# 

**牙周帮概要设计规约**

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

[修订历史记录 3](#_Toc534280734)

[1. 引言 4](#_Toc534280736)

[1.1. 概要设计依据 4](#_Toc534280737)

[1.2. 参考资料 4](#_Toc534280738)

[1.3. 假定和约束 4](#_Toc534280739)

[2. 概要设计 5](#_Toc534280740)

[2.1. 系统总体架构设 5](#_Toc534280741)

[2.2. 系统软件结构设计 6](#_Toc534280742)

[2.3. 接口设计 8](#_Toc534280743)

[2.4. 界面设计 10](#_Toc534280744)

[2.5 数据库设计 15](#_Toc534280745)

[2.6 系统出错处理设计 22](#_Toc534280746)

[附录 23](#_Toc534280747)

# 修订历史记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编写日期** | **SEPG** | **版本** | **说明** | **作者** | **。。。。** |
| 2018-12-30 |  | v1 |  | 冯玉山，徐伟喆，赵伟，袁文皓 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 1. 引言

### 1.1. 概要设计依据

* 需求规约
* 需求（分析）规约说明书
* 项目UML模型

### 1.2. 参考资料

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

### 1.3. 假定和约束

* 开发期限至2019/1/12
* 经费限制1000以内
* 设备限制为电脑，手机此类可以登陆网页，微信的电子设备
* 用户个人资料需要翔实，医生与病人之间可正常交流

## 2. 概要设计

### 2.1. 系统总体架构设

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。

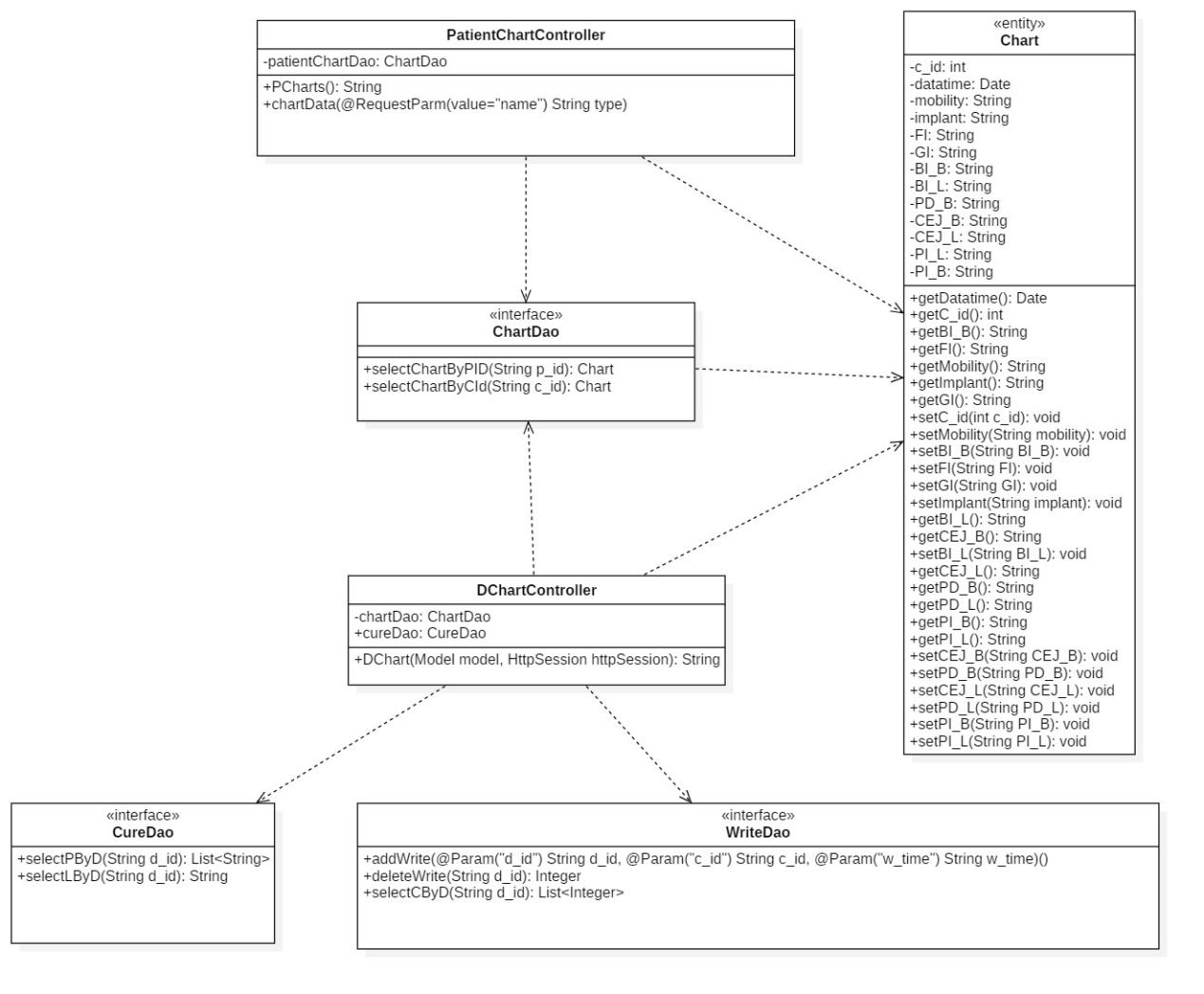
### 2.2. 系统软件结构设计

#### 模块说明

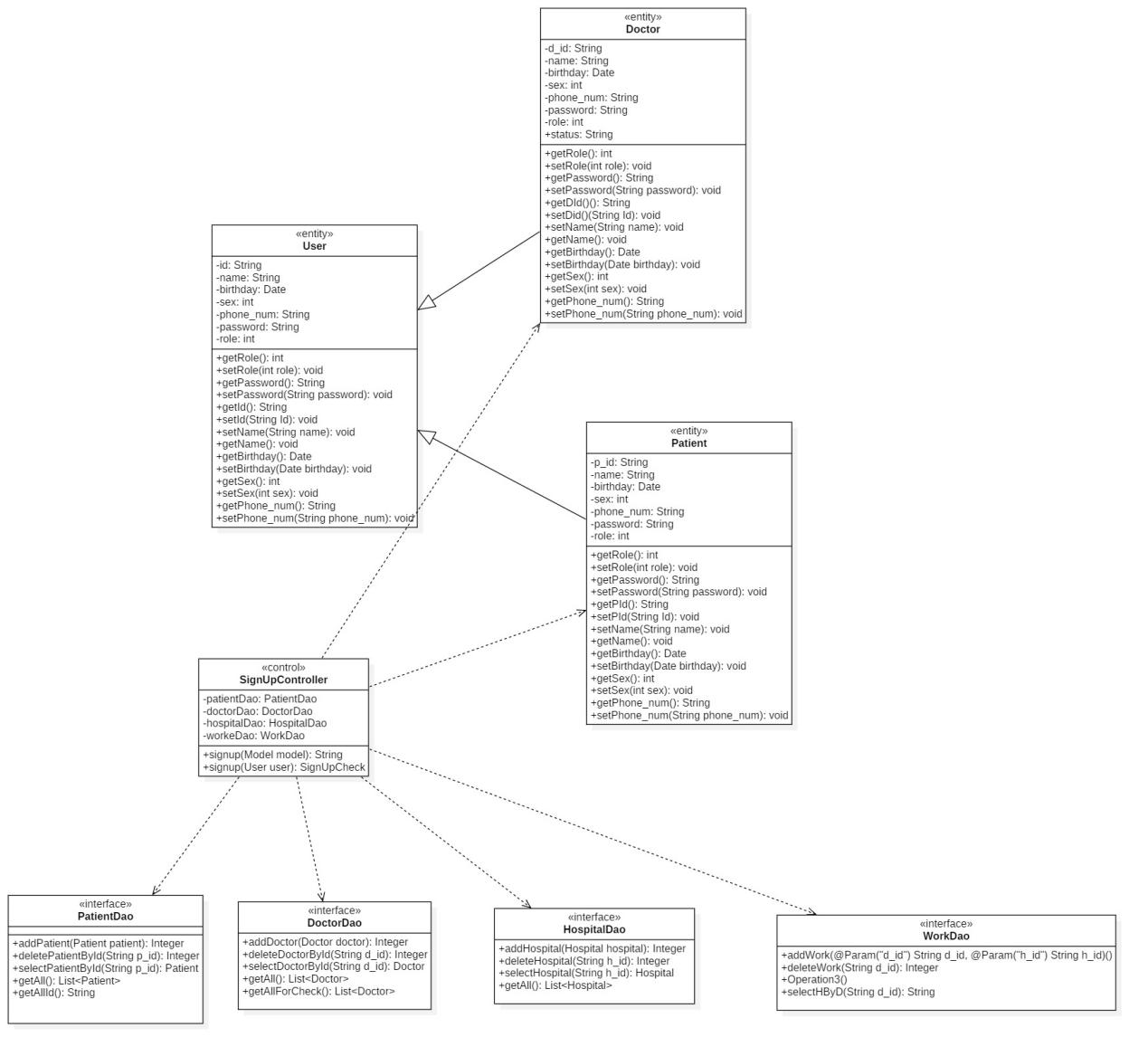
根据项目的需求规约，以及考虑到具体写起来时候的简便，因此主要设计了这么三个模块子系统：诊断表管理子系统，用户管理子系统，预约管理子系统。

#### 类调用关系图

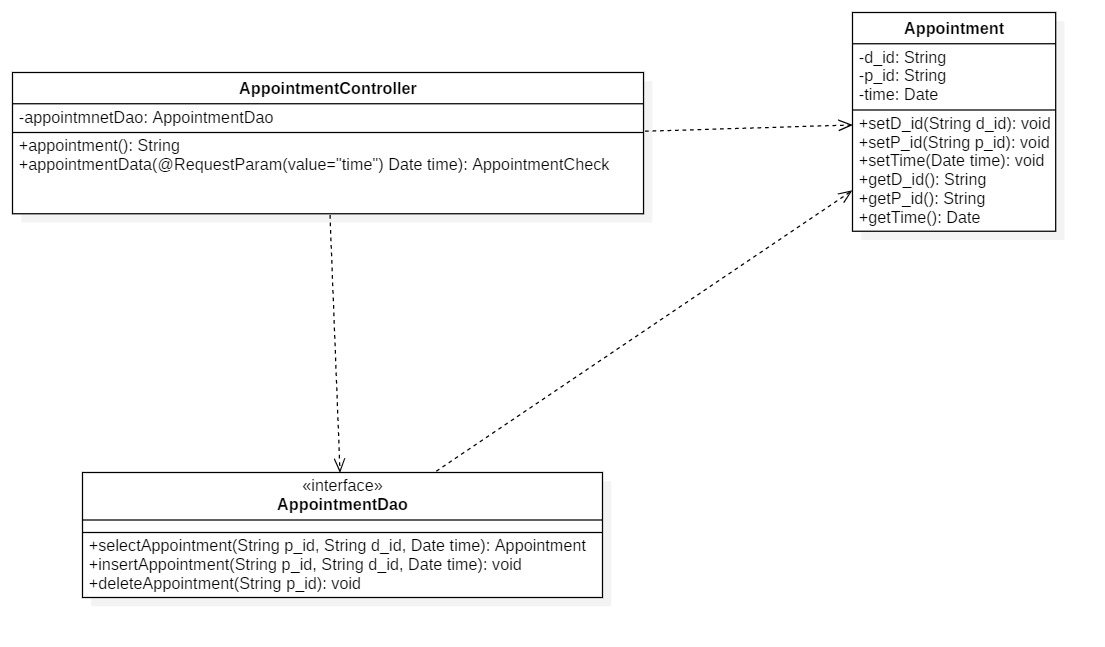
1. 诊断表管理子系统



1. 用户管理子系统



1. 预约管理子系统



### 2.3. 接口设计

#### 2.3.1系统内接口设计

* AppointmentDao

接口方法：

public Appointment selectAppointemntById(String p\_id);

public Integer delectAppointment(String p\_id);

public Integer insertAppointment(Appointment appointment);

public Integer updateAppointment(Timestamp s\_time, String p\_id);

public String selectD\_id(String p\_id);

* ChartDao

接口方法：

public List<Chart> selectChartById(String p\_id);

public List<String>selectCIdById(String p\_id);

public ChartWithDoctors selectInformationByCId(String c\_id);

public Chart selectChartByCId(String c\_id);

* CureDao

接口方法：

public List<String> selectPByD(String d\_id);

public String selectLByD(String d\_id);

public String selectLByP(String p\_id);

* DoctorDao

接口方法：

public Integer addDoctor(Doctor doctor);

public Integer deleteDoctorById(String d\_id);

public Doctor selectDoctorById(String d\_id);

public List<Doctor> getAll();

public List<Doctor> getAllForCheck();

* HospitalDao

接口方法：

public Integer addHospital(Hospital hospital);

public Integer deleteHospital(String h\_id);

public Hospital selectHospital(String h\_id);

public List<Hospital> getAll();

* PatientDao

接口方法：

public Integer addPatient(Patient patient);

public Integer deletePatientById(String p\_id);

public Patient selectPatientById(String p\_id);

public List<Patient> getAll();

public String[] getAllId();

public String[]selectDoctorBypId(String p\_id);

* UserListDao

接口方法：

public Integer addUserList(UserList userList);

public UserList selectUserById(String user\_id);

* WorkDao

接口方法：

public Integer addWork(@Param("d\_id") String d\_id, @Param("h\_id") String h\_id);

public Integer deleteWork(String d\_id)

public String selectHByD(String d\_id);

* WriteDao

接口方法：

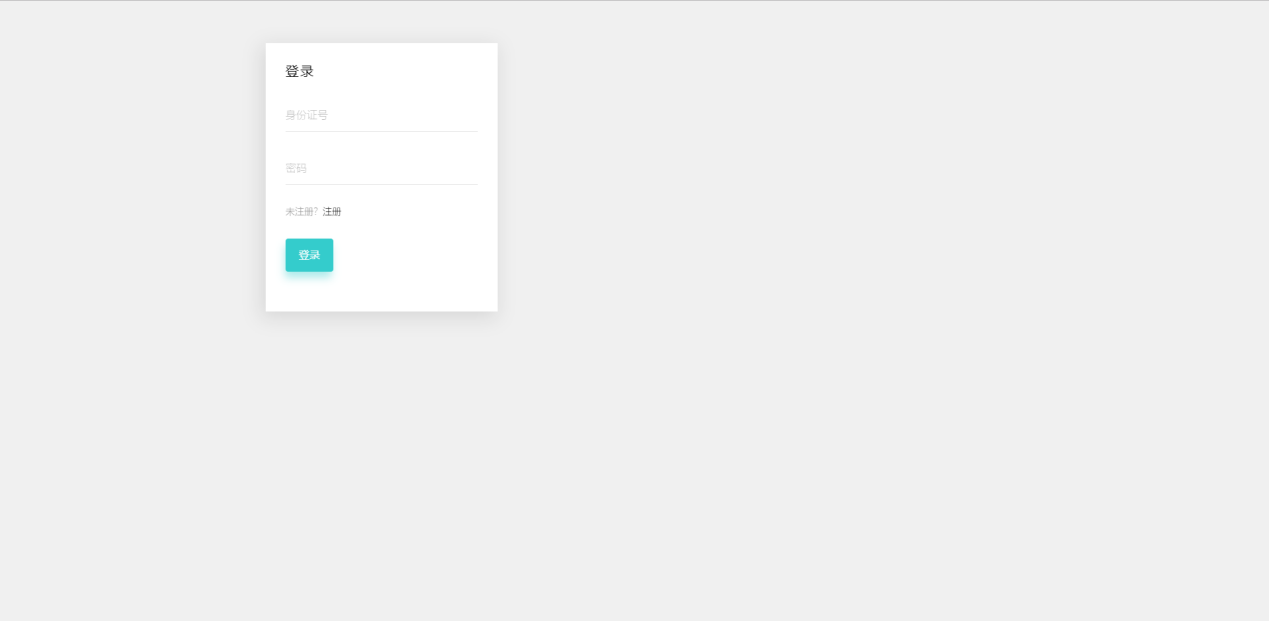
public Integer addWrite(@Param("d\_id") String d\_id, @Param("c\_id") String c\_id,

public Integer deleteWrite(String d\_id);

public String selectCByD(String d\_id);

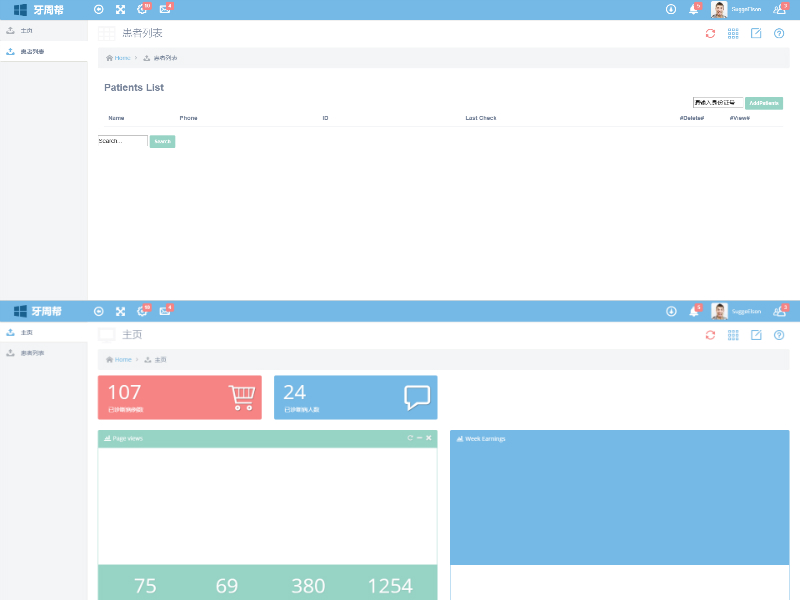
### 2.4. 界面设计

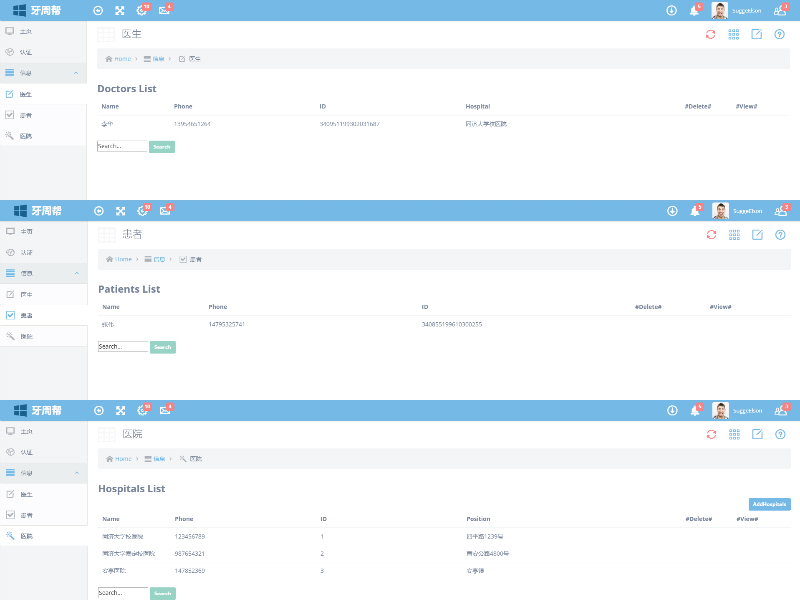
网页端用户设计如下图所示。用户首先进入主页。

点击登录后，进入登陆界面。

根据用户的角色，自动进入到不同的界面中。

1. **医生界面**：医生界面包括主页和查看病人信息的页面。



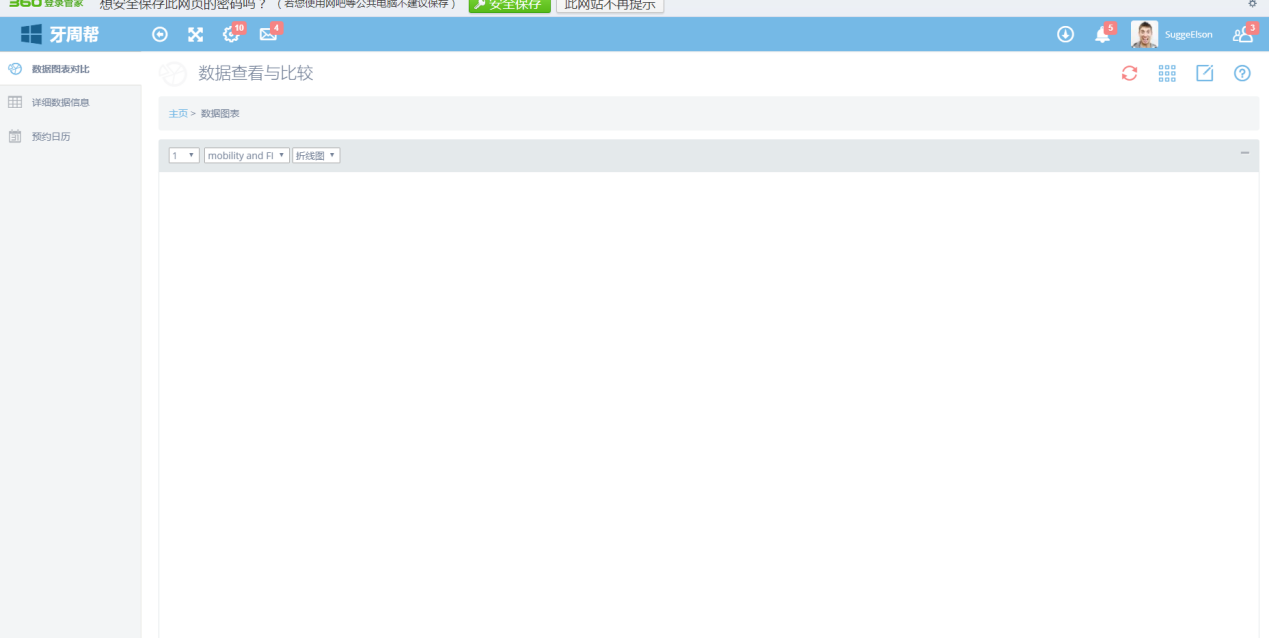
1. **管理员界面**：包括对医生、病人、医院信息的管理界面。
2. **病人主界面：**

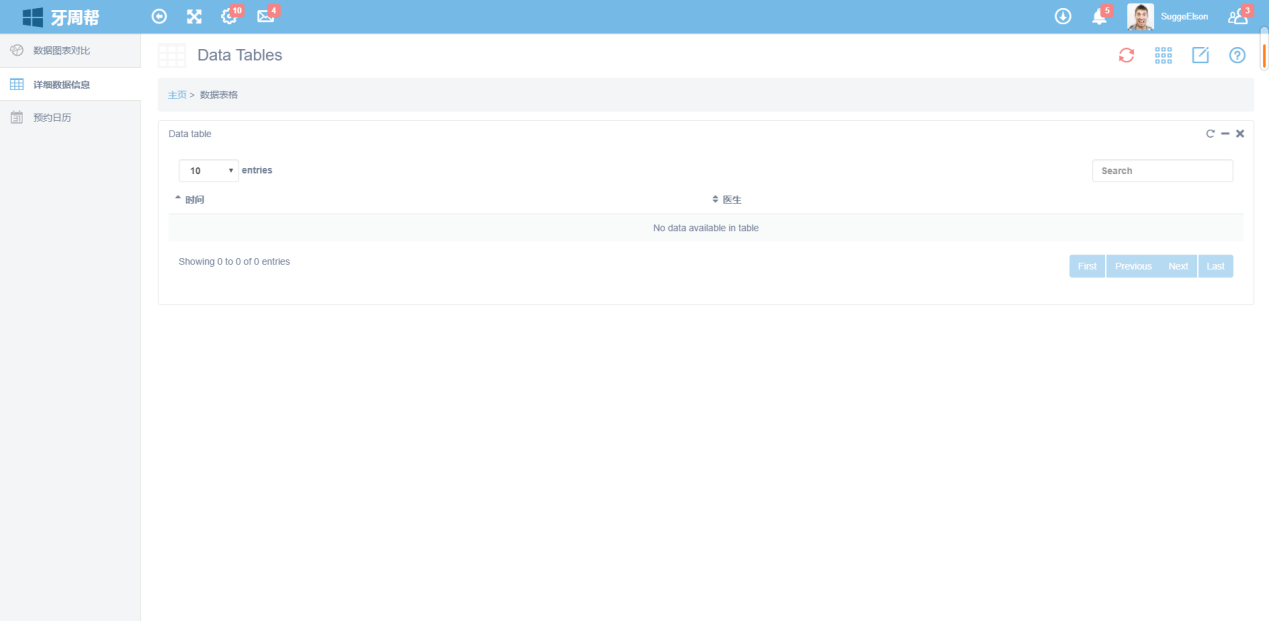
**在这个界面病人可以选择进入相应的功能界面或是查看个人信息**

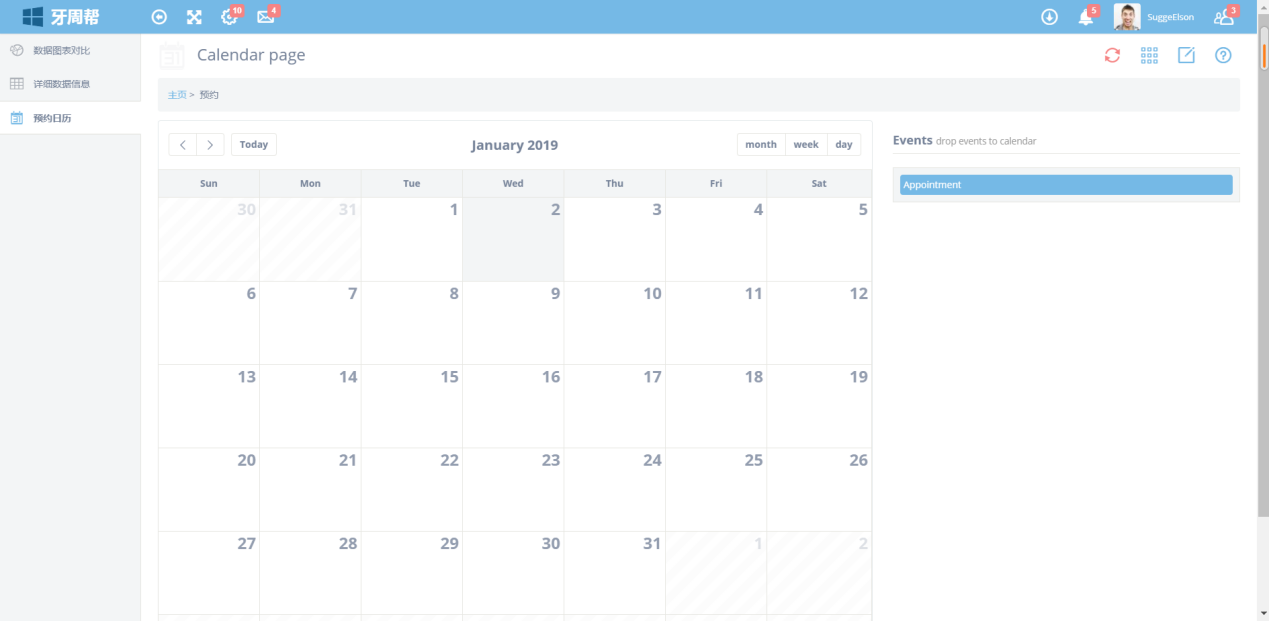


1. **病人功能界面：**

**三个功能界面，分别是数据对比，详细数据查看以及预约功能的界面。**

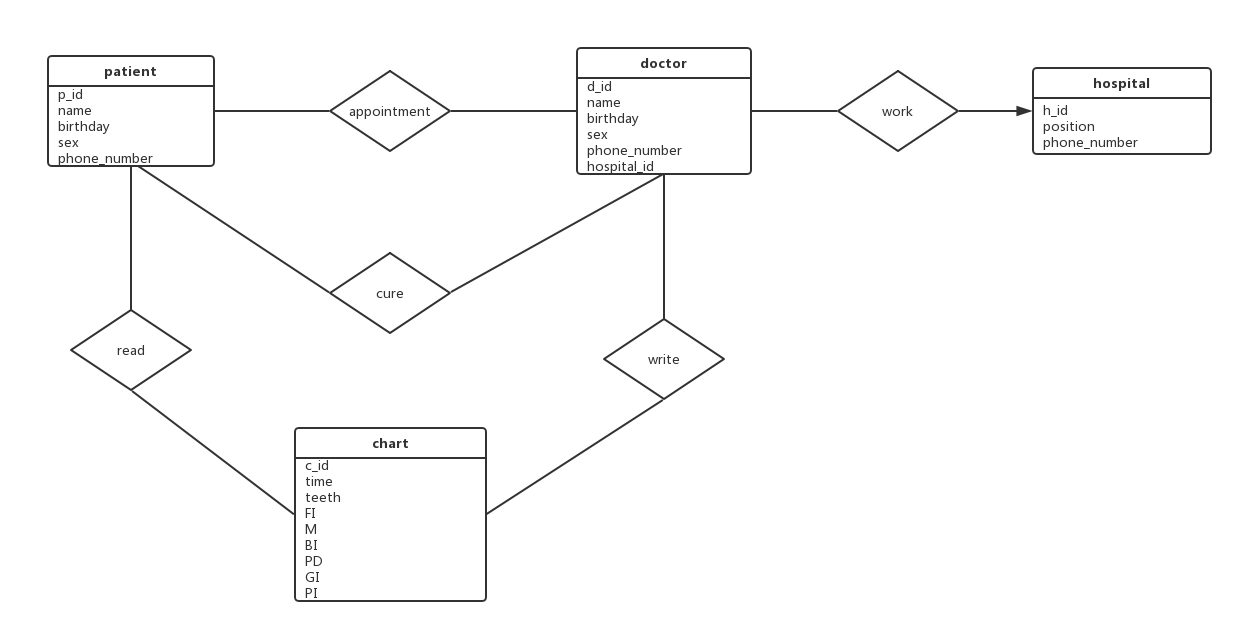






### 2.5 数据库设计

#### 2.5.1数据库E-R图



#### 2.5.2数据库设计

病人基本信息patient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| p\_id | varchar(45) | 作为病人的唯一标识符 | “400000000000000000” | PK |
| name | varchar(45) | 病人的姓名 | “李四” |  |
| birthday | Date | 病人的生日 | “2018-10-09 22：00：00” |  |
| sex | int(1) | 病人性别 | 0（1） | 0表示男性，1表示女性 |
| phone\_num | varchar(11) | 病人手机号码 | "1345678904" |  |
| password | varchar(45) | 病人登陆密码 | "20huo" |  |
| role | int(1) | 用户身份 | 0 | 病人列表role为1，1表病人，0表医生 |

医生基本信息doctor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| d\_id | varchar(45) | 作为医生的唯一标识符 | “400000000000000000” | PK |
| name | varchar(45) | 医生的姓名 | “李四” |  |
| birthday | Date | 医生的生日 | “2018-10-09 22：00：00” |  |
| sex | int(1) | 医生性别 | 0（1） | 0表示男性，1表示女性 |
| phone\_num | varchar(11) | 医生手机号码 | "1345678904" |  |
| password | varchar(45) | 医生登陆密码 | "20huo" |  |
| role | int(1) | 用户身份 | 1 | 因为是医生列表，所以role为0，1表病人，0表医生 |

医院基本信息hospital

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| h\_id | varchar(45) | 作为医院的唯一标识符 | “1” | PK |
| position | varchar(45) | 医院的地址 | “四平路1239号” |  |
| phone\_num | varchar(45) | 医院的电话 | “12345678911” |  |
| hospital\_name | varchar(45) | 医院的名字 | “同济大学校医院” |  |

预约信息表appointment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| d\_id | varchar(45) | 引入doctor表的主码作为外码 | “400000000000000000” | FK |
| p\_id | varchar(45) | 引入patient表的主码作为外码 | “400000000000000000” | PK,FK |
| s\_time | datetime | 预约时间 | “2018-10-09 22：00：00” |  |
| status | int(2) | 预约状态 | 0（1） | 0表示病人已预约，等待医生同意。1表示医生通过预约请求。 |

牙周数据表chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| c\_id | int(11) | 作为牙周检查表的唯一标识符 | “10000000000” | 标识项，唯一索引 |
| time | datetime | 预约时间 | “2018-10-09 22：00：00” |  |
| mobility | varchar(400) | 牙齿松动度 |  | 取值1、2、3 |
| implant | varchar(400) | 植体 |  |  |
| FI | varchar(400) | 根分叉病变 |  | 取值0-3 |
| GI | varchar(400) | 牙龈指数 |  | 取值0-3 |
| BI\_B | varchar(400) | 颊侧出血指数 |  | 取值0-5 |
| BI\_L | varchar(400) | 舌侧出血指数 |  | 取值0-5 |
| PD\_B | varchar(400) | 颊侧探诊深度 |  | -1000~1000 |
| PD\_L | varchar(400) | 舌侧探诊深度 |  | -1000~1000 |
| CEJ\_B | varchar(400) | 颊侧龈缘 |  | -1000~1000 |
| CEJ\_L | varchar(400) | 舌侧龈缘 |  | -1000~1000 |
| PI\_B | varchar(400) | 颊侧菌斑指数 |  | 取值0-3 |
| PI\_L | varchar(400) | 舌侧菌斑指数 |  | 取值0-3 |

治疗表cure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| d\_id | varchar(45) | 引入doctor表的主码作为外码 | “400000000000000000” | PK,FK |
| p\_id | varchar(45) | 引入patient表的主码作为外码 | “400000000000000000” | PK,FK |
| s\_time | datetime | 治疗开始时间 | “2018-10-09 22：00：00” |  |
| l\_time | datetime | 治疗结束时间 | “2018-10-09 22：00：00” |  |

病人与牙周表关系表read

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| p\_id | varchar(45) | 引入patient表的主码作为外码 | “400000000000000000” | FK |
| c\_id | int(11) | 引入cure表的主码作为外码 | “10000000000” | PK,FK |

医生牙周表关系表write

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| d\_id | varchar(45) | 引入doctor表的主码作为外码 | “400000000000000000” | FK |
| c\_id | int(11) | 引入cure表的主码作为外码 | “10000000000” | PK,FK |
| w\_time | datetime | 牙周表生成时间 | “2018-10-09 22：00：00” |  |

医生工作表work

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| d\_id | varchar(45) | 引入doctor表的主码作为外码 | “400000000000000000” | PK,FK |
| h\_id | varchar(45) | 引入hospital表的主码作为外码 | “1” | PK,FK |

用户信息表user\_list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 数据类型 | 描述 | 示例 | 备注 |
| user\_id | varchar(45) | 用户id | “400000000000000000” |  |
| user\_password | varchar(45) | 用户密码 | "20huo" |  |
| user\_role | int(1) | 用户角色 | 0 | 0为医生，1为病人 |

### 2.6 系统出错处理设计

#### 2.6.1出错信息

1. 用户在提交信息时出现网络中断或数据库故障（如修改个人信息、登录等操作）

1.1 出错信息：提交失败

1.2 处理方法：提示用户提交失败

1. 用户在获取信息时出现网络中断或数据库故障（如获取牙周表、获取病人列表等操作）

2.1 出错信息：获取信息失败

2.2 处理方法：提示用户获取信息失败

1. 系统宕机

3.1 出错信息：无

3.2 处理方法：系统管理员检查各项服务的运行状况并进行维护

1. 用户输入信息格式错误（个人信息、填牙周表等操作）

4.1 出错信息：提交失败

4.2 处理方法：提示用户输入格式错误，等待用户重新输入

#### 2.6.2补救措施

1. 对数据库信息进行经常性的备份，以减少数据库故障导致的数据丢失。
2. 引入版本管理工具git，对于出现的错误可以进行有效控制，通过版本回退等操作使软件可以重新运行。
3. 阻止用户的不规范操作，如：对错误的输入进行弹窗提示、阻止用户访问权限之外的页面等。

#### 2.6.3系统维护设计

1. 对系统功能的维护：周期性地备份数据库。
2. 提供运行日志管理和安全审计功能，各项系统操作记录在运行日志中，可追踪系统的历史使用情况，方便对系统的维护。
3. 使用git进行版本管理，灵活使用分支功能有效避免了版本混乱的情况。