

郑州大学西亚斯国际学院

本科毕业论文(设计)

题目 在线医疗平台的设计与实现

——医生模块

指导教师 邵彥 职称 副教授

学生姓名 赵仕杰 学号 2019325140266

专 业 软件工程（专升本）

班 级 2019 软件工程专升本 2 班

院（系） 电子信息工程学院

完成时间 2021 年 4 月 2 日

在线医疗平台的设计与实现——患者模块

摘 要

在进入信息化时代的今天，各行各业的信息化空前普及，疫情的爆发虽然给中国社会的经济发展带来了一定影响，但同时也刺激了线上医疗行业的发展，加速了中国医疗互联网化，让人们对于医疗服务模式产生了更加深入理解，然而对于现阶段的医院行业，高度的信息化管理只存在于大型的医院当中，对于中小型医院信息的普及化还不能全面覆盖，医生与患者之间也不能畅所欲言。在技术架构上，本系统业务主要基于 PyCharm 与 MySQL 实现，并针对一些特殊需求，使用了合适的技术进行实现。利用前端框架 Bootstrap，对不同尺寸的设备进行适配并测试，满足了系统要求医生和病人能够在 PC 端和移动设备都能操作的需求。本篇将从在线医疗系统的背景、设计意义、需求分析以及实现的功能等展开研究。

关键词 在线医疗，PyCharm，MySQL。

Design and implementation of online medical platform -- patient module

ABSTRACT

Entered the information age today, informationization unprecedented popularity in all walks of life, although outbreaks to brought some impact on the economic development of Chinese society, but also stimulated the development of the online health care industry, accelerate the China medical Internet, let people to medical service pattern has a more in-depth understanding, but for the present stage of the hospital industry, a high degree of information management only exists in large hospital, for small and medium-sized hospital information popularization is not universal coverage, also can't speak freely between doctors and patients. In terms of technical architecture, the business of this system is mainly realized based on PyCharm and MySQL, and appropriate technologies are used to realize it according to some special requirements. The front-end framework Bootstrap was used to adapt and test devices of different sizes, which met the requirements of the system that doctors and patients could operate both on PC and mobile devices. This paper will study the background, design significance, requirement analysis and functions of online medical system.

KEY WORDS The online medical, PyCharm, MySQL.

目 录

中文摘要.....	
英文摘要.....	
1 绪论.....	1
1.1 开发背景.....	1
1.2 课题研究的意义.....	1
1.3 论文内容的介绍.....	2
2 系统相关技术介绍.....	3
2.1 三层架构.....	3
2.1.1 开发原理.....	3
2.1.2 表示层.....	3
2.1.3 业务逻辑层.....	3
2.1.4 数据访问层.....	3
2.2 Python 简介.....	4
2.3 PyCharm 简介.....	4
2.4 Django 简介.....	4
2.5 MySQL 简介.....	5
2.6 jQuery 简介.....	5
2.7 Bootstrap 简介.....	6
3 系统需求分析和整体设计.....	7
3.1 系统可行性研究.....	7
3.1.1 技术可行性.....	7
3.1.2 经济可行性.....	7
3.1.3 操作可行性.....	7
3.1.4 小结.....	8
3.2 系统需求分析.....	8
3.3 系统总体功能设计.....	8
3.4 系统总体数据库设计.....	9
4 医生模块的详细设计与实现.....	11
4.1 医生登录功能的实现.....	11
4.2 医生中心的实现.....	11
4.3 在线诊断功能的实现.....	11

4.4 查询医生信息功能的实现.....	12
4.5 电子处方功能的实现.....	14
5 模块和系统测试.....	16
5.1 医生登录功能测试.....	16
5.2 医生注册功能测试.....	16
结 论.....	18
致 谢.....	19
参考文献.....	20
附 录.....	21

1 绪论

1.1 开发背景

2020 年，在疫情最为严峻的时期，国家卫健委连续发文，要求各地积极运用“互联网+”、大数据等信息技术，减少线下诊疗压力和交叉感染风险^[16]。近年来，以互联网为依托的健康教育、医疗信息查询、电子健康档案、电子处方、远程医疗和康复等多种方式的医疗健康服务，正在悄然改变着传统医疗服务模式。2015 年，《国务院关于积极推进“互联网+”行动的指导意见》出台，提出加快发展基于互联网的医疗、健康等新兴服务，鼓励医药行业利用电子商务平台优化采购、分销体系，提升企业经营效率。我国从 2012 年开始就推行了“分时段预约诊疗”、市民卡“诊间结算”到目前，杭州“互联网+智慧医疗”推进已有 6 年^[2]。通过借助大数据、物联网技术，以及微信、APP 等载体，形成医疗智慧化建设，打破信息孤岛。发展基于互联网的在线医疗服务，逐步建立跨医院的医疗数据共享交换标准体系。积极利用移动互联网提供在线问诊、开电子处方等服务，引导医疗机构面向中小城市和农村地区开展基层检查、上级诊断等远程医疗服务。线上医患交流是互联网医疗的核心环节，由医患双方作为医疗服务的主体，借助新技术延展了医疗服务的场景和边界。对线上医患交流过程和效果的深入考察，是促进互联网医疗服务蓬勃发展的突破点和必经之路。基于线上医患交流目前在实践层面的进展状况，从传播学人际传播和群体间传播的视角切入，探索线上医患交流行为模式、效果及影响机制，对促进互联网医疗的蓬勃发展，拓展互联网医疗的服务内容及用户渗透率，进而缓解我国医疗服务的诸多困境，提高人们的健康水平，具有重要的理论与现实意义。

1.2 课题研究的意义

对于现阶段的医院行业，高度的信息化管理只存在于大型的医院当中，对于中小型医院信息的普及化还不能全面覆盖，医生与患者之间也不能畅所欲言。针对以上问题本平台通过及时咨询、在线医患沟通平台、医师从业人员以及民众之间，建立起了基于 Internet 的有效的沟通渠道，为医生与患者搭起了一个很好的沟通桥梁，医生不仅可以在线排查患者病情、开电子处方，而且可以查询医生信息及科室编辑等。在未来的医疗体系下，感冒等普通的病症，人们无论身处何处，只需要在医疗平台上和医

生进行远程诊断，在线付费后即可享受到线上诊断的服务，全程方便、快捷、有效。在线医疗不仅可以避免疫情期间患者的交叉感染，还极大缓解了我国医疗资源分布不均衡的问题。进一步加强医疗信息的科学性、权威性、指导性、实用性、服务性、及时性，更好地为患者提供更全面更专业的优质健康服务。

1.3 论文内容的介绍

本文开篇首先介绍了课题研究的背景、目的及意义。其次，讲述系统所涉及的基础理论知识，以及使用到相关技术的介绍。然后对系统进行需求分析，可行性分析及系统功能和数据库的设计。接着是分别介绍各个功能的操作流程和界面展示，最后对系统进行测试。

2 系统相关技术介绍

2.1 三层架构

2.1.1 开发原理

三层架构中主要功能与业务逻辑一般要在业务逻辑层进行信息处理和实现，其中三层体系架构中的客户端和数据库要预设中间层，成为组建层。三层架构中的三层具有一定的逻辑性，即是把三层设置到同一个计算机系统中，把业务协议、合法校验以及数据访问等程序归置到中间层进行信息处理，一般客户端无法和数据库进行数据传输，主要是利用 COM/DCOM 通讯和中间层构建衔接通道，实现中间层与数据库的数据传输，进而实现客户端与数据库的交互。

2.1.2 表示层

表示层又称表现层 UI，位于三层构架的最上层，与用户直接接触，主要是 B/S 信息系统中的 Web 浏览页面。作为 Web 浏览页面，表示层的主要功能是实现系统数据的传入与输出，在此过程中不需要借助逻辑判断操作就可以将数据传送到 BLL 系统中进行数据处理，处理后会将其处理结果反馈到表示层中。换句话说，表示层就是实现用户界面功能，将用户的需求传达和反馈，并用 BLL 或者是 Models 进行调试，保证用户体验。

2.1.3 业务逻辑层

业务逻辑层 BLL 的功能是对具体问题进行逻辑判断与执行操作，接收到表现层 UI 的用户指令后，会连接数据访问层 DAL，访问层在三层构架中位于表示层与数据层中间位置，同时也是表示层与数据层的桥梁，实现三层之间的数据连接和指令传达，可以对接收数据进行逻辑处理，实现数据的修改、获取、删除等功能，并将处理结果反馈到表示层 UI 中，实现软件功能。

2.1.4 数据访问层

数据访问层 DAL 是数据库的主要操控系统，实现数据的增加、删除、修改、查询等操作，并将操作结果反馈到业务逻辑层 BLL。在实际运行的过程中，数据访问

层没有逻辑判断能力，为了实现代码编写的严谨性，提高代码阅读程度，一般软件开发人员会在该层中编写 `Data AccessCommon`，保证数据访问层 DAL 数据处理功能。

2.2 Python 简介

Python 是一种计算机程序设计语言，由吉多·范罗苏姆创造，第一版发布于 1991 年^[3]。Python 提供了高效的高级数据结构，还能简单有效地面向对象编程。其解释器易于扩展，可以使用 C 或 C++ 扩展新的功能和数据类型，也可用于可定制化软件中的扩展程序语言^[9]。由于 Python 语言的简洁性、易读性以及可扩展性，在国外用 Python 做科学计算的研究机构日益增多，一些知名大学已经采用 Python 来教授程序设计课程。Python 在设计上坚持了清晰划一的风格，这使得 Python 成为一门易读、易维护，并且被大量用户所欢迎的、用途广泛的语言。因此 Python 语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表，甚至开发科学计算应用程序。

2.3 PyCharm 简介

PyCharm 是一种 Python IDE，带有一整套可以帮助用户在使用 Python 语言开发时提高其效率的工具，PyCharm 拥有一般 IDE 具备的功能，比如调试、语法高亮、Project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该 IDE 提供了一些高级功能，以便用于支持 Django 框架下的专业 Web 开发。这些功能在先进代码分析程序的支持下，使 PyCharm 成为 Python 专业开发人员和刚起步人员使用的有力工具。

2.4 Django 简介

Django 是一个 Web 应用程序框架的免费开放源代码，是用 Python 写成的^[4]。Django 的主要目的是简便、快速的开发数据库驱动的网站。它强调代码复用，多个组件可以很方便的以“插件”形式服务于整个框架，Django 有许多功能强大的第三方插件，你甚至可以很方便的开发出自己的工具包。这使得 Django 具有很强的可扩展性。

Django 采用 MTV (Model, Template, View) 的设计，就是把 Web 应用分为数据模型（存取层）、模板（表现层）、视图（业务逻辑层）三层^[7]。其中，模型的职责是处理与数据相关的一切事务；视图负责处理业务逻辑；模板的职责是将前端的页

面展示给用户。

2.5 MySQL 简介

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下公司。MySQL 最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的 RDBMS (Relational Database Management System, 关系数据库管理系统) 应用软件之一^[6]。

MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

2.6 jQuery 简介

jQuery 是一个快速、简洁的 JavaScript 框架，是继 Prototype 之后又一个优秀的 JavaScript 代码库。jQuery 有以下特点：

(1) 快速获取文档元素

jQuery 的选择机制构建于 CSS 的选择器，它提供了快速查询 DOM 文档中元素的能力，而且大大强化了 JavaScript 中获取页面元素的方式。

(2) 提供漂亮的页面动态效果

jQuery 中内置了一系列的动画效果，可以开发出非常漂亮的网页，许多网站都使用 jQuery 的内置的效果，比如淡入淡出、元素移除等动态特效。

(3) 创建 AJAX 无刷新网页

AJAX 是异步的 JavaScript 和 XML 的简称，可以开发出非常灵敏无刷新的网页，特别是开发服务器端网页时，比如 PHP 网站，需要往返地与服务器通信，如果不使用 AJAX，每次数据更新不得不重新刷新网页，而使用 AJAX 特效后，可以对页面进行局部刷新，提供动态的效果。

(4) 提供对 JavaScript 语言的增强

jQuery 提供了对基本 JavaScript 结构的增强，比如元素迭代和数组处理等操作。

(5) 增强的事件处理

jQuery 提供了各种页面事件，它可以避免程序员在 HTML 中添加太多事件处理代码，最重要的是，它的事件处理器消除了各种浏览器兼容性问题。

(6) 更改网页内容

jQuery 可以修改网页中的内容，比如更改网页的文本、插入或者翻转网页图像，jQuery 简化了原本使用 JavaScript 代码需要处理的方式。

2.7 Bootstrap 简介

Bootstrap 是美国 Twitter 公司的设计师 Mark Otto 和 Jacob Thornton 合作基于 HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它即是由动态 CSS 语言 Less 写成。Bootstrap 一经推出后颇受欢迎，一直是 GitHub 上的热门开源项目，包括 NASA 的 MSNBC 的 Breaking News 都使用了该项目。国内一些移动开发者较为熟悉的框架，如 WeX5 前端开源框架等，也是基于 Bootstrap 源码进行性能优化而来。

3 系统需求分析和整体设计

3.1 系统可行性研究

可行性研究的目的是不是解决问题，而是确定问题是否值得去解决。下面便从技术可行性、经济可行性和操作可行性三方面分别研究[15]。

3.1.1 技术可行性

该系统的开发环境和配置都是比较好安装的，本平台使用 Python 语言开发，使用比较成熟的 MySQL 数据库进行对系统前台及后台的数据交互，根据技术语言对数据库，结合需求进行修改维护，可以使得系统运行更具有稳定性和安全性，从而完成实现系统的开发。

(1) 硬件可行性分析

系统的硬件要求方面不存在特殊的要求，只需要在普通的硬件配置就能够轻松的实现，只是需要确保系统的正常工作即可，以及拥有较高的效率。如果有特别低的硬件，它可以导致系统的低性能以及效率低，从而导致整个网站的运行不顺畅。以目前普遍的个人计算机的配置而言，这是十分容易实现的。因此，本系统的开发在硬件方面是可行的。

(2) 软件可行性分析

本系统将使用 PyCharm 进行开发，使用 MySQL 进行前后台的交互。因此，考虑到系统的实际情况，选择 Python 作为系统开发技术，通过以上分析，系统的设计和实现在软件中是可行的。

综上所述，我们从两个方面进行了可行性研究，可以看出系统的开发没有问题。

3.1.2 经济可行性

系统是基于 Python 语言开发的软件，采用 PyCharm 平台以及 MySQL 小型数据库，均属于开源免费产品使用，对现在的开发成本以及维护成本上来说，是比较低廉的。

3.1.3 操作可行性

本系统采用 Python 技术，利用网络就能够进行访问和操作，且界面简单易操作，

用户只要平时有在用电脑，都能进行访问和操作。本系统具有易操作、易管理、交互性好的特点，在操作上是非常简单的。因此本系统可以进行开发。系统的开发流程遵循软件开发思想，分阶段完成系统分析、设计、实现等研究内容网，系统界面友好，使用者不需要经过专门的培训即可熟练操作

综上所述，此系统开发目标已明确，在技术和经济等方面都可行，并且投入少、见效快。因此系统的开发是完全可行的。

3.1.4 小结

通过对以上对系统的经济、技术和运行方面的可行性分析，最终发现本系统的技术相当成熟，有友好的界面、操作简单、运行安全可靠。

3.2 系统需求分析

“在线医疗平台的设计与实现——医生模块”是针对于患者与医生可以在线交流设计的。该系统采用 B/S 模式，后台的数据库采用目前比较流行的 MySQL，该数据库系统在安全性、准确性、运行速度方面有绝对的优势，并且处理数据量大，效率高；前台采用 PyCharm 作为主要的开发工具，可实现与 MySQL 数据库的无缝连接。

3.3 系统总体功能设计

系统功能图如图 3-1 所示：

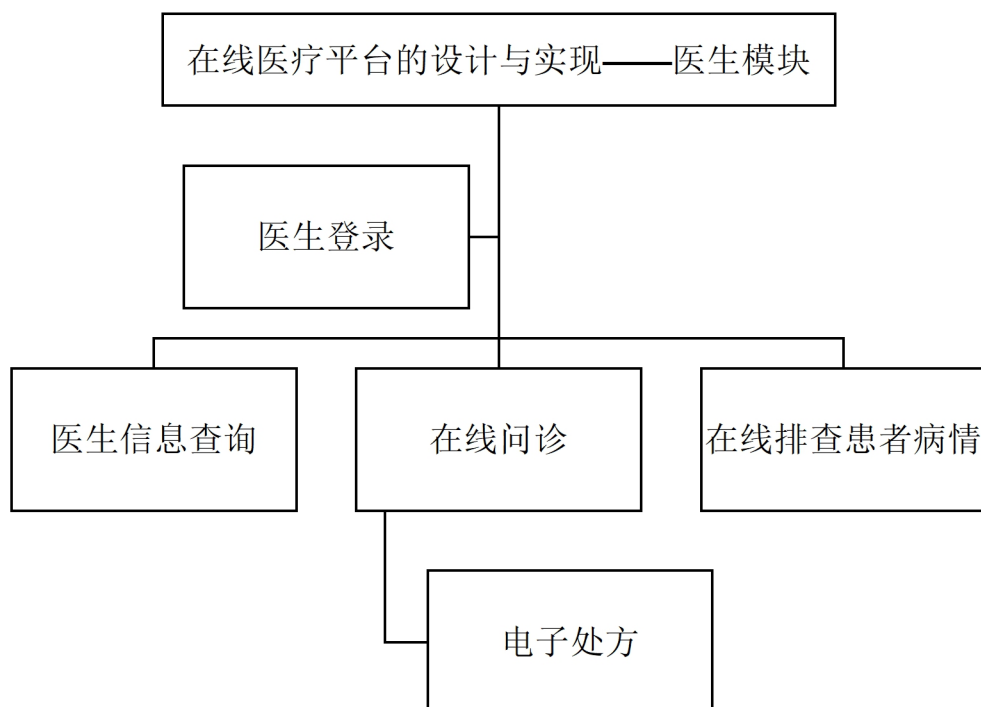


图 3-1 系统功能图

3.4 系统总体数据库设计

该数据库共有七个表，其中共有用户、信息管理、聊天三个主体。本数据库没有使用外键，避免冗余。

医生信息表如图 3-2 所示：

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生 ID
d_name	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生姓名'
d_local	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生住址'
d_department	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生科室'
d_hospital	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生所在医院'
d_phone	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生电话'
d_title	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生简介'
d_experience	VARCHAR(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生经历'
d_education	VARCHAR(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生学历'
d_skills	VARCHAR(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生技能'
d_motto	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	座右铭'
is_working	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	是否工作

图 3-2 医生信息表

医生账户表如图 3-3 所示：

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生账户 ID
d_account	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生账户'
d_password	VARCHAR(25)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生密码'

图 3-3 医生账户表

问诊信息表如图 3-4 所示：

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	问诊信息 ID
d_department	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	科室'
d_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	医生 ID
info_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	挂号 ID
is_working	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	是否进行中
is_done	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	是否结束
opentime	DATETIME(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	开始时间
is_subsequent_visit	TINYINT(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	是否为复诊
subsequent_visit_time	DATETIME(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	复诊时间
done_time	DATETIME(6)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	结束时间
u_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	用户 ID
ill	VARCHAR(150)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	电子处方'

图 3-4 问诊信息表

聊天问诊表如图 3-5 所示：

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	问诊 ID
appointment_id	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	问诊信息 ID
master	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	发送人'
message	VARCHAR(150)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	信息'
flag	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	聊天锚点

图 3-5 聊天问诊表

数据库 E-R 图如图 3-6 所示：

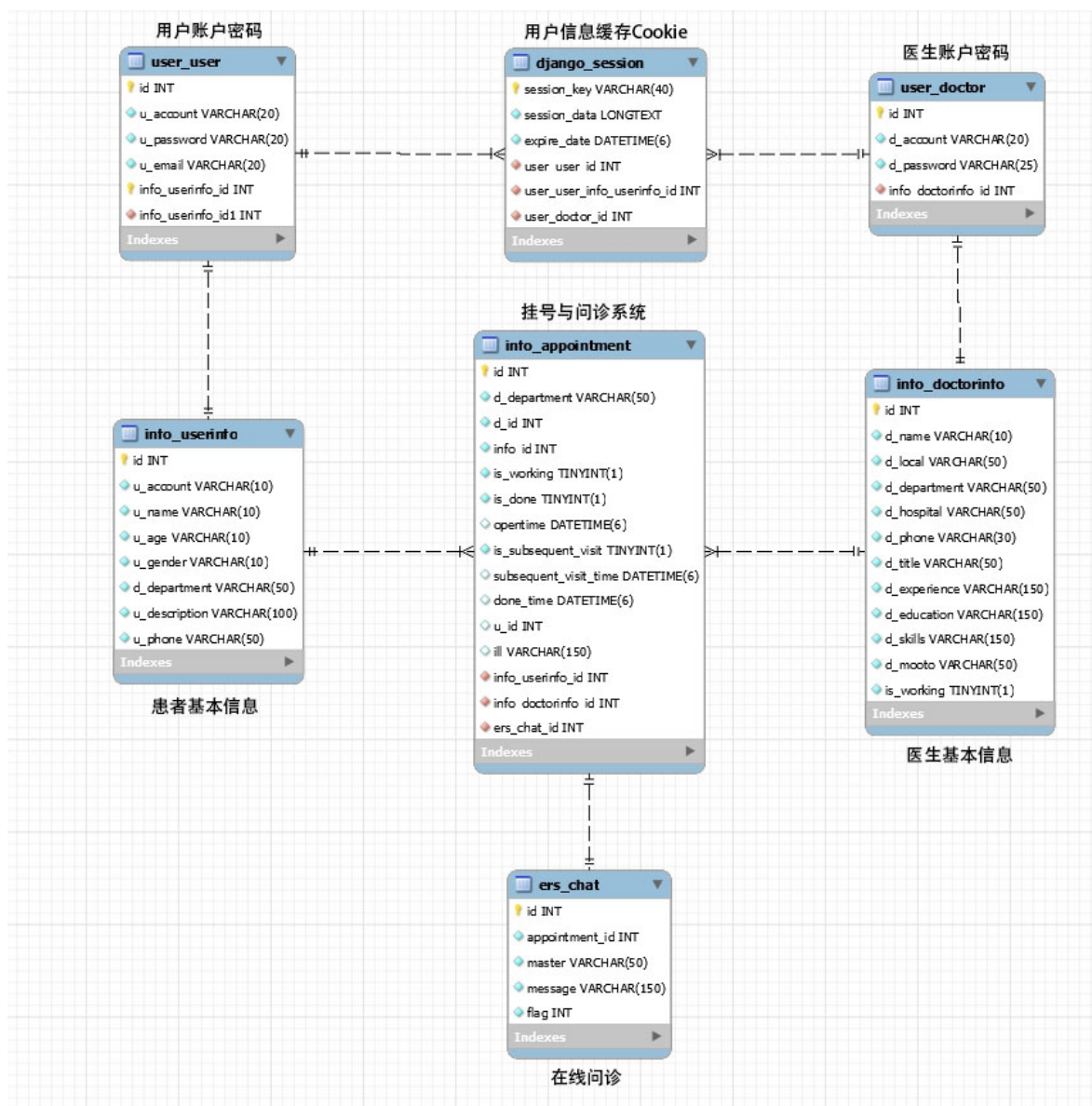


图 3-6 E-R 图

4 医生模块的详细设计与实现

4.1 医生登录功能的实现

医生点击登录出现登录框，输入账户密码即可登录。

登录界面图如图 4-1 所示：



图 4-1 登录界面图

4.2 医生中心的实现

医生登录后即可跳转医生中心，左边为未能问诊，右边为已经问诊。

医生中心界面图如图 4-2 所示：

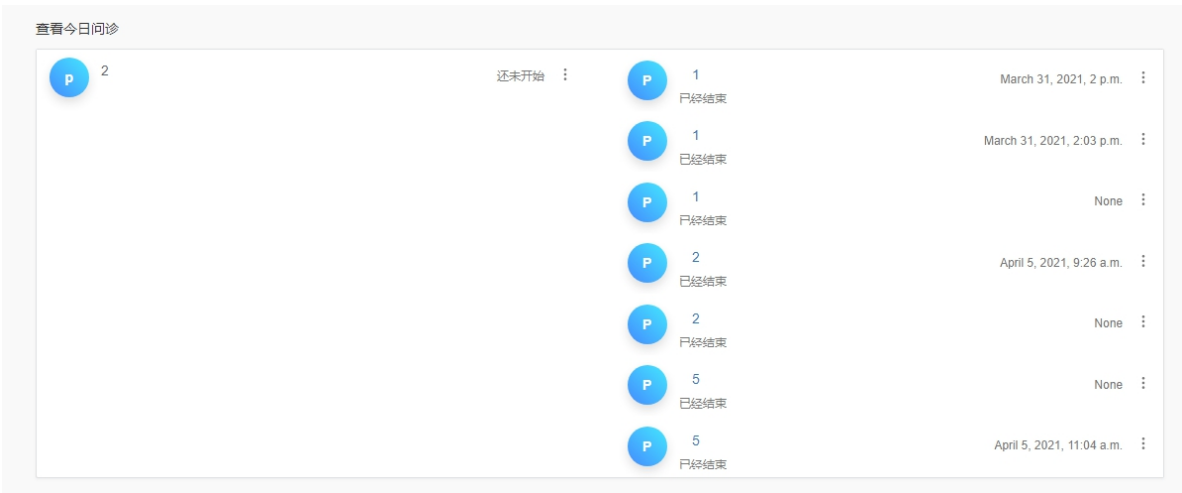


图 4-2 医生界面图

4.3 在线诊断功能的实现

在线问诊可以同患者交流，实现在线诊断。

在线问诊界面图如图 4-3 所示：

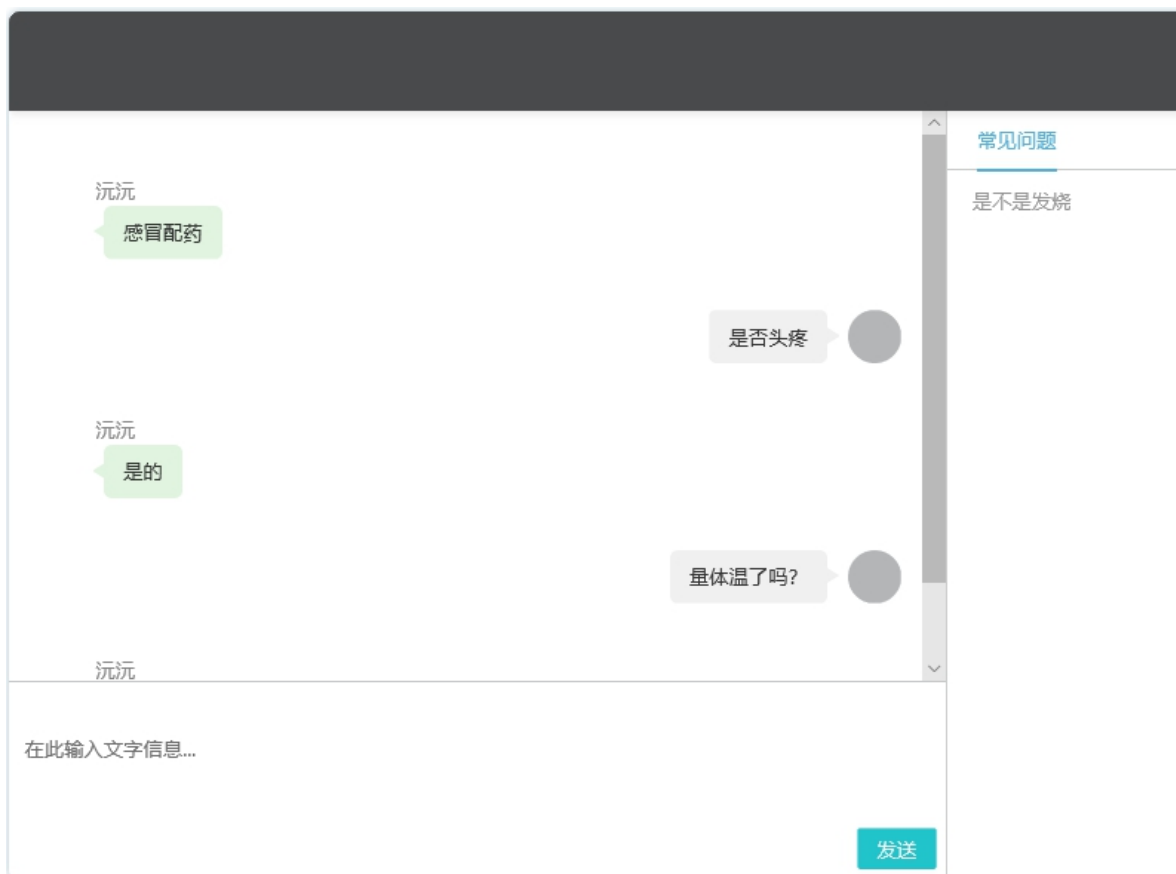


图 4-3 医生在线问诊界面

4.4 查询医生信息功能的实现

通过点击网站的医生大头像，就可以进入医生的主页，查看医生的医疗水平，学历等多方面信息。

查询医生界面如图 4-4、4-5、4-6 所示：



图 4-4 查询医生界面图



图 4-5 查询医生界面图



个人
简历

127.0.0.1:8000/info/doctor/1#experience

经历

学历

药剂科

1.武汉大学博士毕业;发表SCI8篇,总IF>25;2.发表核心期刊20篇;3.获得国家自然科学基金,湖北省自然科学基金,中央高校研究项目,武汉市基金,武汉大学教学改革项目

经历

副主任药师

擅长儿童/哺乳/孕期用药, 疫苗咨询。副主任药师, 医学硕士, 执业药师。头条号“胡钰药师”, 致力于用药科普, 欢迎关注。

擅长方向

感冒 狂犬病 流感 激素避孕 疫苗咨询 安全用药

武汉市普仁医院

图 4-6 查询医生界面图

4.5 电子处方功能的实现

该病患已获得处方，患者可以再历史中查看。

电子处方界面图如图 4-7、4-8 所示：

病状总结

给 的电子处方:

关闭

发送

图 4-7 电子处方界面图



8

吃点快克或者白加黑

April 5, 2021, 11:04 a.m. ⋮

图 4-8 电子处方界面图

5 模块和系统测试

5.1 医生登录功能测试

在医生登录过程中只有输入正确的账号密码才会登陆成功；否则会弹出“密码错误或无此账号”的错误提示。

登录错误提示图如图 5-1 所示：



图 5-1 登录错误提示图

5.2 医生注册功能测试

在医生注册过程中不得输入已注册的账户名称，否则会弹出“账户名已存在”的错误提示。账户输入错误提示图如 5-2 所示：



图 5-2 账户输入错误提示图

在医生注册过程中要输入两次相同的密码，否则会弹出“请输入相同的密码”的错误提示。如图 5-2 所示：

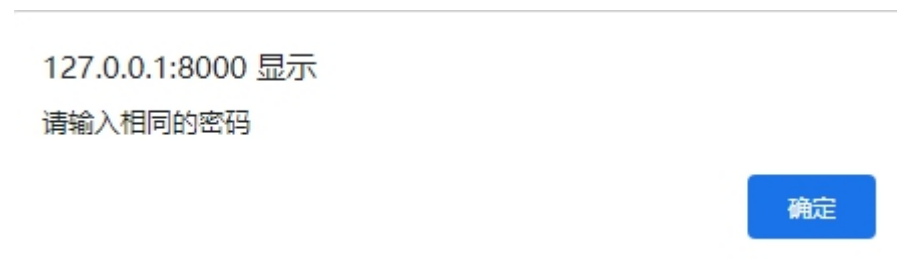


图 5-2

在医生注册过程中要输入 6 位以上的密码，否则会弹出“账户密码不可少于 6 位”的错误提示。如图 5-3 所示：



图 5-3

在医生注册过程中要填写正确的邮箱格式，否则会弹出“邮箱格式不正确”的错误提示。邮箱错误提示图如 5-4 所示：



图 5-4 邮箱错误提示图

结 论

毕业论文和设计是本科学习阶段一次非常难得的理论与实际相结合的机会，通过这次比较完整的在线医疗系统的设计，我摆脱了单纯的理论知识学习状态，和实际设计的结合锻炼了我的综合运用所学的专业基础知识，解决实际操作的能力，同时也提高了我查阅文献资料、实际操作解决问题的能力水平，而且通过这次做毕业设计，使我的能力得到了锻炼，经验得到了丰富，并且在意志力、抗压能力以及耐力都得到了不同程度的提升。这都是我们都希望看到的，同时也是毕业设计的目的所在。

在线医疗平台主要是建立在现代信息网络科学的基础上的，这种线上诊疗已经深深的影响了整个商业的领域，本文从来看理论和实践两个角度，采用 Python 技术和数据库技术，构建在线医疗平台，在线医疗平台的深入讨论。最初，我们需要学习很多和网站以及与在线医疗相关的各种知识和技巧，然后最对本系统的全面框架进行研究。本文做了大量的软件需求分析的工作，在设计的过程当中尽量做到与实际操作是一样的，所以本系统有非常强的使用价值，完全实现了一个实际的应用，尽可能的完成了该平台所需要的全部功能。

从不知道毕业论文如何写，到顺利如期的完成本次的毕业设计以及论文，使我充满了信心，同时也让我对本专业的发展前景充满了信心。但是该系统仍有很多不成熟，在系统设计过程中有许多技术缺陷存在。在设计的过程中也涉及到了很多自己无法解决的问题，主要通过找专业的网站和论坛来解决这些问题，对于圆满完成我的毕业设计，他们也贡献了很大一部分力量。

经过本次在线医疗平台的设计让我有了更多的信心，从中我也收获到了许多，在今后的学习和工作中，我会更加努力，以丰厚的成果和出色表现来回报自己，我相信在未来的道路上，我会走的更好、更远。

致 谢

经过一个学年的努力，我的毕业论文终于告一段落。在感到欣慰的同时，在此对帮助我、支持我的老师以及专家、学者、同学表示感激。

在本次论文设计过程中，感谢我的学校，给了我学习的机会，在学习过程中，老师从选题指导、论文框架到细节修改，都给予了细致的指导，提出了很多宝贵的意见与推荐，老师以其严谨求实的治学态度、高度的敬业精神、兢兢业业、孜孜以求的工作作风和大胆创新的进取精神对我产生重要影响。他渊博的知识、开阔的视野和敏锐的思维给了我深深的启迪。

本设计在邵彧老师的悉心指导和严格要求下已完成，从课题选取、方案论证到具体设计和调试，无不凝聚着邵彧老师的心血和汗水，在两年的本科学习和生活期间，也始终感受着导师的精心指导和无私的关怀，我受益匪浅。在此向邵彧老师表示深深的感谢和崇高的敬意。

不积跬步何以至千里，本设计能够顺利的完成，也归功于各位任课老师的认真负责，使我能够很好的掌握和运用专业知识，并在设计中得以体现。正是有了他们的悉心帮忙和支持，才使我的毕业论文工作顺利完成。

感谢所有授我以业的老师，没有这些年知识的积淀，我没有这么大的动力和信心完成这篇论文。感恩之余，诚恳地请各位老师对我的论文多加批评指正，使我及时完善论文的不足之处。

另外，更要提前向评审我论文的专家们表示感谢！由于本人时间、精力和能力有限，对本文的撰写和毕设的设计可能还存在不足，您们的批评和建议将成为我进一步学习的动力。

毕业在即，在今后的工作和生活中，我会铭记师长们的教诲，继续不懈努力和追求，来报答所有支持和帮忙过我的人！

参考文献

- [1] Journal.Exploring the freemium business model for online medical consultation services in China[J].Information Processing & Management,2021,58(3).
- [2] 吕涌涛,韩加亮.互联网医院建设实践探索[J].中国卫生信息管理杂志,2021,18(1):1-1.
- [3] Eric Matthes.Python 编程：从入门到实践[M].人民邮电出版社:北京,2016.
- [4] 刘长龙.Python 高效开发实战——Django、Tornado、Flask、Twisted[M].北京：电子工业出版社，2016.
- [5] 王珊、萨师煊.数据库系统概论（第 5 版）[M].高等教育出版社:北京,2014.
- [6] （美）施瓦茨等著.高性能 MYSQL[M].北京:电子工业出版社，2013.
- [7] 胡杨.Django 企业开发实战.北京：人民邮电出版社，2019.
- [8] （美）杰佛等著.Django Web 开发指南[M].北京：机械工业出版社，2009.
- [9] 张良均,王路,谭立云,等.Python 数据分析与挖掘实战[M].机械工业出版社:北京,2015:1-1.
- [10] 相万让.网页设计与制作（第 3 版）[M].人民邮电出版社:北京,2012.
- [11] 郑娅峰，张永强.网页设计与开发[M].北京：清华大学出版社，2016
- [12] 相万让.网页设计与制作（第 3 版）[M].人民邮电出版社:北京,2012.
- [13] 卢潇.软件工程.北京：清华大学出版社；北京交通大学出版社，2005
- [14] 杨文龙，姚淑珍，吴芸.软件工程[M].北京：电子工业出版社.
- [15] 张湘辉等. 软件开发的过程与管理[M]. 北京:清华大学出版社,2005.
- [16] 杜泽.2020 年在线医疗需求急增[J].中国信息界,2021.

附 录

医生登录的代码:

```
def doctorlogin(request):  
    uname = request.POST.get('uname')  
    pwd = request.POST.get('password')  
    if uname and pwd:  
        # Correct password, and the user is marked "active"  
        userList = doctor.objects.filter(d_account=uname, d_password=pwd)  
  
        if userList:  
            request.session['username'] = uname  
            request.session['is_login'] = True  
            request.session.set_expiry(24 * 60 * 60)  
            appointments = appointment.objects.filter(d_id=uname, is_done=0)  
            appointments_isdone = appointment.objects.filter(d_id=uname, is_done=1)  
            return render(request, 'domain.html',  
                          {'username': uname,  
                           'appointments': appointments,  
                           'appointments_isdone': appointments_isdone})  
        else:  
            return render(request, 'doctorlog.html',  
                          {'code': '密码错误或无此账号!', 'sign': True})
```

医生信息的代码:

```
# 医生信息  
def doctorinfo_view(request, did):  
    doctors = doctorinfo.objects.get(id=did)  
    data = {'id': doctors.id, 'name': doctors.d_name,  
           'local': doctors.d_local, 'department': doctors.d_department,  
           'hospital': doctors.d_hospital, 'phone': doctors.d_phone,  
           'title': doctors.d_title,  
           'experience': doctors.d_experience,  
           'education': doctors.d_education, 'skills': doctors.d_skills,  
           'mooto': doctors.d_mooto}  
    return render(request, 'doctor.html', {'list': data})
```

在线诊断的代码:

返回聊天页面数据

```
def chat(request):
    username = request.session.get('username')
    userid = user.objects.get(u_account=username).id
    userinfos = userinfo.objects.filter(u_account=username).last()
    doctor = doctorinfo.objects.get(d_department=userinfos.d_department)
    appointments = appointment.objects.get(u_id=userid, is_done=0)
    appointments.opentime = timezone.localtime(timezone.now())
    appointments.is_working = True
    appointments.save()
    data = {'id': appointments.id, 'time': appointments.opentime,
           'doctor_name': doctor.d_name,
           'reason': userinfos.u_description, 'department': userinfos.d_department,
           'doctor_id': doctor.id, 'age': userinfos.u_age,
           'gender': userinfos.u_gender, 'name': userinfos.u_name }
    return render(request, 'chat.html', {"list": data})
```

发送信息

```
def push_message(request):
    data = json.loads(request.body)
    appointmentid = data["appointmentid"]
    username = data["master"]
    message = data["message"]
    flag = data["flag"]
    plus_message = chats.objects.create(appointment_id=appointmentid,
                                         master=username, message=message, flag=flag)
    return HttpResponse(request.body)
```

获得聊天信息

```
def get_message(request):
    data = json.loads(request.body)
    appointmentid = data["appointmentid"]
    other_side = data["otherside"]
    flag = data["cflag"]
    flag = flag - 1
    print(flag)
    check_new_message = chats.objects.filter(appointment_id=appointmentid,
                                              master=other_side).count()
    if flag != check_new_message:
        new_message = chats.objects.get(appointment_id=appointmentid,
                                         master=other_side, flag=flag)
        flag = flag + 1
        data = {'newmessage': new_message.message, 'flag': flag}
        return HttpResponse(json.dumps(data, ensure_ascii=False),
                             content_type='application/json')
```

返回医生问诊页面

```
def dochat(request, aid):
    return render(request, 'dochat.html', {'id': aid})
```

页面无刷传参消息的代码:

```
var flag=0;

function SendMsg() {
    var text = document.getElementById("text");
    if (text.value == "" || text.value == null) {
        alert("发送信息为空, 请输入!")
    } else {
        $.ajax({
            url: '/ers/push/',
            type: "POST",
            data: JSON.stringify({
                appointmentid: {{id}},
                message: text.value,
                master: 'd',
                flag: flag
            }),
            success: function(data) {
                AddMsg('default', SendMsgDispose(text.value));
                text.value = "";
                flag = flag + 1;
            }
        });
    }
}

function SendMsgDispose(detail) {
    detail = detail.replace("\n", "<br>").replace(" ", "&nbsp;");
    return detail;
}
```

页面无刷接收消息的代码:

```

function get_chat() {
    $.ajax({
        url: '/ers/pull/',
        type: "POST",
        data: JSON.stringify({
            appointmentid: {{id}},
            cflag: cflag,
            otherside: 'p',
        })
    }, success: function(data) {
        cflag = cflag+1;
        console.log(cflag);
        AddMsg('{{username}}', SendMsgDispose(data.newmessage));
        return data.flag;
    });
}

```

页面无刷控制器的代码:

```

cflag = setInterval(get_stu_list, 1000);
function SendMsgs(text) {
    var inner = document.getElementById("text");
    inner.innerHTML = text;
}

```

电子处方的代码:

```

def updateill(request):
    ill = request.POST.get('ill')
    id = request.POST.get('info_id')
    print(id)
    change = appointment.objects.get(info_id=id)
    change.ill = ill
    change.save()
    return HttpResponse("done!")

```