的操作，例如修改医院信息等。

### **2.2.2 软件结构**



### **2.2.3具体描述**

1. 医院登记

功能：往预约挂号平台内登记新的医院信息。

输入：医院代码、医院名称、搜索显示优先级。

输出：操作执行的结果。

1. 信息管理

功能：对已有的医院进行管理：修改医院的基础信息或从平台移除该医院。

输入：医院代码、医院名称、搜索显示优先级。

输出：操作执行的结果。

1. 医院检索

功能：根据检索条件查询指定医院的相关基础信息。

输入：医院代码或医院名称。

输出：医院代码、医院名称、搜索显示优先级、相关操作等。

1. 角色定义

功能：定义角色，并为每个角色分配相应的权限。

输入：角色名称、权限。

输出：操作执行的结果。

1. 角色分配

功能：为每个管理用户分配相应的角色。

1. 系统公告

功能：修改系统公告

输入：系统公告内容。

输出：操作执行的结果。

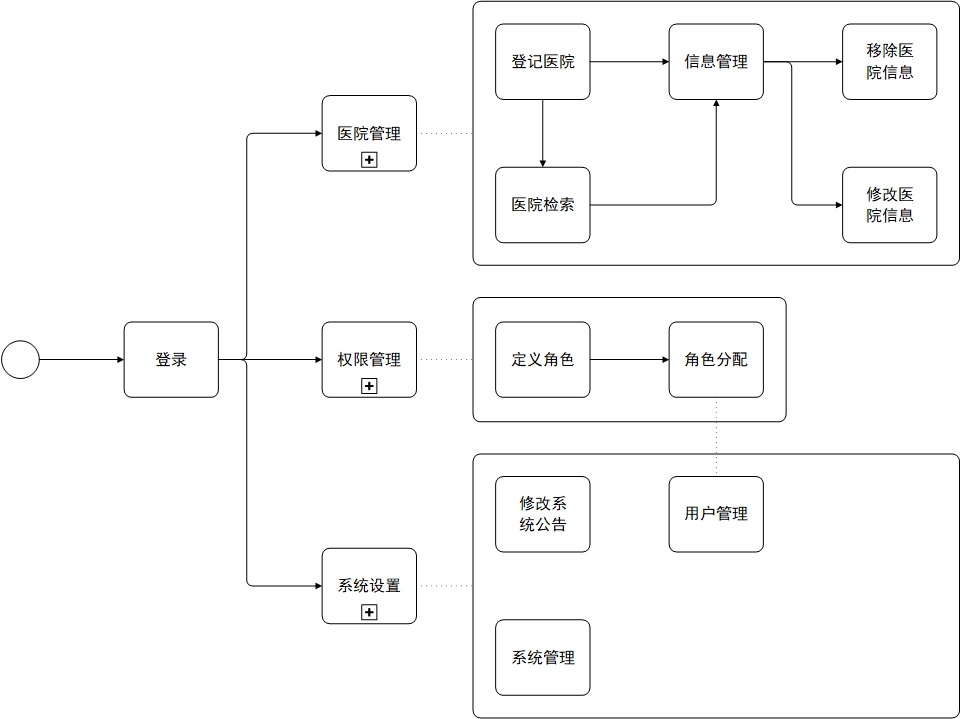
1. 系统管理

功能：对系统基础配置项进行修改等操作。

输入：

输出：

### **2.2.4流程图**



**2.2.5数据库设计**

**E-R图：**

**数据字典：**

医院表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据格式 | 是否为空 | 约束 | 说明 | 中文解释 |
| hos\_name | varchar(40) | 否 |  |  | 医院名称 |
| hos\_code | varchar(10) | 否 | 主键 |  | 医院代码 |
| hos\_grade | char(1) | 是 |  |  | 医院等级 |
| hos\_area | varchar(2) | 是 |  |  | 所在区域 |
| hos\_addr | varchar(80) | 是 |  |  | 详细地址 |
| hos\_phone | varchar(12) | 是 |  |  | 电话热线 |
| hos\_brief | varchar(4096) | 是 |  |  | 医院介绍 |

### 专家表

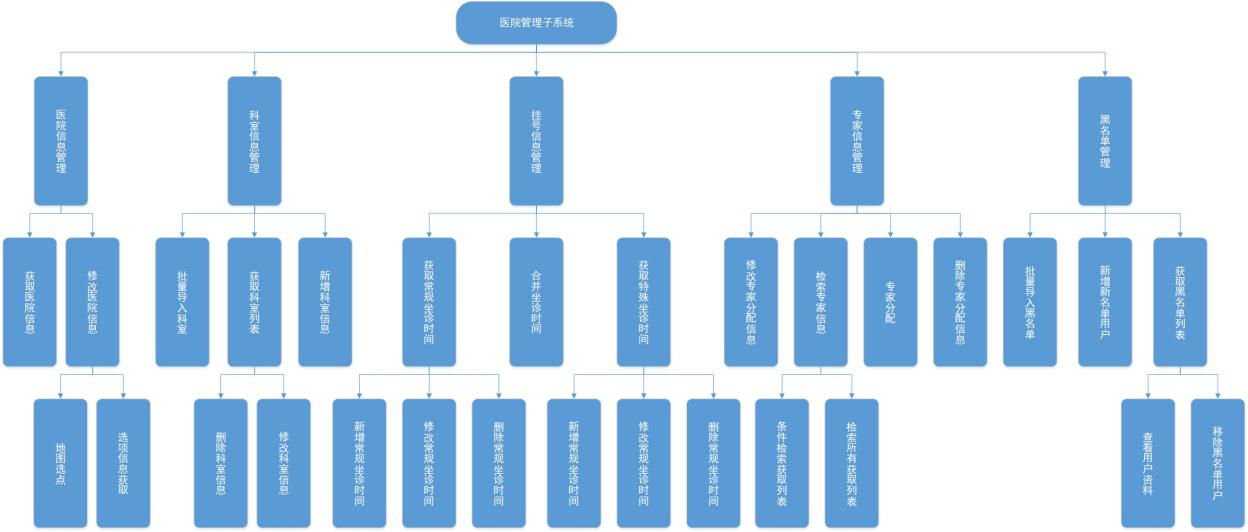
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据格式 | 是否为空 | 约束 | 说明 | 中文解释 |
| exp\_code | varchar(10) | 否 | 主键 |  | 专家号 |
| exp\_name | varchar(40) | 否 |  |  | 专家姓名 |
| exp\_account | varchar(12) | 否 | 外键 | 用户表账号外键 | 专家账号 |
| exp\_position | varchar(20) | 是 |  |  | 专家职位 |
| exp\_gender | varchar(2) | 是 |  |  | 专家性别 |
| exp\_photo | BLOB | 是 |  |  | 专家照片 |
| exp\_intro | varchar(4096) | 是 |  |  | 专家介绍 |
| exp\_depart | varchar(10) | 是 | 外键 | 科室表代码外键 | 所属科室 |
| exp\_hospital | varchar(10) | 是 | 外键 | 医院表代码外键 | 所属医院 |

### **2.3 医院管理子系统**

**2.3.1需求概述**

1. 医院信息管理：医院管理员可以设置医院相关的信息（比如医院介绍、荣誉、联系方式、坐车线路等信息）。
2. 科室管理：管理本医院科室信息。
3. 专家管理：根据不同科室，管理相应科室的专家信息。
4. 挂号设置管理：设置每个专家的坐诊时间和挂号的个数、挂号费。
5. 黑名单管理：对半年内预约挂号五次不来的，视作恶意用户，取消其半年内享受网络预约挂号的权利，自动加入黑名单，在黑名单内的用户，半年内禁止使用该系统进行预约挂号。对于网上预约和现场取号时身份不一致的情况，取消预约挂号者半年内享受网络预约挂号的权利。系统配置加入黑名单的爽约次数或取消次数和禁止使用此系统进行预约挂号的时间。表现良好，经医院审批，可提前从黑名单除名。

### **2.3.2软件结构**



### **2.3.3具体描述**

1. 获取医院信息：

功能：打开医院信息页面时，自动从数据库中获取当前医院的信息，并将其呈现在页面上。注意，系统管理员能够查看所有医院的信息，但不具有修改权限。

输入：医院代码。

输出：医院名称、联系热线、医院网站、医院等级、医院地址、乘车路线、医院介绍、百度地图位置点。

1. 修改医院信息：

功能：医院管理员拥有修改当前医院信息的权限，除了医院名称无法修改外，其他相关的信息都能够进行修改。

输入：联系热线、医院网站、医院等级、乘车路线、医院介绍。

输出：响应代码、响应消息。

A 地图选点：

功能：医院管理员能够在地图上找到医院所在位置，并将该地理位置作为医院地址。

输入：百度地图坐标点。

输出：响应代码、响应消息。

B 选项信息获取：

功能：医院等级这一类通过下拉框选取的项目，需要为其新增一个方法来统一实现。

输入：需要获取项目组成的数组。

输出：响应代码、响应消息、输入项目的内容。

1. 获取科室信息：

功能：打开科室信息页面是，自动从数据库中调取当前医院的所有科室信息，并将其呈现在页面上。

输入：医院代码。

输出：科室名称、科室介绍。

1. 新增科室信息：

功能：医院管理员通过点击“新增科室”按钮，在弹出框中输入科室名称以及科室介绍来实现新增科室功能。根据响应代码进行判断，若执行成功，则将输入的内容更新到科室表中（页面显示）。

输入：科室名称、科室介绍。

输出：响应代码、响应消息。

1. 批量导入科室：

功能：当新增的科室过多时，管理员可以通过上传组织清晰的EXCEL文件来实现对科室信息新增的批处理。在执行成功后，调用获取科室信息方法，重新在页面上呈现最新的科室列表。

输入：EXCEL文件。

输出：响应代码、响应消息。

1. 获取科室列表：

功能：进入科室管理页面时，自动从服务器获取当前医院的所有科室信息。

输入：医院代码。

输出：科室名称、科室介绍。

1. 修改科室信息：

功能：医院管理员能够对已有的科室信息进行修改。

输入：科室名称、科室介绍。

输出：响应代码、响应消息。

1. 删除科室信息：

功能：医院管理员对已有科室信息进行删除。

输入：科室名称。

输出：响应代码、响应消息。

1. 挂号信息管理：请参考2.4医院专家管理子系统坐诊时间管理。
2. 检索专家分配信息：

功能：通过指定条件来实现精细检索或检索所有信息。

A 条件检索获取列表：

功能：可选择专家姓名或专家号作为检索条件。

输入：专家姓名（专家号）、医院代码。

输出：专家姓名、所属科室。

B 检索所有获取列表：

功能：不需要输入任何检索条件，直接获取当前医院的所有专家信息。

输入：医院代码。

输出：专家姓名、所属科室。

1. 专家分配：

功能：为专家分配所属的科室。

输入：专家号、专家姓名、所属科室。

输出：响应代码、响应消息。

1. 修改专家分配信息：

功能：从科室信息中修改指定专家信息。

输出：专家号、专家姓名、所属科室。

输出：响应代码、响应信息。

1. 删除专家分配信息：

功能：从科室信息中删除指定专家信息，例如该专家离职或调职。

输入：专家号。

输出：响应代码、响应消息。

1. 新增黑名单用户：

功能：医院管理员能够手动将一些用户加入到黑名单中。

输入：用户账号、加入原因。

输出：响应代码、响应消息。

1. 批量导入黑名单：

功能：上传Excel文件，文件内包含用户账号以及加入原因。

输入：Excel文件。

输出：响应代码、响应消息。

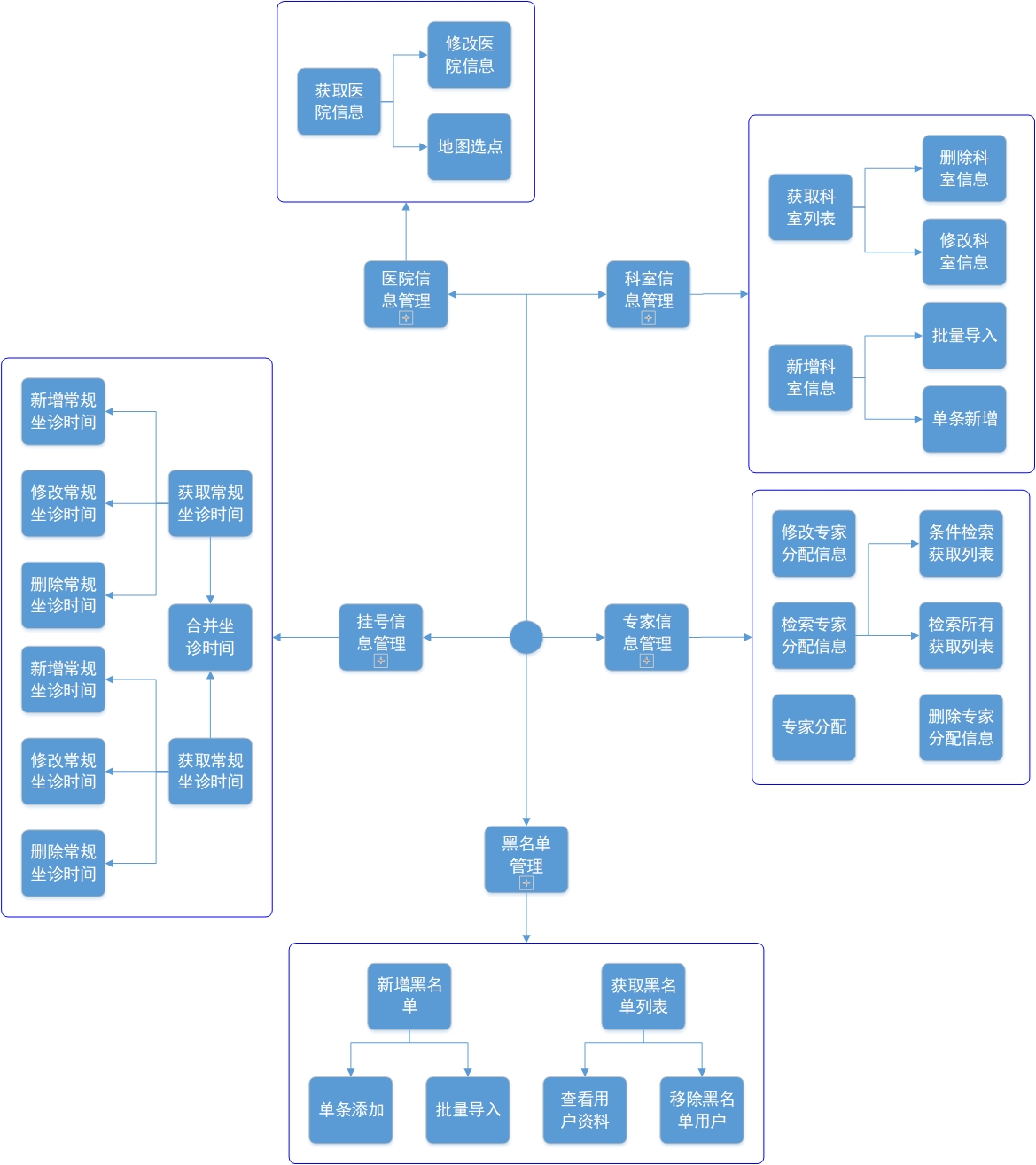
1. 获取黑名单列表：

功能：通过检索条件（用户账号或用户姓名）或直接检索所有的方式获取黑名单列表。

输入：用户账号（用户姓名）。

输出：用户账号、用户姓名、加入时间、加入原因。

### **2.3.4流程图**



### **2.3.5数据库设计**

**ER图：**

**数据字典：**

科室表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 约束 | 说明 | 中文解释 |
| oid | varchar(10) | 否 | 主键 | 唯一标识科室 | 科室号 |
| office\_name | varchar(10) | 否 |  |  | 科室名称 |
| office\_intro | varchar(1024) | 是 |  | 最多512中文 | 科室介绍 |
| office\_hos | varchar(10) | 否 | 医院表外键 | 对应医院代码 | 所属医院 |

黑名单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 约束 | 说明 | 中文解释 |
| bid | varchar(10) | 否 | 主键 | 唯一标识名单 | 黑名单号 |
| account | varchar(12) | 否 |  | 用户账号 | 用户账号 |
| name | varchar(20) | 否 |  |  | 用户姓名 |
| date | char(10) | 否 |  | YYYY-MM-dd | 加入时间 |
| reason | varchar(256) | 是 |  | 加入原因 | 加入原因 |

### **2.3.6程序界面**

详见：

1. **后台管理系统原型图—>医院平台—>医院信息**
2. **后台管理系统原型图—>医院平台—>科室管理**
3. **后台管理系统原型图—>医院平台—>专家管理**
4. **后台管理系统原型图—>医院平台—>挂号设置**
5. **后台管理系统原型图—>医院平台—>黑名单管理**

### **2.3.7类设计**

**A. 类图：**

**B. 控制器重要方法：**

**C. 模型类重要成员：**

(1)

**D. 程序清单：**

### **2.4 医院专家管理子系统**

### **2.4.1需求概述**

1. 个人信息管理：完善自己的个人信息，姓名，照片，性别，年龄，级别，擅长领域，便于用户挂号时了解和选择专家。
2. 坐诊时间管理：设置自己的坐诊时间表，便于用户挂号参考。同时医生可以设置停诊时间，一旦医生设置了停诊时间，应立即取消该医生的所有预约号，通知短信通知患者本人。

### **2.4.2软件结构**

### E:\project\NIIT\大四上\qdyypt\详细设计\医院专家管理子系统\医院专家管理子系统软件结构.jpg医院专家管理子系统软件结构

### **2.4.3具体描述**

1. **照片上传**：

功能：上传个人照片。

输入：图片。

输出：图片上传操作结果。

1. **个人信息获取**：

功能：获取个人信息。

输入：用户账号。

输出：专家号、专家姓名、专家账号、专家职位、专家性别、专家照片、专家介绍、所属科室、所属医院

1. **个人信息修改**：

功能：对专家个人信息（姓名、职位、个人介绍等）进行修改。

输入：姓名、性别、职位、介绍等。

输出：个人信息修改操作结果。

1. **预约用户获取**：

功能：根据检索条件获取预约用户部分可见信息以及预约状态。

输入：预约号、用户名称或预约日期。

输出：预约号、用户名称、预约日期、用户状态。

1. **预约状态修改**：

功能：修改指定用户的预约状态，预约状态共三种，分别为已预约、已就诊以及已过期。

输入：预约号、用户状态。

输出：预约状态修改操作执行的结果。

1. **状态批量修改：**

功能：批量修改用户的预约状态。

输入：预约日期、预约号、预约状态。

输出：批量修改操作执行的结果。

1. **获取常规坐诊时间：**

功能：专家进入坐诊时间管理页面时就已经从服务器数据库中将常规坐诊时间信息全部读取，之后对常规坐诊时间的处理全部基于该操作获取的数据基础上。注意，该部分数据不会独立呈现在页面，需要和之后获取的特殊坐诊时间合并后呈现在坐诊时间表中。

输入：专家号。

输出：坐诊号、坐诊人数、坐诊费用、已预约人数、能否修改、能否预约。

1. **设置常规坐诊时间：**

功能：位于页面正上方的按钮，点击后弹出“设置常规坐诊时间”面板，在该面板中可以对坐诊数据进行新增、修改以及删除操作。

**A 新增常规坐诊时间**：

输入：专家号、坐诊号、坐诊人数、坐诊费用。

输出：响应代码、响应信息。

**B 修改常规坐诊时间**：

输入：专家号、坐诊号、坐诊人数、坐诊费用。

输出：响应代码、响应信息。

**C 删除常规坐诊时间**：

输入：专家号、坐诊号。

输出：响应代码、响应信息。

1. **获取特殊坐诊时间**

功能：从数据库将 sdiagnose 数据表的相关内容读取并呈现在特殊坐诊时间表中。通过表格的操作项，能够对特殊坐诊记录进行修改以及删除操作。

输入：专家号。

输出：坐诊人数、坐诊费用、坐诊时间、已预约人数、能否预约。

**A 修改特殊坐诊时间**：

输入：专家号、坐诊号、坐诊人数、坐诊费用、坐诊时间。

输出：响应代码、响应信息。

**B 删除特殊坐诊时间**：

输入：专家号、坐诊号。

输出：响应代码、响应信息。

1. **设置特殊坐诊时间**

功能：因某种原因需要医院专家额外进行坐诊，例如某段期间流感肆虐，医院增加坐诊的人手。同时也能起到代班的作用，原坐诊专家因有事发出停诊通知，告知就医患者停诊期间由另一名专家坐诊，此时可以让另一名专家设置特殊坐诊时间，从而实现代班功能。

**A 新增特殊坐诊时间**：

输入：坐诊时间、预约人数、预约费用。

输出：响应代码、响应信息。

1. **合并坐诊时间：**

功能：将常规坐诊时间和特殊坐诊时间合并。特殊坐诊时间优先级高于常规坐诊时间，如果两者在相同的时间，以特殊坐诊时间的信息为准。

输入：常规坐诊时间记录、特殊坐诊时间记录。

输出：合并后的坐诊号、坐诊人数、坐诊费用、已预约人数、能否修改、能否预约。

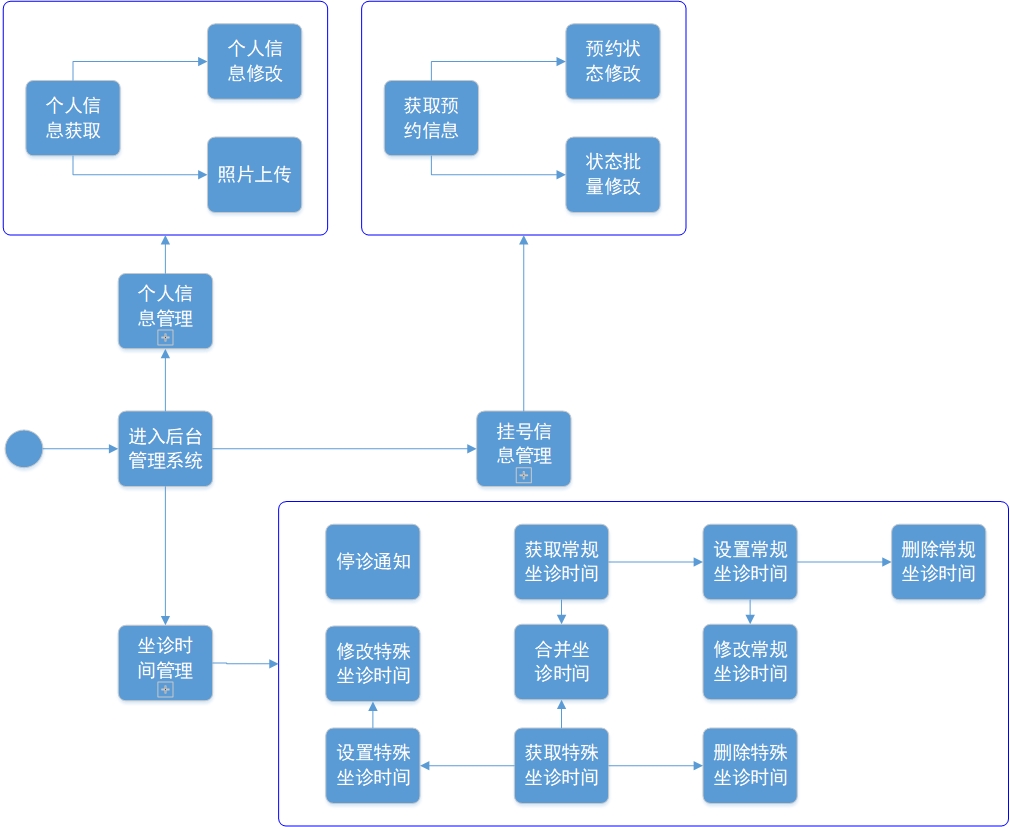
1. **停诊通知：**

功能：医院专家因某事而未能进行坐诊，向在停诊期间已预约的患者发出通知，并在平台上将自己的预约状态修改为停诊。

输入：开始日期、结束日期、专家号、专家姓名、停诊原因。

输出：停诊通知设置操作执行的结果。

### **2.4.4流程图**



### **2.4.5数据库设计**

**ER 图：**

**数据字典：**

常规坐诊表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据格式 | 空 | 约束 | 说明 | 中文解释 |
| dg\_code | varchar(2) | 否 | 组合主键 | 唯一标识坐诊时间 | 坐诊时间代码 |
| dg\_number | int | 否 |  |  | 坐诊人数 |
| dg\_cost | int | 否 |  |  | 坐诊费用 |
| dg\_apNumber | int | 是 | 默认0 | 初始为0 | 已预约人数 |
| dg\_isModified | char(1) | 是 | 默认1 | 0:不能修改 1:修改 | 能否修改 |
| dg\_belongTo | varchar(10) | 否 | 组合主键+外键 | 专家表外键 | 所属专家 |
| dg\_isAppointd | char(1) | 是 | 默认1 | 0:不能预约1:预约 | 能否预约 |

特殊坐诊表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据格式 | 空 | 约束 | 说明 | 中文解释 |
| dg\_code | varchar(2) | 否 | 组合主键 | 唯一标识坐诊时间 | 坐诊时间代码 |
| dg\_number | int | 否 |  |  | 坐诊人数 |
| dg\_cost | int | 否 |  |  | 坐诊费用 |
| dg\_apNumber | int | 是 | 默认0 | 初始为0 | 已预约人数 |
| dg\_isModified | char(1) | 是 | 默认1 | 0:不能修改 1:修改 | 能否修改 |
| dg\_date | char(19) | 否 |  | 用以判断这一周是否存在特殊坐诊 | 特殊坐诊日期 |
| dg\_belongTo | varchar(10) | 否 | 组合主键+外键 | 专家表外键 | 所属专家 |
| dg\_isAppointd | char(1) | 是 | 默认1 | 0:不能预约1:预约 | 能否预约 |

### **2.4.6程序界面**

详见：

1. **后台管理系统原型图—>专家平台—>坐诊时间管理**
2. **后台管理系统原型图—>专家平台—>挂号信息管理**
3. **后台管理系统原型图—>个人平台—>个人信息**

### **2.4.7类设计**

（1）类图：

（2）控制器重要方法（伪代码）：

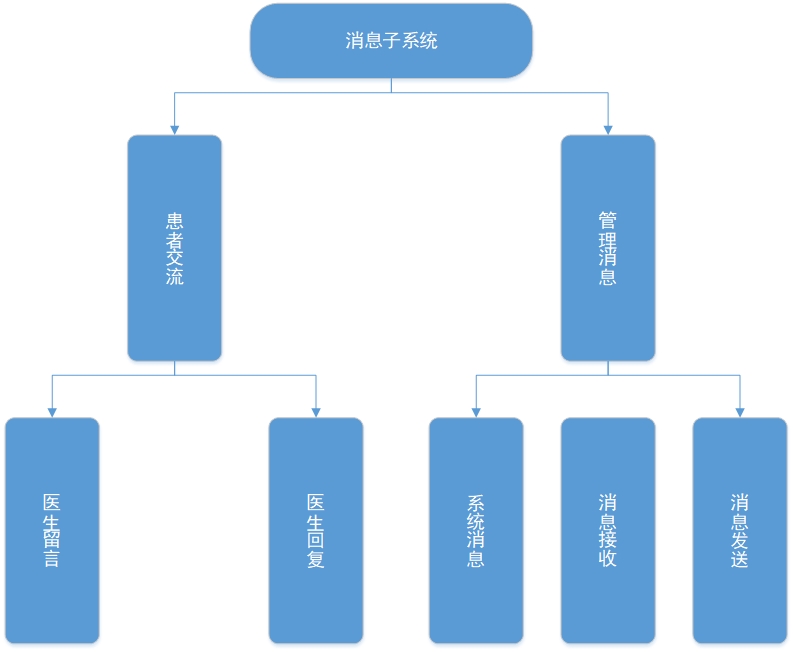
*/\*\*  
 \* 获取坐诊时间表：包括常规坐诊时间以及特殊坐诊时间  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** getDiagnose(Request $req = **null**) {  
 $model = **new** dgModel;  
 $modelSpecial = **new** dgsModel;  
 $base = **new** Base;  
 $exp\_code = $req->post(**"code"**);  
  
 *// 根据专家号获取该专家的坐诊时间表（常规坐诊以及特殊坐诊）* $data = $model::*all*(**function**($query) **use**($exp\_code) {  
 $query->where(**"dg\_belongTo"**, $exp\_code);  
 });  
 $dataSpecial = $modelSpecial::*all*(**function**($query) **use**($exp\_code) {  
 $query->where(**"dgs\_belongTo"**, $exp\_code);  
 });  
 $list = *array\_push*($data, $dataSpecial);  
 **return** $base->turnToJSONResp(**"1"**, **""**, $list);  
}  
  
*/\*\*  
 \* 设置坐诊时间  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** addDiagnose(Request $req = **null**) {  
 *// type 用以区分当前操作的对象是常规坐诊时间还是特殊坐诊时间* $type = $req->post(**"type"**);  
 $base = **new** Base;  
  
 **if**($type === **"0"**) {  
 $model = **new** dgModel;  
 } **else if**($type === **"1"**) {  
 $model = **new** dgsModel;  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"type 类型有误！"**, **null**);  
 }  
 $model->**dg\_code** = $req->post(**"code"**);  
 $model->**dg\_number** = $req->post(**"number"**);  
 $model->**dg\_cost** = $req->post(**"cost"**);  
 $model->**dg\_apNumber** = $req->post(**"apNumber"**);  
 $model->**dg\_belongTo** = $req->post(**"expCode"**);  
 *// 特殊坐诊时间表特有字段* **if**($type === **"1"**) {  
 $model->**dg\_date** = $req->post(**"date"**);  
 }  
 **if**($model->save()) {  
 **return** $base->turnToJSONResp(**"1"**, **"设置坐诊时间成功！"**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"设置坐诊时间失败！"**, **null**);  
 }  
}  
  
*/\*\*  
 \* 修改坐诊时间  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** modifyDiagnose(Request $req = **null**) {  
 *// type 用以区分当前操作的对象是常规坐诊时间还是特殊坐诊时间* $type = $req->post(**"type"**);  
 $code = $req->post(**"code"**);  
 $base = **new** Base;  
  
 **if**($type === **"0"**) {  
 $model = **new** dgModel;  
 } **else if**($type === **"1"**) {  
 $model = **new** dgsModel;  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"type 类型有误！"**, **null**);  
 }  
 *// 根据坐诊号检索相应信息* $data = $model::*get*($code);  
 $data->**dg\_code** = $req->post(**"code"**);  
 $data->**dg\_number** = $req->post(**"number"**);  
 $data->**dg\_cost** = $req->post(**"cost"**);  
 $data->**dg\_apNumber** = $req->post(**"apNumber"**);  
 $data->**dg\_belongTo** = $req->post(**"expCode"**);  
 *// 特殊坐诊时间表特有字段* **if**($type === **"1"**) {  
 $data->**dg\_date** = $req->post(**"date"**);  
 }  
 **if**($data->save()) {  
 **return** $base->turnToJSONResp(**"1"**, **"修改坐诊时间成功！"**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"修改坐诊时间失败！"**, **null**);  
 }  
}  
  
*/\*\*  
 \* 删除坐诊时间  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** deleteDiagnose(Request $req = **null**) {  
 *// type 用以区分当前操作的对象是常规坐诊时间还是特殊坐诊时间* $type = $req->post(**"type"**);  
 $code = $req->post(**"code"**);  
 $base = **new** Base;  
  
 **if**($type === **"0"**) {  
 $model = **new** dgModel;  
 } **else if**($type === **"1"**) {  
 $model = **new** dgsModel;  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"type 类型有误！"**, **null**);  
 }  
 $data = $model::*get*($code);  
 **if**($data->delete()) {  
 **return** $base->turnToJSONResp(**"1"**, **"删除坐诊时间成功！"**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"1"**, **"删除坐诊时间成功！"**, **null**);  
 }  
  
}  
  
*/\*\*  
 \* 停诊通知  
 \** ***@param*** *Request|null $req  
 \*/***public function** stopDiagnose(Request $req = **null**) {  
 *// 将专家坐诊状态修改为停诊：将该部分功能移至个人信息管理  
 // 将停诊期间内的坐诊时间修改为不可预约：这部分的修改需要通过事务来实现批处理* $model = **new** dgsModel;  
 $startDate = **"2017-10-11"**; *// 测试使用* $endDate = **"2017-10-13"**; *// 测试使用* $result = $model::*all*(**function**($query) **use** ($startDate, $endDate) {  
 $query->where(**"dg\_date"**, **">="**, $startDate)  
 ->where(**"dg\_date"**, **"<="**, $endDate);  
 });  
 **foreach**($result **as** $item) {  
 $item->**dg\_isAppointed** = **"0"**;  
 $item->save();  
 }  
 *// 根据时间范围，获取停诊期间内患者的用户账号等信息* $userList = getUserList();  
 *// 向这些用户发送信件：同样需要通过事务来实现批处理*}

### **2.5留言子系统**

### **2.5.1需求概述**

医生留言：提供医生和患者的互动交流平台。每个医生有自己专用账号和交流页面，患者在查询浏览了该医生的基本信息后，如果有问题需要咨询该医生，可以登录平台后在该医生的页面留言提问，医生登录后可以看到自己的留言页面上患者提出的问题，并做相应解答。

### **2.5.2软件结构**



### **2.5.3具体描述**

（1）医生留言：

功能：就医患者可以给相应的医生留言，以便获取对病情更进一步的了解。

输入：留言内容、发件人账号、发件人姓名、发送时间、收件人账号、收件人姓名。

输出：留言操作执行的结果

（2）医生回复：

功能：医生在后台管理系统中看到就医患者的留言后能够及时地对患者的疑问进行解答。

输入：回复内容、发件人账号、发件人姓名、发送时间、收件人账号、收件人姓名。

输出：回复操作执行的结果。

（3）系统消息：

功能：系统管理员能够通过后台管理系统向指定用户发送系统消息。系统消息不能被回复。

输入：收件人账号、收件人姓名、发送时间、发件人账号、发件人姓名、消息内容、消息标题。

输出：消息发送操作执行的结果。

（4）消息接收：

功能：使用该平台的所有人（系统管理员除外）都能够接收消息。

输入：无

输出：发件人账号、发件人姓名、发送时间、消息标题、消息内容、收件人账号、收件人姓名。

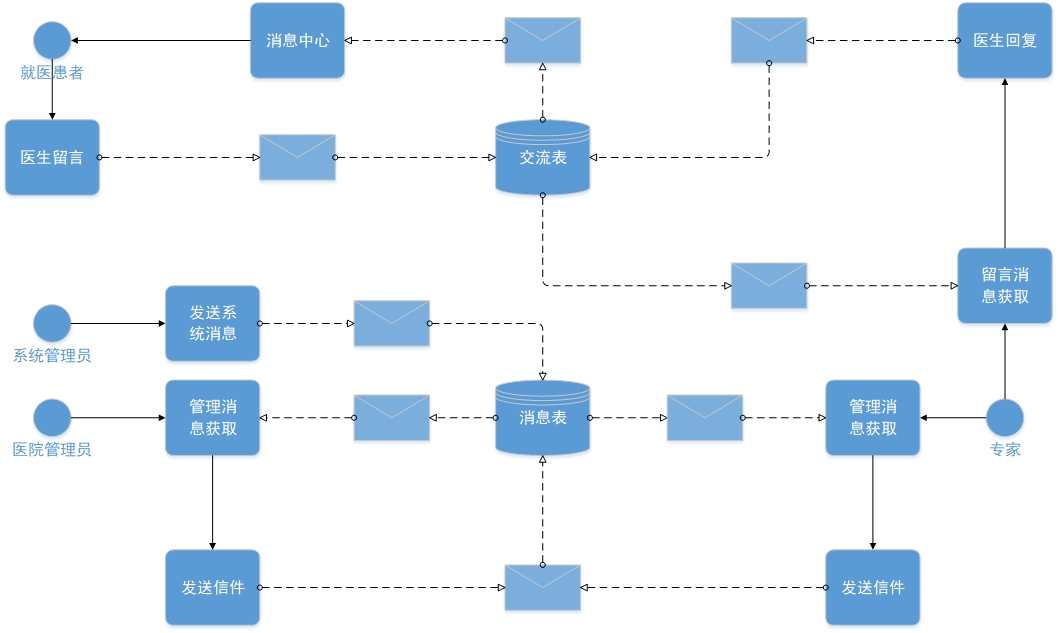
（5）消息发送：

功能：使用该平台的所有人都能够发送消息，但不同角色能够发送消息的对象范围有所不同。

输入：收件人账号、收件人姓名、发送时间、发件人账号、发件人姓名、消息内容、消息标题。

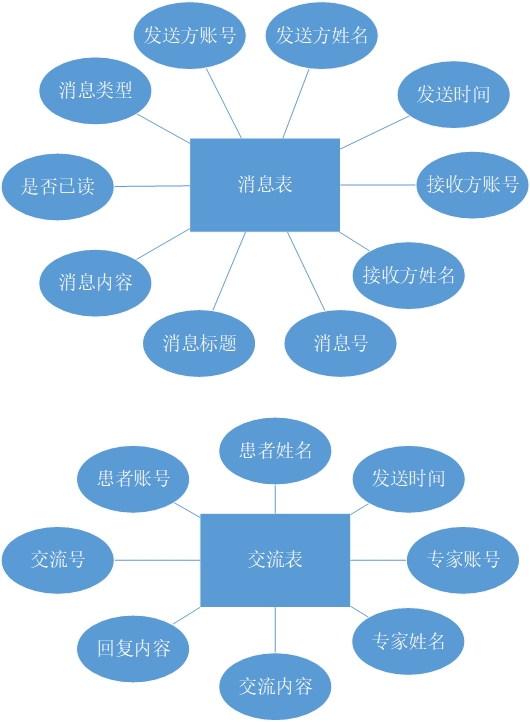
输出：消息发送操作执行的结果。

### **2.5.4流程图**



### **2.5.5数据库设计**

ER图：



数据字典：

交流表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 约束 | 说明 | 中文解释 |
| cid | varchar(20) | 否 | 主键 | 唯一标识消息 | 交流号 |
| sender\_account | varchar(12) | 否 |  | 发送者账号 | 患者账号 |
| sender\_name | varchar(10) | 否 |  | 发送者姓名 | 患者姓名 |
| sendTime | char(19) | 否 |  | YYYY-MM-dd HH:mm:ss | 发送时间 |
| receiver\_account | varchar(12) | 否 |  | 接收者账号 | 专家账号 |
| receiver\_name | varchar(10) | 否 |  | 接收者姓名 | 专家姓名 |
| sender\_content | varchar(4096) | 否 |  | 发送者留言 | 交流内容 |
| Receiver\_content | varchar(4096) | 是 |  | 接收者回复 | 回复内容 |

消息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 为空 | 约束 | 说明 | 中文解释 |
| mid | varchar(20) | 否 | 主键 | 唯一标识消息 | 消息号 |
| sender\_account | varchar(12) | 否 |  |  | 发送方账号 |
| sender\_name | varchar(10) | 否 |  |  | 发送方姓名 |
| sendTime | char(19) | 否 |  | YYYY-MM-dd HH:mm:ss | 发送时间 |
| receiver\_account | varchar(12) | 否 |  |  | 接收方账号 |
| receiver\_name | varchar(10) | 否 |  |  | 接收方姓名 |
| title | varchar(100) | 否 |  |  | 消息标题 |
| content | varchar(4096) | 否 |  |  | 消息内容 |
| isRead | char(1) | 是 | 默认0 | 0:未读 1:已读 | 是否已读 |
| type | char(1) | 否 |  | 0:写信 1：收信 | 消息类型 |

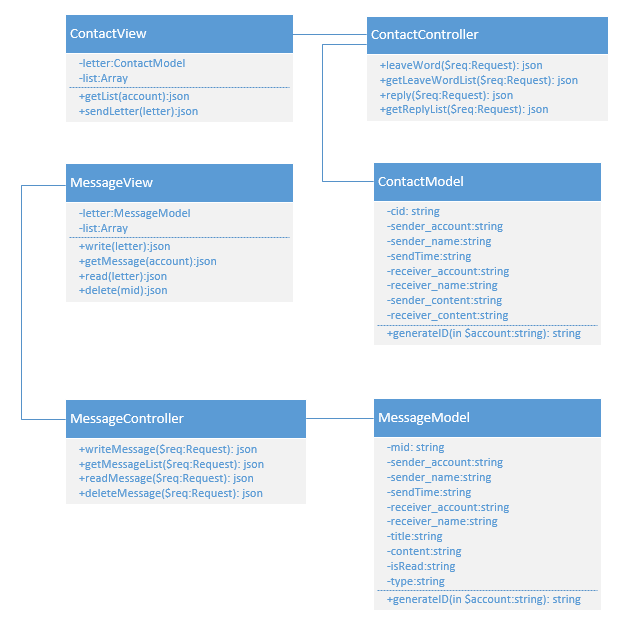
### **2.5.6程序界面：**

详见：

1. **后台管理系统原型图—>消息平台—>消息管理**
2. **后台管理系统原型图—>系统管理—>系统公告**
3. **后台管理系统原型图—>系统管理—>系统消息**

### **2.5.7类设计：**

A：类图



B：控制器重要方法列表：

1. ContactController：

*/\*\*  
 \* 就医患者获取专家回复信件列表  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** getReplyList(Request $req = **null**) {  
 $model = **new** ContactModel;  
 $base = **new** Base;  
  
 *// 从数据库中查询当前患者的留言信息* $list = $model::*all*(**function**($query) **use** ($req){  
 $query->where(**"sender\_account"**, $req->post(**"account"**));  
 });  
 *// 返回 JSON 格式数据* **return** $base->turnToJSONResp(**"1"**, **""**, $list);  
}  
  
*/\*\*  
 \* 就医患者向医生留言  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *array|string  
 \*/***public function** leaveWord(Request $req = **null**) {  
 $model = **new** ContactModel;  
 $base = **new** Base;  
  
 *// 根据患者账号自动生成 cid* $cid = $model->generateID(**'spirit'**);  
 *// 将前端传递的数据插入到 contact 表中* $model->**cid** = $cid;  
 $model->**sender\_account** = $req->post(**"senderAccount"**);  
 $model->**sender\_name** = $req->post(**"senderName"**);  
 $model->**sendTime** = *date*(**"Y-m-d h:i:s"**);  
 $model->**receiver\_account** = $req->post(**"receiverAccount"**);  
 $model->**receiver\_name** = $req->post(**"receiverName"**);  
 $model->**sender\_content** = $req->post(**"content"**);  
 **if**($model->save()) {  
 **return** $base->turnToJSONResp(**"1"**, **"留言成功！"**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"留言失败！"**, **null**);  
 }  
}  
  
*/\*\*  
 \* 获取当前专家交流信息  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** getLeaveWordList(Request $req = **null**) {  
 $model = **new** ContactModel;  
 $base = **new** Base;  
  
 *// 从 contact 表中查询当前专家收到患者的留言信息* $list = $model::*all*(**function**($query) **use**($req){  
 $query->where(**"receiver\_account"**, $req->post(**"account"**));  
 });  
 **return** $base->turnToJSONResp(**"1"**, **""**, $list);  
}  
  
*/\*\*  
 \* 专家收到就医患者的留言后进行回复  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *array|string  
 \*/***public function** reply(Request $req = **null**) {  
 $model = **new** ContactModel;  
 $base = **new** Base;  
  
 *// 根据 cid 从 contact 表中找出对应的留言记录，然后将回复更新到该记录中* $data = $model::*get*($req->post(**"cid"**));  
 $data->**receiver\_content** = $req->post(**"content"**);  
 **if**($model->save()) {  
 **return** $base->turnToJSONResp(**"1"**, **"回复成功！"**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"回复失败！"**, **null**);  
 }  
}

1. MessageController：

*/\*\*  
 \* 消息发送  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** writeMessage(Request $req = **null**) {  
 $model = **new** MessageModel;  
 $base = **new** Base;  
  
 $model->**mid** = $model->generateID(**"spirit"**);  
 $model->**sender\_account** = $req->post(**"senderAccount"**);  
 $model->**sender\_name** = $req->post(**"senderName"**);  
 $model->**sendTime** = *date*(**"Y-m-d h:i:s"**);  
 $model->**receiver\_account** = $req->post(**"receiverAccount"**);  
 $model->**receiver\_name** = $req->post(**"receiverName"**);  
 $model->**title** = $req->post(**"title"**);  
 $model->**content** = $req->post(**"content"**);  
 **if**($model->save()) {  
 **return** $base->turnToJSONResp(**"1"**, **""**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"发送消息失败！"**, **null**);  
 }  
}  
  
*/\*\*  
 \* 获取消息列表  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** getMessageList(Request $req = **null**) {  
 $model = **new** MessageModel;  
 $base = **new** Base;  
  
 *// 获取发送消息列表* **if**($req->post(**"type"**) === **"0"**) {  
 $list = $model::*all*(**function**($query) **use**($req){  
 $query->where(**"sender\_account"**, $req->post(**"account"**));  
 });  
 }  
 *// 获取接收消息列表* **else** {  
 $list = $model::*all*(**function**($query) **use**($req){  
 $query->where(**"receiver\_account"**, $req->post(**"account"**));  
 });  
 }  
 **return** $base->turnToJSONResp(**"1"**, **""**, $list);  
}  
  
*/\*\*  
 \* 修改信件状态为已读  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** readMessage(Request $req = **null**) {  
 $model = **new** MessageModel;  
 $base = **new** Base;  
  
 *// 根据 mid 从 message 表中获取相应消息* $data = $model::*get*($req->post(**"mid"**));  
 **if**($data) {  
 *// 将信件修改为已读* $data->**isRead** = **"1"**;  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"找不到该信件内容！"**, **null**);  
 }  
 **if**($model->save()) {  
 **return** $base->turnToJSONResp(**"1"**, **""**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"设置已读失败！"**, **null**);  
 }  
}  
  
*/\*\*  
 \* 删除信件  
 \** ***@param*** *Request|null $req  
 \** ***@return*** *\think\response\Json  
 \*/***public function** deleteMessage(Request $req = **null**) {  
 $model = **new** MessageModel;  
 $base = **new** Base;  
  
 $data = $model::*get*($req->post(**"mid"**));  
 **if**($data) {  
 **if**($data->delete()) {  
 **return** $base->turnToJSONResp(**"1"**, **"删除信件成功！"**, **null**);  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"删除信件失败！"**, **null**);  
 }  
 } **else** {  
 **return** $base->turnToJSONResp(**"0"**, **"找不到该信件！"**, **null**);  
 }  
}

C：模型重要成员列表

1. ContactModel：

**class** Message **extends** Model {  
 **public function** generateID($sender\_account) {  
 $data = $this::*all*(**function**($query) **use** ($sender\_account) {  
 $query->where(**"sender\_account"**, $sender\_account);  
 });  
 $length = *count*($data) + 1;  
 **return** ($sender\_account . $length);  
 }  
}

1. MessageModel：

**class** Message **extends** Model {  
  
 */\*\*  
 \* 生成 mid  
 \** ***@param*** *$sender\_account  
 \** ***@return*** *string  
 \*/* **public function** generateID($sender\_account) {  
 $data = $this::*all*(**function**($query) **use** ($sender\_account) {  
 $query->where(**"sender\_account"**, $sender\_account);  
 });  
 $length = *count*($data) + 1;  
 **return** ($sender\_account . $length);  
 }  
}

D：程序清单

| 文件路径 | 文件 | 文件说明 |
| --- | --- | --- |
| message/Controller/ | Contact.php | 患者、专家交流控制类 |
| message/Controller | Message.php | 消息管理控制类 |
| message/Model | Contact.php | 患者、专家交流模型类 |
| message/Model | Message.php | 消息管理模型类 |
| admin/View/ | contact.html | 患者交流模板 |
| admin/View/ | message.html | 消息管理模版 |
| front/View | mail.html | 专家留言模版 |