**医患平台知情同意书电子化管理系统的设计与实现**

**摘要**

社会进步的同时，法律也会不断完善。义务教育的普及，教育水平的提高，都慢慢引导大众的法律意识也在提高，近年来，知情同意权越来越被人重视。知情同意权由知情权和同意权两部分组成。互联网的迅速发展，个人信息的承载形式也从纸质记录转换成了电子档记录，大数据时代来临，使得获取个人信息的方式也变得更容易，如果信息主体之外的第三人获得个人信息且进行不当操作，则会严重损害他人的权益，严重者危及人身安全和人格尊严。现实施的知情同意规则是保护个人信息的最主要手段，信息主体可以通过它来实现对个人信息的控制和自主决定。所以如何在信息化的时代，充分发挥互联网信息的管理，在保障安全的情况下完成知情同意书的签署管理工作是个人信息保护亟待解决的问题。

医疗领域中的医疗大数据是由无数个个人医疗数据组成的，个人医疗数据包括个人信息数据，即一个人可被识别的数据信息记载，如姓名、身份证号码、地理位置等身份隐私性信息，和个人医疗数据，即患者诊疗数据、电子病例、保险信息等医疗隐私性信息，两部分组成。“医疗大数据”通过个人基因、生物识别以及电子病历等数据，对患者信息进行统筹收集、利用和分析，并通过互联网平台共享、交叉检索形成信息数据。它是一种集数据搜集、传输、储存和共享于一体的综合化技术【5】。在大数据共享，用于商业用途的过程中，如果未对患者的个人信息进行有效脱敏，势必会对权益主体的隐私无意泄漏。所以无论患者信息的权属在个人还是在医疗机构，在共享过程中都要先获取信息主体的知情同意。

通过对PHP基础和PHP框架Laravel的学习，本系统计划实现一种基于信任机制的电子知情同意管理，将授信的医疗项目先加入该系统中，然后在各项目下可以创建知情同意书，再将知情同意书发送到患者端，患者在自己对应的医疗项目中查阅和签署知情同意书，签署结果记录在系统中，完成知情同意书的电子信息化，简化传统签署过程的繁杂，避免错漏，并且可以及时更新知情同意书的版本，方便患者和医疗机构对知情同意书的管理和追溯。

1. **关键字**：知情同意 PHP laravel mysql 系统设计
2. **前言**
3. 国内外研究现状：

在国外，知情同意的概念最早是在医学领域和生命伦理学领域提出的，然后才逐渐发展到个人信息领域。在互联网时代，国外的学者渐渐对个人信息领域知情同意规则有了不同的见解。麦卡锡(Maccarthy)提出在该规则下，信息主体的同意是进行信息处理和使用的合法性基础；沙哈尔(Shahar)和施耐德(Schneider)认为该规则易于实施，能够替代一般的监管行为；班贝克(Bamberger)和穆里根(Mulligan)则认为由于该规则具有一定的灵活性，因此可以根据信息主体的倾向不同而采取不同的实施细节【3】。然而也有消极的言论，索洛夫(Solove)详尽分析了同意规则所存在的两个困境，即信息主体的认知问题以及隐私保护的结构问题；学者鲁宾斯坦(Rubinstein)则更进一步指出，在大数据时代该规则早已是走向瓦解，没有任何可以修复的余地。

国内学者在知情同意规则方面的研究起步较晚，但是我国的信息技术产业发 展迅速，所以在个人信息保护方面暴露的矛盾和问题也更突出，目前学者们 基本持两种态度，一方面对知情同意规则表示否定和质疑，认为该规则已经 被虚化成了人和机器之间无意识的交互操作，信息主体有可能并不知情且被 迫同意，只能做出无可奈何的选择，已经失去了其原本的合理性，不那么必 要也不那么真实。另一方面还是比较积极的，大多数人认为知情同意规则还 是有其存在的理论基础和现实基础，是人们对个人信息支配的体现，是不可 或缺的一部分，只是现在存在一定的局限性，并没有和互联网大数据实现完 美的契合。

笔者认为，持消极言论的学者忽略了知情同意规则背后的价值理念，尽管所描述问题是不可避免的必然性后果，但是我们应该做的是通过技术手段让其更合理合法化，而不是彻底摒弃它。更重要的一个因素是，在医学和生命伦理学的领域，该规则的存在对保障患者的权益影响颇深，是保护研究对象的重要手段。计算机科学的发展带动医疗模式的转变，各国临床实验和科研伦理相关法律法规的完善，知情同意书有变得越来越复杂冗长的趋势，势必会带动知情同意电子信息化。2016年美国食品和药品监督局（Food and Drug Administration,FDA）已将电子知情同意定义为：有可能使用多种电子媒介，包括文本、图表、音频、视频、收听广播、互联网、生物识别设备和读卡器的电子系统和流程，以便传达与实验相关的信息并获得记录知情同意的过程，并对包括研究者、申办方及伦理委员会关于电子知情同意的一些疑问进行了解答。【4】

2、本课题研究目标：

在对知情同意规则转型的国内外研究中，初步确定了本课题的研究目标：实现一个医患平台线上管理知情同意书的签署管理系统，包括四个模块，分别为：授信医疗项目管理模块，知情同意书管理模块，知情同意书下发模块和患者签署模块。系统整体采用MVC架构模式，借助于PHP语言实现后台逻辑，辅以laravel框架和laravel-admin工具，搭配mysql数据库进行后台数据管理。系统环境引入开源应用容器引擎Docker。

1. 论文组织结构：

本文有六个章节，第一部分为前言，简单介绍了本系统研究的背景和出发点以及系统的总体目标。第二部分为相关技术的介绍，本系统所涉及的技术均为笔者为该课题新学的知识，所以篇幅略长。第三部分是该系统详细的需求分析和包括经济技术法律三方面的可行性分析。第四部分为系统的主体部分，即该管理系统各个模块功能的设计与实现，包括数据库部分的设计等。第五部分是该管理系统的成果展示，以及总结与展望，下为详述。

三、系统相关技术介绍

由于笔者实习工作的原因，所在岗位用PHP语言进行开发，所以本系统技术路线的选择也是基于PHP语言，使用PHP语言的Laravel框架，搭配使用优秀的数据库管理系统MY SQL来实现后台数据管理，使用Laravel框架的一个工具Laravel-admin完成前端页面的搭建。系统的环境采用环境容器引擎Docker来搭建，在其中配置nginx等各个服务。系统总体采取MVC模式架构，以下介绍笔者对各项工具的认知。

1. **PHP**

PHP语言是一种专注于web开发的语言，而且是免费开源的，它原始的名字是personal home page 的缩写，现在已经更名为了“PHP：Hypertext Preprocessor”,意为超文本预处理器。

它是一种嵌入式脚本语言，类似于ASP语言，可以灵活的嵌入到超文本标记语言的脚本语言中，在服务器端直接运行，进而转换成HTML语言，组成HTML动态网页的一部分。PHP语言由于它开源的特点，发展至今功能已经很全面，操作也并不复杂，还可以与mysql，oracle，microsoft sql server等主流数据库进行交互，因为它对数据库领域的丰富支持，也使得它拥有了众多的拥护者。同样开源的Mysql数据库和Apache服务器软件与之成为黄金搭档，搭配Linux成一套LAMP来搭建动态网站的技术栈体系。

PHP语言的底层语言是c语言，对于开发人员来说是极容易上手的，而且它有很强的可移植性，即不需要做任何修改就可以在各服务器平台之间进行移植。PHP的执行过程是通过浏览器向后台服务器发送请求，后台服务器通过解析处理，访问数据库，并把访问结果返回给用户。在整个执行过程中系统资源消耗量少，在解释执行PHP脚本时，不会增加额外的资源消耗，使得PHP执行效率非常高。【2】

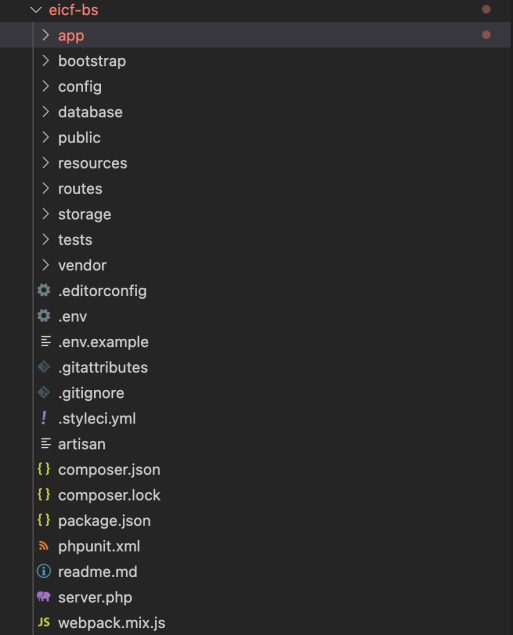
PHP在拥有众多优点的同时当然也有它的缺点，它是一种弱类型的语言，相对来说很灵活，但是减少了对开发者的编程约束，每个人的代码习惯不同，写出来的码各有不同。而PHP采用的又是解释性运行机制，所以在解析过程中可能会出现一些出乎意料的解析问题。每一个PHP页面被解释执行完以后，所有的相关资源都会被回收，在语言级别上来讲没有办法做到让某个对象常驻内存，而所有的变量都是页面级的，页面执行完毕后所有的变量都会被清空，不像JSP中，java bean的scope值有四种：page，application，session，request，分别对应四个生存期，页面，程序，回话，请求。

1. **Laravel**

PHP语言由于它简单高效且可以跨平台等优点，被各大企业采用，来开发web网站，比如众所周知的新浪，网易，腾讯，还有国外社交平台Facebook。于是在不断的发展中，诞生了许多PHP的开发框架，以及自己的依赖包管理器composer现在主流的PHP框架例如，Laravel , Symfony , yii ,还有国内的ThinkPHP,这些框架都是基于MVC模式为基础开发的，实现了程序的组件化和代码规范化。

本课题中使用了laravel框架，它是出现最晚的框架但是却是当下使用范围最广的框架。Laravel 作为一款优雅的 MVC 结构 的 PHP 框架，其框架逻辑清晰，代码结构简单，非常适合进行团队合作开发和代码的重用，同时随着框架版本的升级，可非常简单的对项目进行移植，使得项目的整体性能能够随框架的升级得到同样的提高。【1】

laravel框架基本结构如下图所示：



结合本系统的目录介绍一下这个框架的结构，其中：

1. app

该目录包含了项目下的核心代码，用户编写的代码绝大多数会放到此处，包含控制器，中间件，模型。

1. Bootstrap

该目录下的文件是框架的启动文件和自动载入配置文件，cache文件下存放了路由和服务的缓存文件，为框架提升性能。

1. Config

项目的配置文件，比如数据库的连接配置，第三方依赖文件等等。

1. Database

该目录下存放数据库的迁移文件和填充文件。

1. Public

该目录是站点目录，存放项目的入口文件index.php，系统的静态资源文件（css，image，upload），以及后期使用的外部静态文件（JS，css，图片等）。

1. Resources

该目录下存放一些原生资源文件，LESS，SASS，CoffeeScript，以及视图文件和一些本地化的文件。

1. Routes

该目录下存放该应用定义的所有路由，包含web.php、api.php 和 console.php。

1. Storage

该目录下的logs是应用的日志文件，framework存放框架生成的缓存文件（如session缓存），另外本地上传的文件也可存放在该目录下。

1. Tests

该目录下有一个开箱即用的自动化测试示例，测试类均要求以Test开头，然后用php vendor/bin 命令进行测试。

1. Vendor

存放composer依赖的第三方类库文件。

1. .env

系统相关的环境配置文件。

1. Artisan

脚手架文件，artisan命令用来自动生成控制器、模型等文件。

1. Laravel-admin

laravel-admin是一个laravel[框架](https://so.csdn.net/so/search?q=%E6%A1%86%E6%9E%B6&spm=1001.2101.3001.7020" \t "/Users/sanqi./Documents\\x/_blank)的一个后台管理系统的简单开发版，利用laravel-admin可以快速构建起网站后台管理系统。不用编写前端页面，仅仅需要几行代码即可帮助我们搭建一个CRUD后台。

首先下载并安装好laravel项目，设置正确的数据库连接，然后就可以安装laravel-admin插件：

composer require encore/laravel-admin:1.\*

接着发布资源：

php artisan vendor:publish --provider=”Encore\Admin\AdminServiceProvider”

执行完该命令会生成一个配置文件config/admin.php，安装地址，数据库连接，表名等都可在这个配置里进行修改。

最后一步完成安装：

php artisan admin:install

启动服务后，在浏览器打开 http://localhost/admin/ ,使用用户名 admin 和密码 admin登录。

1. MVC设计模式

laravel框架是基于MVC软件设计模式架构的，MVC模式即用模型来操作数据，用视图来展示数据，而控制器用于分发指令，实现model和view的有效分离，又通过controller实现model和view的同步更新，是当下比较主流的一种设计模式。laravel框架使用路由来将各个请求分发到指定的控制器，让控制器去协调model操纵数据库和view展示数据，使得应用的逻辑结构更加清晰，代码可读性更好，系统的可扩展性和可维护性也得到提升。

1. Model

在模型层中的代码封装了对数据持久层的操作逻辑和业务逻辑，控制层会根据业务逻辑调用模型层的代码对数据库的数据进行操作。

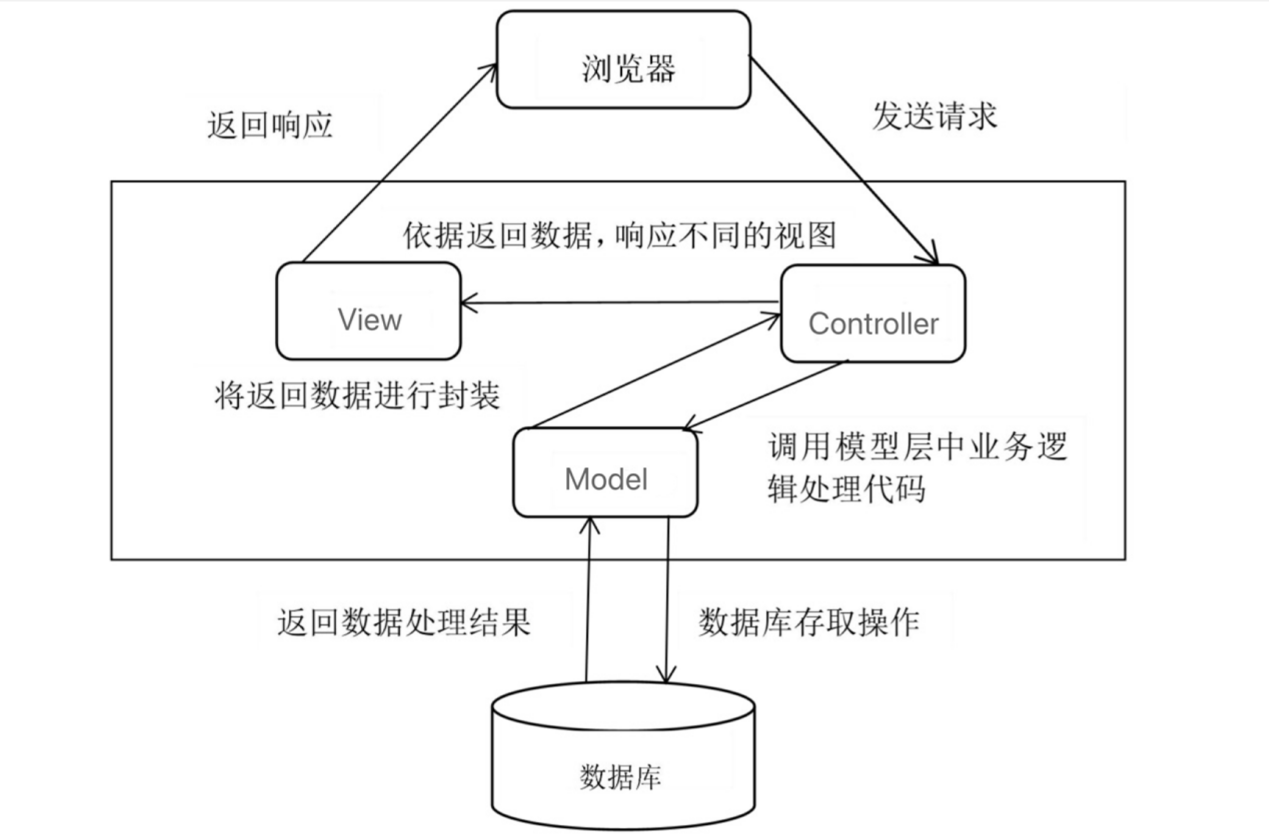
1. View

视图层提供程序入口给用户。在发送数据请求后，相关组件通过对接收数据的处理以及对处理结果的封装等操作，依据不同的返回类型，对用户进行相应的视图应答【6】。视图层是人和机器之间传递消息的通道，所以要尽可能为用户提供一个简单灵活，清晰舒适的视图界面，以提高用户的使用体验。

1. Controller

控制器从视图界面获得用户提交的请求数据，然后将请求转发至对应的服务器端程序中，调用Model层对数据进行处理，再将处理结果响应到用户端。

MVC设计模式这种分层设计的思想，将程序的逻辑清晰分开，各个层次的代码各司其职互不干扰，开发效率得到有效提升。模块划分降低了程序耦合度，假如视图层的代码需要进行更新或者扩展，并且与后台数据不产生逻辑关联，那么可以只编译视图层代码，控制器和模型层的代码不必进行二次重复编译，提高了系统的鲁棒性。而且该模式极大程度提高了代码可复用性，在原有的模型层中加入少量的业务逻辑代码，就能避免大量重复而轻松完成功能扩展，完善程序。而在系统实现的过程中，也可以将开发人员根据该结构进行前后端分离分工，互不干扰，同时开发，缩短工期。



Model+view+controller之间逻辑关系

1. Docker技术

Docker是一个开源的应用容器引擎，通俗来讲就是创建容器的工具，本身并不是一个容器。docker客户端负责发送请求，服务器端负责构建容器，分发工作。借助Docker容器技术，运维开发人员不需要在进行项目部署时，搭建复杂的环境配置，只需要将开发好的程序打包放入Docker容器中，就完成了项目的发布。

传统的虚拟机技术启动时间过长，长达数十分钟，而Docker启动时间却达到了毫秒级别，极大程度缩短了程序的部署时间。Docker对系统的资源实现了相对程度的最优化使用，在配置相同的主机上，Docker可以同时运行更多数量的应用。因为Docker是开源的，所以其团队提供了很多高质量的官方镜像，都是可以免费下载使用的。如果有特殊需求，也可以以官方的镜像作为基础模版进行二次更改。同时，得益于分层存储等关键性技术的运用，使得应用复用、应用更新维护等也变得更加便捷降低了更新维护和扩展的成本。

1. Mysql

在web应用开发中，mysql当下已经成为最流行的关系型数据库。它最开始是由瑞典的MYSQL AB 公司开发的，2008年被SUN公司收购，次年SUN公司又被Oracle公司收购，所以目前属于Oracle公司。关系型数据库将数据存放在不同的表中，而不是全都堆积在一个大仓库中，很大程度上提高了增删改查的灵活性和效率。MYSQL同PHP一样也是开源的，虽然它是一个轻量级的数据库管理系统，但是它完全可以支持目前绝大多数web应用的数据管理，使用的也是标准的SQL语句，支持多种语言，为它们提供了API接口，尤其和PHP达到近乎完美的契合。MYSQL的核心程序采用完全的多线程编程，实现了快速且稳定的基于线程的内存分配系统，可以持续使用还不必担心其稳定性，最大化了CPU的利用率。

1. Nginx

Nginx是一个轻量级的web服务器/反向代理服务器，同时也是一个很优秀的电子邮件代理服务器，在BSD-like 协议下发行，底层逻辑也是用c语言编写的，在占用更少的内存和资源的条件下依旧能很好的完成高并发的任务。

代理服务器是用户和目标服务器之间的中介服务器，用户无需知道目标服务器的地址即可通过反向代理服务器获得目标服务器的资源，作为web服务器的前置机以降低网络和服务器的负载均衡，提高web访问效率。普通的代理服务器仅支持内部网络对internet发出的请求，而反向代理服务器可以支持外部网络的请求访问内部网络。

nginx用线程处理用户的请求，而线程是共享内存的，因此只需要开启少量的进程，多个线程就可以共享进程的内存。在同类型的网页服务器中它的并发能力明显优于apache，在处理静态文件时，处理性能能比apache高三倍以上。作为一款负载均衡的服务器它可以支持七层负载均衡，负载能力也优于apache。

nginx的启动容易，运行简单，更优秀的是不出意外它可以做到7\*24小时不间断运行，几个月都不用重新启动，在不间断服务的情况下还能进行版本更新升级操作。

但是区别于apache的同步进程模型，nginx是异步的。apache一个连接对应一个进程，nginx多个连接对应着一个进程，这个连接可以达到万级别，所以它的稳定性不如apache，当一个进程死掉时，会影响到多个用户的使用。

nginx的优点众多但是它的技术成熟度还达不到apache的程度，相对来说bug较多，现在多用于处理静态和反向代理。

1. **需求分析和可行性分析**

可行性分析：

1. 法律可行性

技术栈中使用的工具均为开源免费，符合BSD开源协议，因此在法律上可行。

1. 技术可行性

目前很多网站都开始用Nginx了，并发数要好一些，淘宝网甚至根据Nginx专门制作了一个开源项目Tengine，大量的产品使用保证了技术栈的稳定以及问题的收敛，因此在技术上可行。

1. 经济可行性

我国针对互联网产品的各类法规越来越多，但是大量小微企业并没有人力可以投入进行开发知情同意的功能，因此通过调用本产品可以低成本的对企业法律合规进行升级，并且并发数以及数据库大小都不大，成本低效益高，因此经济上可行。

本系统将实现一个知情同意书的线上下发和签署并对医疗项目和知情同意书进行管理的平台，共分为四个模块，

开发的基本原则

1、

2、

3、

4、

系统开发总体需求：

知情同意书在医疗行业中是患者表示自愿进行医疗治疗的文件证明，患者签署知情同意书是充分尊重患者权利，最大限度降低医疗风险与避免不必要的医疗纠纷的必要措施。目前，知情同意书多以纸质文件的方式进行签署，模式较为传统，签署时容易产生错漏情况；签署后，纸质文件不易保管，容易遗失。随着医疗卫生信息化的不断发展，对于知情同意书的便携签署与高效管理是相关领域人员亟待解决的问题。针对相关技术中传统纸质版知情同意书易错漏、易丢失、管理困难的问题，目前尚未提出有效的解决方案。本设计的主要目的在于提供一种基于信任机制的电子知情同意管理系统及方法，以解决相关技术中传统纸质版知情同意书易错漏、易丢失、管理困难的问题。为了实现上述目的，第一方面，本设计提供了一种基于信任机制的电子知情同意 管理系统及方法，包括：医疗项目模块、知情同意书模块、知情同意书下发模块、患者签署模块。

各个模块功能需求：

1、医疗项目模块：注册授信医疗项目

2、知情同意书模块：创建和编辑知情同意书，并将知情同意书发送到知情同意书下发模块。每次创建一个知情同意书新版本，按照每个授信医疗项目 对应保留授信项目下全部版本的知情同意书，并且赋予版本编号。知情同意书是通过富文本编辑器创建的富文本(HTML)格式文件。

3、知情同意书下发模块：将知情同意书下发到患者能够访问的应用端。应用端包括APP程序、微信小程序与web应用。

4、患者签署模块：患者通过应用端加入授信医疗项目，查阅下发的知情同意书并通过点击同意按钮签署知情同意书。患者在签署前已完成系统用户权限认证，无需手写签名。知情同意书的版本会不定期更新，患者登录后签署最新版本知情书同意书。患者签署模块将签署后的记录反馈到知情同意书模块中进行保存。

功能需求（各个模块需求）

数据库需求

安全性需求

预期实现目标：1234、、、功能。

从管理者角度的功能分析。权限（用例图）、、、

系统流程图

1. 系统设计（分了几个模块）

系统设计总体目标，总体架构图

[0021] 将医疗项目注册成为授信医疗项目； [0022] 在每个授信医疗项目下，创建或者编辑一个或者多个知情同意书； [0023] 将知情同意书发送到患者能够访问的应用端； [0024] 患者通过应用端加入授信医疗项目，查阅并且签署知情同意书。 [0025] 所述患者通过应用端加入授信医疗项目，查阅并且签署知情同意书，步骤如下： [0026] 患者打开应用端； [0027] 患者登录电子知情同意管理系统； [0028] 患者申请加入某个医疗项目； [0029] 患者阅读知情同意书； [0030] 患者判断是否同意知情同意书； [0031] 若同意则签署最新版本知情同意书并且进入所述医疗项目，若不同意则返回登录 界面。 [0032] 所述签署最新版本知情同意书，若患者签署当时最新版本知情同意书后，所述医 疗项目再次发布更新后的最新版本知情同意书，则需要患者重新签署，流程如下： [0033] 授信医疗项目再次录入知情同意书内容； [0034] 电子知情同意管理系统自动生成最新版本知情同意书； [0035] 患者重新签署新版知情同意书。

各个模块功能设计

数据库设计：数据库是管理系统的基础，合理设计数据库中各个数据表的结构是系统设计考虑的关键点。本管理系统重点涉及到四个数据表，结构如下图所示：

医疗项目模块，

知情同意书模块：所述知情同意书是通过上传一份包含知情内容的图片格式文件。 所述知情同意书下发模块以JSON格式API的方式提供知情同意数据下发接口。

知情同意书下发模块，

知情同意书下发模块是对第三方或自有产品开放的api，每当一个新用户进入或者知情同意版本更新之后，都会通过api收到一个知情同意签署的通知以及内容。

患者签署模块

患者签署知情同意后，app或者小程序会通过接口形式将用户数据传回服务端，服务端对患者的签署行为进行记录。

设计原则：系统性，实用性，安全性，规范性、、、

体系结构：B/S模式，MVC三层架构模式

可以画一个类似的系统结构图。

数据处理流程：数据采集，数据匹配和分析，数据发送，数据接受，数据存储，

接口设计：内部接口，外部接口，用户接口？？（不知道这个有必要没有）

各个模块的功能架构，功能结构图

1. 系统实现（典型模块的实现，主要代码）

系统关系类图

数据库的E-R图：

各种数据表

成果展示：代码示例，界面示例。

1. 系统测试与维护

测试项目名称以及测试内容和测试结果。

**5.2.1 测试方案**

（1）用户名合法化判定

在用户注册时，系统会检查用户名是否合法，本系统规定用户名只能使用字母、数字和下划线。通过如下程序进行判定，当用户名出现非法字符时就报错退出。

/\*\*

\* Get a validator for an incoming registration request.

\*

\* @param array $data

\* @return \Illuminate\Contracts\Validation\Validator

\*/

protected function validator(array $data)

{

return Validator::make($data, [

'name' => ['required', 'string', 'max:255'],

'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],

'password' => ['required', 'string', 'min:8', 'confirmed'],

]);

}

/\*\*

\* Create a new user instance after a valid registration.

\*

\* @param array $data

\* @return \App\User

\*/

protected function create(array $data)

{

return User::create([

'name' => $data['name'],

'email' => $data['email'],

'password' => Hash::make($data['password']),

]);

}

（2）知情同意的输入测试

本系统中，管理员在添加知情同意的详细信息时，对每个字段的值都有其类型的要求，如果类型不符合则无法创建，应用以下代码实现。

/\*\*

\* Make a grid builder.

\*

\* @return Grid

\*/

protected function grid()

{

$grid = new Grid(new Inform());

$grid->column('id', \_\_('ID'));

$grid->column('name', \_\_('名字'));

$grid->column('desc', \_\_('Desc'));

$grid->column('appid', \_\_('对应业务方ID'));

$grid->column('app', \_\_('对应业务方'));

$grid->column('project', \_\_('对应项目'));

$grid->column('type', \_\_('知情类型'));

$grid->column('category', \_\_('知情分类'));

$grid->column('content', \_\_('内容'));

$grid->column('version', \_\_('版本'));

$grid->column('status', \_\_('状态'));

$grid->column('created\_at', \_\_('创建时间'));

$grid->column('updated\_at', \_\_('最后更新时间'));

return $grid;

}

（3）知情同意信息展示测试

管理员在添加知情协议信息后，再打开详情页时应展示与创建时相同的内容，由以下代码实现。

/\*\*

\* Make a show builder.

\*

\* @param mixed $id

\* @return Show

\*/

protected function detail($id)

{

$show = new Show(Inform::findOrFail($id));

$show->field('id', \_\_('Id'));

$show->field('name', \_\_('名字'));

$show->field('desc', \_\_('描述'));

$show->field('appid', \_\_('对应业务方ID'));

$show->field('app', \_\_('对应业务方'));

$show->field('project', \_\_('对应项目'));

$show->field('type', \_\_('知情类型'));

$show->field('category', \_\_('知情分类'));

$show->field('content', \_\_('内容'));

$show->field('version', \_\_('版本'));

$show->field('status', \_\_('状态'));

$show->field('created\_at', \_\_('创建时间'));

$show->field('updated\_at', \_\_('最后更新时间'));

return $show;

}

（4）新增知情同意弹窗测试

在知情同意内容版本更新之后，调用接口时应正确返回新版本知情同意内容。

1. 总结与展望，结论，

在知情同意系统中，只能查看、查询、删除、修改并且导出用户、教练和项目的基本信息。但在现实生活中，用户可以在导出后修改甚至删除知情信息。此设计无法执行这些任务。其次，在现实生活中，知情同意与项目之间可能存在许多联系，这些联系在系统设计中有些是无法体现的。这就是我认为这个系统设计上的缺陷。今后我们将积累更多的知识来解决这些问题，努力实现上述任务。

1. 图目录
2. 表目录
3. 参考文献

【1】崔楠.基于PHP Laravel框架的HTML5广告页面制作系统的设计与实现 （D）西安电子科技大学 2015

【2】马海苓.基于PHP和MYSQL技术的网络教学平台系统的设计与实现（D）齐鲁工业大学 2016

【3】杨民来.大数据时代知情同意规则研究 （D）江西财经大学 2021

【4】乔雯俐，王思洁，赵心清，李雪宁，杨梦婕 电子知情同意的发展现状与伦理审查要点分析 （A）复旦大学附属中山医院伦理委员会

【5】王欢.个人医疗数据利用中知情同意的伦理问题研究 （D）天津医科大学 2020

【6】柯万超.基于Java Web的高校科研成果管理系统设计与实现 （D）阜阳师范学院 2021

1. 致谢

落笔于此，意味着大学生活也快画上句点。回首间仿佛还能看见，初入校园时大家十八岁的青涩脸庞。岁月匆忙，青春四年，疫情三年，尽管求学之路遇到过诸多困难，但大家都不曾停下脚步，依旧砥砺前行。感谢相遇，而离别是为了更好的重逢，大家都重新出发了，向着更光明的未来。

首先感谢指导我论文的杨帆老师。由于就业选择和疫情的影响，我没能回学校参加实习，且没能在老师提供的选题中完成毕设，而是在实习单位进行选题。在前期沟通过程中杨老师远程进行指导，不仅同意了我校外实习做毕设的请求，还为我在职业选择的困境中答疑解惑，指点迷津。在后期的选题，研究过程中都给予悉心指导，帮助我开拓思路，提出改进方案。在杨老师的指导下完成了该毕设论文。杨老师对学生认真负责，关怀备至，学识渊博，学术态度端庄，对杨老师的感激和景仰之情无以言表。

感谢指导我毕设的老师兼领导黄宇升经理，组长曹庆贺，以及给予我莫大帮助的同事梁燕鹏，杨士凡，马潇等。职业生涯的第一步，能遇见你们是我这前半年最幸运的事。无论工作，毕设，还是生活方方面面你们都无时无刻不在关怀着我。

感谢大学生涯里遇到过的每一位老师。能得到您们的栽培是我荣幸之至。曾几何时我也想成为一名和您们一样的老师，只可惜世事变化无常，不能坚持自己的梦想是一种遗憾，但在我心里，老师永远都是我们人生路上的明灯，迷途中的引路人。虽然没有再续学术之路，但路漫漫其修远兮，吾将上下而求索。我会谨记老师的教导，努力成长。再次感谢老师们的费心教导。

最后我将感谢我的家人为我倾注的半生心血，我今天的一切都是你们倾尽所有换来的。

感谢我的同学和朋友们一路的陪伴和相互扶持，让我的人生丰富多彩。希望未来的道路上，我们都在，互相帮助，一路同行。