

南京航空航天大学

大学生创新训练计划项目

研究总结报告

项目名称：                    计算机信息传输系统

项目负责人：  谢骏鑫                    学号：162110132

项目成员：      谢骏鑫                    学号：162110132

                    林龙河                    学号：162140128

                    蔡承臻                    学号：162150226

学号：

指导教师：      陈兵                    单位：计算机科学与技术学院

研究时间：      2021 年 11 月～ 2022 年 04 月

南京航空航天大学 教务处制

# 大学生创新训练计划项目研究总结报告

## 撰写要求

1. 研究总结报告主要包括项目的研究意义及价值、具体项目研究过程（包括研究内容与方法）、结论和成果、项目的特色或创新点、应用前景、收获和体会、尚存在的问题及进一步研究的设想等内容。

2. 研究总结报告正文格式：首行缩进 2 个字符，五号宋体，固定行间距 20 磅。

3. 研究总结报告字数不低于 5000 字。

4. 一律使用 A4 纸，双面打印，一式一份。

5. 论文中采用的术语、符号、代号全文必须统一，并符合规范化的要求。论文中使用新的专业术语、缩略语、习惯用语，要加以注释。国外新的专业术语、缩略语，必须在译文后用圆括号注明原文。

6、图、表要有自明性，即只要看图和图题而不阅读正文，就可理解图意。图位于文中表述之后，编号由“图”和阿拉伯数字组成，阿拉伯数字由前后两部分组成，中间用“.”号分开，前部分数字表示图所在章的序号，后部分数字表示图在该章的序号，例如“图 1.10”、“图 2.15”，图题置于图的编号之后，图的编号和图题要置于图下方的居中位置；表的编号方法同图的编号方法相同，例如“表 1.6”、“表 2.3”，表的编号和表题要置于表上方的居中位置，如某个表需要转页接排，在随后的各页上要重复表的编号，编号后跟表题（可省略）或跟“（续）”，如表 1.2（续）。续表均要重复表的编排。

# 目 录

一、研究意义 .....	1
1.1 .....	1
二、项目研究过程 .....	1
2.1 .....	1
2.2 .....	6
三、项目创新点 .....	7
四、应用前景 .....	7
五、收获和体会 .....	8
六、展望 .....	9

# 一、研究意义

## 1.1

随着机器学习用途的不断扩大，越来越多的机器通过 GPU 卡进行长时间计算，连续运行的计算机可能会出现 CPU 温度高、负载较大等情况，特别严重的会引起火灾等事故，因此需要对计算机关键设备的温度和载荷进行实时监控，一旦超过阈值需要进行报警。定时采集的数据传输到服务器进行存储。在一些场合，可以说，在大多数场合下，CPU 过载或者温度过高都是应该极力避免的，“殃及池鱼，祸及城门”，如果没有智能监控并阻止这种情况发生的系统存在的话，那在当今的计算时代下，后果是不堪设想的。所以，我们开展了这项研究，旨在利用计算机的自动和智能及稳定强大的计算能力将监测到的 CPU 温度和设定阈值进行对比并且能够将此类信息有效地传输出去。

# 二、项目研究过程

## 2.1.研究内容与方法：

我们的研究内容大致分为以下三点：

1) 首先登记需要监管的计算机相关信息，如计算机编号、型号、存放地点、设备管理者姓名、设备管理者手机等相关信息，并存储在服务器的数据库中，这里希望采取基于 WEB 的界面开发，也就是数据库的可视化操作设计。

2) 开发一个计算机驻留的小程序，通过系统调用，获得计算机 CPU 相关温度、负载等信息，这一点到这，并不是我们这个项目研究的方向，但它是整个研究背景下的一个重要的系统调用功能，我们的研究与设计的重点是放在后面这一句话上面，能够基于 socket 或者数据库连接存储到服务器数据库中，我们研究的方向就是利用 Socket 来实现信息的传输和交互并且能够和后端数据库紧密地绑定在一起，起到良好的数据保护和更新功能。

3) 一旦检测到温度超过设定的阈值，则给设备管理者发送信息进行告警。这一块也是我们研究的内容之一，判断温度是否超过阈值是信息传输过程需要进行的操作和程序，而给设备管理者发送报警信息是信息传输的范畴，这些都是我们研究的重点内容。

我们采用的整体研究方法是基本的软件工程设计中自顶向下设计的方法论，我们先分析研究内容需求，思考设计开发方法，选择开发语言，确定开发平台，完善开发文档，进而确立了小组和分工，进而进行开发设计。这是整体上的，接下来我会进一步细化，更加细节地

讲述这个项目研究过程中的具体开发过程和方法。

1) 第一个研究重点是可视化数据的登记输入保管这一板块,我们确立使用 Web 开发来实现。在进一步考虑如何进行 Web 开发的时候,我们遇到一点困惑,选择什么样的平台,利用搜索引擎,我们得到了各种各样的答案,有什么笼统的前端开发,什么 JavaScript、CSS、HTML 等等相关的前端开发语言,但这对于大一刚起步的我们来讲,即使是初高中已经接触了不少时间的编程了,显然是不打算尝试新的语言来开发的,从现在回头去看,其实当时也完全没有必要有这个惧怕,毕竟这就是一个学习的过程而已嘛。于是我们就去寻找有没有能够直接基于我们已经了解或者掌握的语言来开发,最后我们找到并且确立了 Python 的 Django 框架可以用来简单地部署一个 Web 界面和网站,于是我们第一个板块的整体框架就确定下来了,既然是要和数据库交互,这个我们自然而然地选择了广泛使用且易上手的 MySQL 数据库了,其中 Django 框架开发网页被分派了给小组的林同学完成,而自己负责整个项目的数据库部分,事实上,虽然有了分工,但作为负责人和整个项目的代码重构和部署的人,每一块知识自己都接触了不少。

2) 第一个板块的确定其实也间接地让我们确立了我们开发所使用的语言,那就是 Python,作为一个拥有极大的开源性和强大的三方模块库的编程语言,这门语言确实实给我们的开发和功能设计带来了极大的便利。在进行 Django 部署开发网站的过程中,我先是找到教程开发了一个 Django 框架默认的网站样式,但这一过程也出现了不少麻烦,环境的部署过程中出现了很多让人头大的红色报错信息,于是我不断地利用搜索引擎,进行各种尝试和调试,最终也是搭载出来了一个以 MySQL 为数据库的默认 admin 界面的 Django 网站,当然直接用 admin 界面进行数据库的管理和数据的增删改查是极不负责的开发行为,admin 界面就是 Django 框架开发网站不进行任何额外的部署所具备的主界面,这就相当于直接让后台管理和普通用户交互了,这在任何的开发环境都是不允许的,那么后续的开发就是由林同学负责的了,具体开发细则我也不是很清楚了,进行了一段时间的开发后,第一版的效果就是添加了账号的登录功能,并且隐藏了后端 admin 模式,普通用户访问后就是一个数据的增删改查功能的可视化,里面是使用了 HTML 网页语言的,使得页面得以呈现出我们想要的样子。在开发后期,林同学加了一个额外的功能,更加清楚地对管理员和普通用户的身份权限加以了区分,管理员身份登入的可以为新的账号开通权限,普通用户可以使用特定的账号登录添加信息,这样每个用户添加的信息也就能够做到仅自己可见的效果了,毕竟隐私还是要保护的,这里附上最后版本的一个截图。



## 我的服务器

[浏览信息](#) | [添加信息](#)

### 浏览服务器信息

IP	服务器地址	电脑型号	管理员姓名	管理员电话	管理员邮箱	服务器温度	负载	操作
127.0.0.1	天目湖	lenovo	谢	17849890562	1833610970@qq.com	92.5	31.6%	<a href="#">编辑</a> <a href="#">删除</a>
11.11.11.11	常州市溧阳县	dell	张三丰	13810105267	2781079007@qq.com	92.5	31.6%	<a href="#">编辑</a> <a href="#">删除</a>
1.1.1.1	Hogwards	Lenovo	小明	123456789	214460165@qq.com	92.5	31.6%	<a href="#">编辑</a> <a href="#">删除</a>



## login Form

### 账号登录

登录账号	<input type="text"/>
登录密码	<input type="password"/>
<input type="button" value="登录"/>	

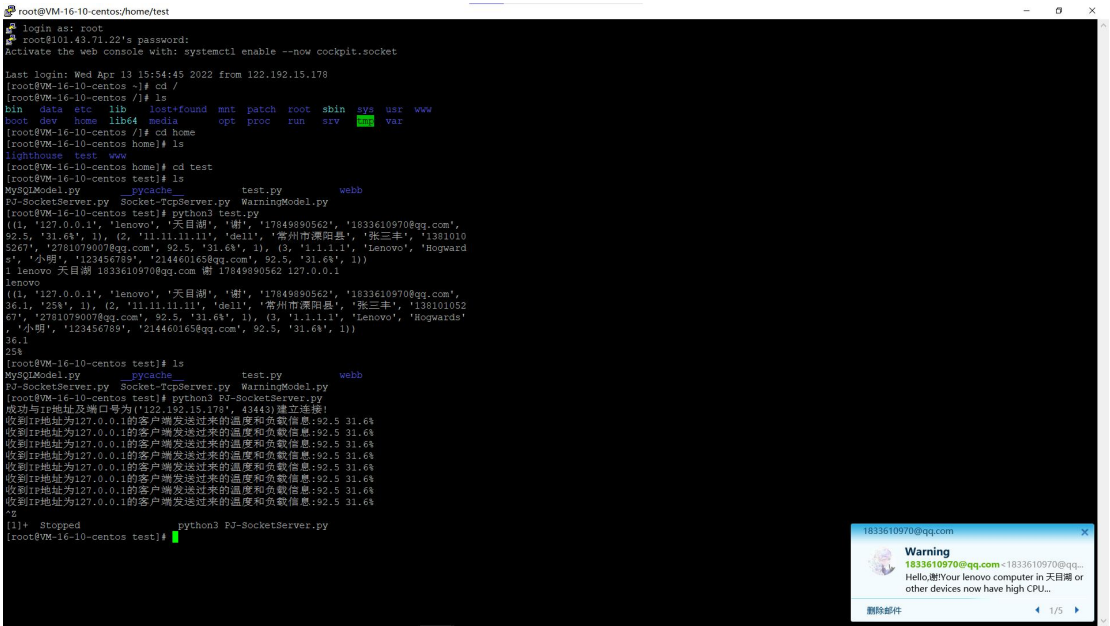
测试账号: nuua 密码: 666 (具有后台权限)

测试账号: nubbb 密码: nuua123456

3) 第二个重要的部分就是数据的交互和管理,在任何系统和软件中,数据的管理、维护、运维、删改都是极为重要关键的,维护不好数据,这个系统的其他功能都无从谈起了。我们在这里的研究采用的是 MySQL 数据库进行数据的运维,数据库由很多类别,市面上也有很多种类,甚至可以使用文件编程来实现数据的管理,但是数据库的管理肯定在很多方面要优于文件管理的,数据的各种管理操作更加方便,共享性更大,冗余性更小,由系统服务器统一管理,可以远程部署和访问,物理独立性和逻辑独立性是非常高的,而且有很好的接口特性,可以和很多程序建立连接和接口。此外,MySQL 的开源性、跨平台性、社区版本的免费使用以及简便操作和强大功能最终使我选择了使用 MySQL 来实现本系统的数据管理。在安装和配置好 MySQL 后,我就花了一段时间迅速学习了 MySQL 数据库的基本操作、数据表的基本操作、表中数据的基本操作、用户管理等大部分基础内容。而后就是要用数据库将整个系

统紧密地连接起来。这里采取的主要操作和方法就是，本地 MySQL 新建了一个数据库 Test 专门用来本项目的数据库管理，Django 的配置板块的数据库模块和本地 MySQL 服务器信息进行绑定，这样能够实现前端 Web 界面输入的信息和后端 MySQL 服务器之间的连接交互。同时，我也在 python Socket 程序中写入了一个 MySQL 操作模块，这样便能够在用户输入完信息后，我能在后端程序中利用和 MySQL 的接口实现数据的获取，那么通过 Socket 连接后获得数据后我可以将传入过来的温度负载等动态更新的信息写入数据库，由于 Django 数据库的绑定，MySQL 数据库中表中数据利用了 SQL 语言更新后，用户也可以实时动态地在前端 Web 界面看到自己登入设备的温度负载等信息的随时动态地更新。这样我利用 MySQL 数据库实现了数据在前后端的动态更新、维护、绑定，也让前后端分离式的架构层次更加清晰。

4) 在有了 3) 中 MySQL 数据库良好的部署之后，前后端数据的动态维护得到了保证，在这样的基础上，我进一步考虑实现温度阈值判断和报警信息发送的功能，利用 Python 的 yagmail 模块直接写了一个 Warning 模块，通过数据库接口获取设备绑定的设备拥有者的电子邮箱，再利用邮箱所提供的简单邮件传输协议 SMTP<sup>1</sup>、邮局协议 POP3<sup>2</sup>、交互式邮件存取协议 IMAP<sup>3</sup>等服务，这里我采用的是 QQ 邮箱进行测试和维护，便可以动态绑定监控的设备信息，并且可以及时地判断是否需要报警然后给绑定的设备拥有者的邮箱发送报警信息，下面附上一个测试中报警信息成功通过系统调用发送邮箱的测试截图。



```
root@VM-16-10-centos:/home/test
login as: root
root@101.43.71.22's password:
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Apr 13 15:54:45 2022 from 122.192.15.178
[root@VM-16-10-centos ~]# cd /
[root@VM-16-10-centos ~]# ls
bin  data  etc  lib  lost+found  mnt  patch  root  shin  sys  usr  www
boot  dev  home  lib64  media  opt  proc  run  srv  var
[root@VM-16-10-centos ~]# cd home
[root@VM-16-10-centos home]# ls
lighthouse  test  www
[root@VM-16-10-centos home]# cd test
[root@VM-16-10-centos test]# ls
MySQLModel.py  pycache  test.py  web
[root@VM-16-10-centos test]# python3 test.py
(1, ('127.0.0.1', 'lenovo', '天目湖', '湖', '17849890562', '1833610970@qq.com',
92.5, '31.6%', 1), (2, ('11.11.11.11', 'dell', '常州市溧阳县', '张三丰', '13810105267', '27810790078qq.com', 92.5, '31.6%', 1), (3, ('1.1.1.1', 'lenovo', 'Hogwards', '小明', '123456789', '214460165@qq.com', 92.5, '31.6%', 1))
1 lenovo 天目湖 1833610970@qq.com 湖 17849890562 127.0.0.1
lenovo
((1, ('127.0.0.1', 'lenovo', '天目湖', '湖', '17849890562', '1833610970@qq.com',
92.5, '31.6%', 1), (2, ('11.11.11.11', 'dell', '常州市溧阳县', '张三丰', '13810105267', '27810790078qq.com', 92.5, '31.6%', 1), (3, ('1.1.1.1', 'lenovo', 'Hogwards', '小明', '123456789', '214460165@qq.com', 92.5, '31.6%', 1))
36.1
25%
[root@VM-16-10-centos test]# ls
MySQLModel.py  pycache  test.py  web
[root@VM-16-10-centos test]# python3 PJ-SocketServer.py
成功与IP地址及端口号为('122.192.15.178', 43443)建立连接!
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
收到IP地址为127.0.0.1的客户端发送过来的温度和负载信息:92.5 31.6%
^
[1]+  Stopped                  python3 PJ-SocketServer.py
[root@VM-16-10-centos test]#
```

<sup>1</sup> SMTP 的全称是“Simple Mail Transfer Protocol”，即简单邮件传输协议。它是一组用于从源地址到目的地址传输邮件的规范，通过它来控制邮件的中转方式。SMTP 协议属于 TCP/IP 协议簇，它帮助每台计算机在发送或中转信件时找到下一个目的地。SMTP 服务器就是遵循 SMTP 协议的发送邮件服务器。

<sup>2</sup> POP3 是 Post Office Protocol 3 的简称，即邮局协议的第 3 个版本，它规定怎样将个人计算机连接到 Internet 的邮件服务器和下载电子邮件的电子协议。它是因特网电子邮件的第一个离线协议标准，POP3 允许用户从服务器上把邮件存储到本地主机（即自己的计算机）上，同时删除保存在邮件服务器上的邮件，而 POP3 服务器则是遵循 POP3 协议的接收邮件服务器，用来接收电子邮件的。

<sup>3</sup> IMAP 全称是 Internet Mail Access Protocol，即交互式邮件存取协议，它是跟 POP3 类似邮件访问标准协议之一。不同的是，开启了 IMAP 后，您在电子邮件客户端收取的邮件仍然保留在服务器上，同时在客户端上的操作都会反馈到服务器上，如：删除邮件，标记已读等，服务器上的邮件也会做相应的动作。所以无论从浏览器登录邮箱或者客户端软件登录邮箱，看到的邮件以及状态都是一致的。



5) 上述方法步骤基本上都是在架构系统的前后端分离体系和接口部分,接下来就是本系统的要求实现的最核心的部分了,就是 Socket 通信模块。简单地说一下 Socket, Socket 翻译过来被称为“套接字”,它基于计算机网络协议中五层网络模型(应用层、运输层、网络层、数据链路层和物理层)中运输层中 TCP 和 UDP 协议<sup>4</sup>,而 Socket 就是在操作系统和数据发送软件层之间的类似接口一样的东西,它通过各种操作将要发送的数据包进行加工处理并添加一些额外信息组合成新的数据包传送给网络层,然后便可实现数据在网络之间的传输。其实本来这一块编程任务是分派给蔡同学完成了,但是在 Python Socket 编程中本身的代码工作量并不太大,所以这一块我也进行了负责,并且之后的优化更新代码重构也是由我自己独自来完成的。本身的代码逻辑由很多资料可以参考,很快便能在本机上完成一个客户端和服务端的代码编写,然后在本机上运行示例可以顺利实现信息的发送和交互。但是紧接着我又考虑到这样的服务端和客户端的交互,一旦当出现多个客户端同时和服务端建立连接进行信息通信传输的情况,是难以正常运作的(这里简单解释一下,谈谈自己的理解,因为服务端程序必须不停地对监听 socket 对象调用 accept 方法,这才能不断地接收新的客户端连接请求,而且还需要额外运行代码对多个客户端连接后,返回的多个数据传输 socket 对象进行数据的收发,又由于默认情况创建的 socket 是阻塞式的,进行 accept 调用时,如果没有客户端连接,程序就会阻塞在此处不再执行后面的代码,同样的,调用 recv 方法在没有数据在本 socket 的接收缓存中也会阻塞,因此通常一个线程里面没有办法不断调用监听 socket 的 accept 方法来负责多个数据传输 socket 消息的收发)。于是进一步的,我去简单地看了一下 Python 的多线程编写,虽然没有很透彻地去理解其中的一些原理,也大抵上利用了 Python threading 模块实现了 Socket 的多线程编写,再结合之前部署的接口和模块,顺利实现了多个客户端同时访问一个服务端的问题,并且测试能够顺利运行。

6) 至此,基本的框架模块的部署已经完毕,系统的最后一步就是进行跨主机跨域之间的通信实现了,我们起初自然地认为按照这些自然的代码逻辑实现可以轻松完成跨域通信,可当我们开始使用不同的主机进行测试的时候发现了遇到了很多问题,主要有两个,一个是本机运行的 Django 框架 Web 界面只能在本机上访问,其他电脑是无法访问的,第二个就是两台主机用 Socket 进行通信的时候,跨域情况下(同一个局域网下就可以)输入本机的 IP 地址无法 ping<sup>5</sup>通,这里我们尝试搜索了很多解决方案,猜测了很多理由:比如路由器设置将其他端口关闭了、主机的防火墙拦截设置、使用的是组建局域网时的私有 IP 地址<sup>6</sup>、忽视

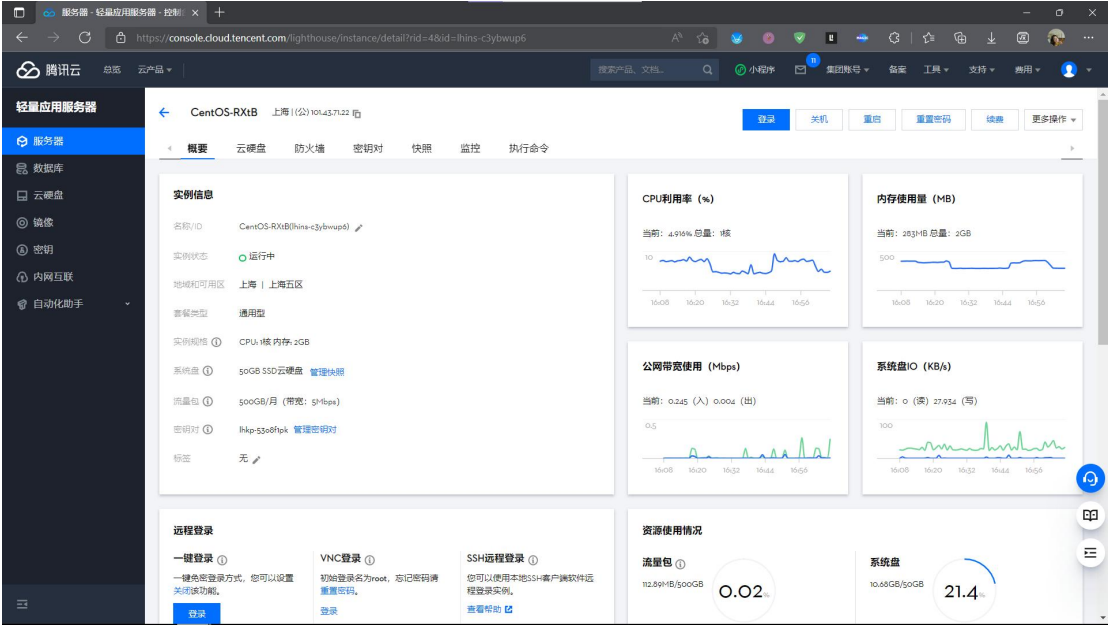
---

<sup>4</sup> 传输控制协议(TCP, Transmission Control Protocol)是一种面向连接的、可靠的、基于字节流的传输层通信协议,由 IETF 的 RFC 793 定义。用户数据报协议(UDP, User Datagram Protocol)是 IP 上层的另一重要协议,它是面向无连接的、不可靠的数据报传输协议。它仅仅将要发送的数据报传送到网络,并接收从网上传来的数据报,而不与远端的 UDP 模块建立连接。

<sup>5</sup> ping (Packet Internet Groper)是一种因特网包探索器,用于测试网络连接量的程序。Ping 是工作在 TCP/IP 网络体系结构中应用层的一个服务命令,主要是向特定的目的主机发送 ICMP (Internet Control Message Protocol 因特网报文控制协议) Echo 请求报文,测试目的站是否可达及了解其有关状态。

<sup>6</sup> 私有地址主要用于在局域网中进行分配,在 Internet 上是无效的。这样可以很好地隔离局域网和 Internet。私有地址在公网上是不能被识别的,必须通过 NAT 将内部 IP 地址转换成公网上可用的 IP 地址,从而实现内部 IP 地址与外部公网的通信。公有地址是在广域网内使用的地址,但在局域网中同样也可以使用,除了私有地址以外的地址都是公有地址。比如我们当时 ping 的以 192.168.0.0-192.168.255.255 之间的地址。

了或者不清楚 DHCP<sup>7</sup>、TCP/IP 等协议的工作原理等等，最终都劳而无功。这个时候我想到了我们平常在访问百度啊腾讯啊等网站的 IP 均可以正常访问，这里面是涉及服务器部署和 DNS 域名<sup>8</sup>注册的一些概念，但也给了我不少想法和思考。既然私有 IP 地址无法连接，那为何不使用公网 IP 来访问呢？就好比访问百度的主页，百度的服务器已经部署好了，网络中就可以映射到它的 IP 地址上去，故我们可以顺利访问，同时考虑到最近云服务器行业的盛行，因此这一点我最终决定使用云服务器来部署和架构。大致上，是使用了学生优惠在腾讯云上租了一台 CentOS 系统的云服务器，也就是 Linux 系统，它具有公网 IP 为 101.43.71.22，然后就是将服务端的 python—Socket 程序、Django 框架、MySQL 数据库服务器全部在云服务器部署完成即可。在这一过程，首先由于原先都在 Windows 操作系统上尝试，在 Linux 系统上会有一些的不同的地方要注意，其次是第一次接触服务器的部署，遇到了不少问题，但通过自己的不懈努力，借助宝塔的云服务器运维系统，以及 Putty 的远程登录服务和 WinSCP 将本地文件一键传输至云服务器的 Linux 系统中，最终也是全部部署迁移完毕，进行利用云服务器的公网 IP 测试顺利实现了整个研究内容和系统预设的功能的集成和实现。这里附上云服务器的一个主界面信息截图。



## 2.2.结论和成果:

借助 Putty、WinSCP 和宝塔 Linux 运维顺利将关键架构部署到云服务器中且能顺利运行，可以通过网址访问的方式进入 Django 框架开发的 Web 界面进行信息的输入和登记，数据库操作的可视化，并且能和 MySQL 数据库保持强关联和交互性，借助云服务器的公网 IP 实现

<sup>7</sup> **DHCP** (Dynamic Host Configuration Protocol, 动态主机配置协议) 是一个局域网的网络协议。指的是由服务器控制一段 IP 地址范围，客户机登录服务器时就可以自动获得服务器分配的 IP 地址和子网掩码。

<sup>8</sup> **域名系统** (英文: **Domain Name System**, 缩写: **DNS**) 是互联网的一项服务。它作为将域名和 IP 地址相互映射的一个分布式数据库，能够使人更方便地访问互联网。DNS 使用 UDP 端口 53。

了可 Ping 和可路由,从而使得服务端和客户端的 Socket 通信能顺利进行,多线程的设计也保证了多个客户端能同时和服务器建立 Socket 连接,且能够顺利传输数据,POP3/IMAP 和 SMTP 协议服务保证了邮件发送的报警信息能顺利传达。总结起来就是,我们实现了 Web 的自由访问,并且能在上面登记输入每个客户端主机的信息,而且能够利用 Socket 正确地 and 客户端通信传输数据,并且能用多线程和多个客户端同时通信,也一并实现了报警功能,非常漂亮地实现了前后端分离的架构。以上大概就是我们本次项目研究所取得的成果。

### 三、项目创新点

本项目的开发有几个特色和创新点,这里简单罗列说明一下:

(1) 本信息传输系统是借助 Django 框架实现了在 Web 界面上进行数据库的增删改查,并且在网站开发中融入了权限管理的思想,对管理员权限和普通用户权限之间设置了区分和不同的权限管理,一定程度上做到了对用户的隐私信息的保障和数据可见性的限制。

(2) 前后端分离的架构思想在本项目中得到了充分体现,通过 MySQL 数据库服务器在数据上起到的桥梁作用,既能够将客户在前端 Web 界面上输入的信息在后端中可以获取,同时也可以在后端将获取的用户信息和通过 Socket 接口传输的信息再进行数据的更改,动态绑定中又将更新后的数据重新在前端 Web 界面得到了展示,同时避免了使用系统的用户直接和后端数据接触,起到了对数据的有效保护,体现了大部分软件普遍具有的前后端分离的良好特性。

(3) 第三个创新亮点就是云服务器的巧妙使用,在普通通信和访问出现了不少问题后,巧妙地使用了云服务器的公网 IP 来进行项目的部署和架构,从而解决了无法访问的问题,使得项目的应用有了更广阔的前景,应用潜力也得到了更大的激发。也极大地方便了访问,同时也使得项目代码在需要迁移的时候有了极大的便利,几乎整个的关键代码全都部署在云服务器,任何人在任何地方都可以进行访问和连接。

(4) 第四个小亮点就是在代码编写的过程中,大块地使用了类与对象的编程模式,使得项目代码的封装更加清晰,因此在代码维护和后期更改的过程中会取得不少的便利。

### 四、应用前景

正如本项目的研究背景,我们的研究是基于当今很多设备会进行长时间的处理器运转而导致温度过高或者过载,产生威胁影响和造成不可估量的损失,因此我们对能够远程监控温度并实时报警提出了需求。基于这样的背景,我们项目的应用背景主要便体现在对于任何一个需要接受管理的设备,进行远程访问注册,登记信息后,我们的程序系统便可以运行起来,实时监控对方的设备的运行信息,一旦遇到可能会发生过热或者过载的情况,提前发送邮件

报警信息提示设备管理人。当然类与对象的编程开发模式使得使用者可以提出更多的需求，可以进一步往代码框架中添加这些新功能，也可以优化代码设计，使得程序的性能和稳定性更高。此外，本项目的 Web 开发、Socket 通信模块以及邮件传输报警本身就是一些不错的、成型的小项目，因此在有 Web 开发和云服务器部署等类似需求的应用场景，我们的项目都可以进行应用。

## 五、收获和体会

在本次项目开发和研究中，历时几个月的磨练和努力，自己收获颇丰，这里简单说明一下：

（1）首先，获得了更扎实的编程能力，在编写项目代码的过程中，在每一个细节的优化和逻辑的设计部分，都需要投入大量精力，调试改错也花费了特别多的精力，自己的编码能力得到了较大的提升；

（2）拓宽了自己的知识面，本项目的大的专业背景是计算机网络，这样一个在当今生活中随处可见、处处应用的东西，曾经对于它的运转总觉得神奇，在这个项目的开发研究中，对于网络中的很多概念 and 知识都有了不少的涉及，而且甚至自己也通过编程实现了这样一个小的网络通信系统。

（3）磨练了自己的意志，几个月的开发中，遇到了很多很多意想不到的问题，有的问题研究一会儿就能得到解决，有的问题却百般尝试也难以解决，以至于最终决定更换一开始的解决方案，重写代码，很好地磨练了自己遇到困难不放弃、迎难而上、永不言弃的意志力和品质。

（4）多方面锻炼了自己的能力和品质，作为项目的负责人，既要和指导老师做好对接沟通工作，也要和小组成员沟通好，设定好解决方案，分配好工作，身先士卒完成工作和任务，积极带动成员积极性，完成项目的所有部分和细节的合并工作，这些都在多方面地磨练了自己的多种能力和品质。

（5）更好地培养了科创精神，在项目开发的过程中，整个项目开发的流程，自顶向下的开发设计，采取前后端分离的开发模式，类与对象的编程模式，模块的组织，严谨的校对改错等等等等，这些方面都非常好地培养了自己的科创精神和动手实践能力。

（6）对于计算机行业有了更深刻的认识，这一点是从自己的专业素养来讲的，开发完这个系统，无论是过程中还是过程后，顺利测试运行之后，深刻感受到在互联网和电子时代下计算机的用途之大，它计算又快又准，不会出错，而且会严格执行逻辑任务，能够智能自动地解决很多需要耗费不少人力资源的问题。

感觉收获还有很多，自己也有了很大的进步和成长，受限于篇幅，姑且就不再赘述了吧。

## 六、展望

展望部分，其实在前文中的叙述也有体现了，整体上看，这个项目还存在如下一些问题：

（1）Web 端界面开发中并没有完全能够做到用户的信息保护和管理员以及普通客户的权限分离，采取的一些设计并没有能够做到真正意义上的用户注册和登录功能，这一块展望能够进一步完善并且做出账号的注册和登录功能，并在后端 MySQL 数据库服务器中维护用户信息，使得功能更加具体完善。

（2）无论是 Django 框架的运行还是 Python-Socket 服务端程序的运行，都需要自己在云服务器上使用 Linux 系统调用命令来人为运行，展望是能够更进一步实现自动化和绑定状态，也即让云服务器持续运行 Django 框架和 Socket 服务端代码，并且融入保活和监测机制，同时能够实现当用户注册登记了信息后就能够自动建立起服务端和客户端的连接，进行信息传输并能够监控报警。

（3）第三个就是在程序代码的各部分可能都会存在一些细微的不足和问题，以及有些实现还是存在一定的缺陷，比如多线程实现多个客户端利用 Socket 和服务端通信会让操作系统创建大量的线程增大开销，而且可能操作系统就不会为一个进程分配这么多的线程，因此还应考虑更加稳定可靠的方式实现多端通信。此外一些功能还可以优化得更好，也可以再加入一些新的功能使之更加完善。