

浙江大学

本科实验报告

课程名称: B/S 体系软件设计

姓 名: 徐文皓

学 院: 计算机科学与技术学院

系: 软件工程系

专 业: 软件工程

学 号: 3210102377

指导教师: 胡晓军

2024 年 01 月 02 日

浙江大学实验报告

课程名称： B/S 体系软件设计 实验类型： 综合

实验名称： 物联网设备管理平台

学生姓名： 徐文皓 专业： 软件工程 学号： 3210102377

同组学生姓名： / 指导老师： 胡晓军

实验地点： 线上 实验日期： 2024 年 01 月 02 日

物联网设备管理平台 软件开发心得

目录

01.	引言.....	3
02.	需求分析和系统设计.....	3
03.	前端开发.....	4
04.	后端开发.....	7
05.	数据库管理.....	8
06.	MQTT 服务器使用	9
07.	测试.....	9
08.	Docker 部署	10
09.	总结与展望.....	10
10.	对课程的建议.....	10
11.	致谢.....	11

01.引言

本文档是物联网设备管理平台的软件开发心得。

经过一学期以来的学习和探索，我完成了物联网设备管理平台这一软件的全栈开发。在这次开发中，我体验了 B/S 体系软件设计和开发的全过程。在这之前，我在软件工程基础这门课程中参与过一个在线支付管理系统的某个子模块的前端开发；本学期，同时也在软件工程管理课程中参与一个垂直搜索引擎的部分前端开发——这些都是 B/S 体系中客户端的部分。在之前小组一起做则会由于分工不同而缺少对一些必要过程的体验，而在本次课程中，一切都由我独立完成，从前端开发者的角色转换成全栈开发者的角色，我接触到很多新东西。

在此，有所感，为文以记之。



图 1 项目技术选型

02.需求分析和系统设计

需求分析是软件开发过程的第一步，是极其重要的一步。

在需求分析阶段，面对老师提供的实验要求，我需要弄清楚自己需要做什么，才好保障整个开发过程按照预期的路线前进。老师给的实验要求还是比较清晰明确的，我的工作是将这些需求明确和细化，使其更加具有可行性和可操作性。

由于在开发初期并不明确自己的技术水平是否可以实现理想的开发成果，也考虑到后续开发迭代的需要，我的功能性需求分析是以较粗粒度开展的。相比于老师的需求，在这里需要提出一些错误处理和有助于提高用户友好性的优化，例如重复性验证、非空性验证和格式正确性验证等，都是之后需要实现的功能。而在非功能性需求方面，我想提出一些切实可行的、能够实现的，而不是在其他课程中“八股文”一样的形式——例如比较良好的性能、URL 控制、可视化水平包括密码加密等等——这些在后续开发的过程中基本算是实现了。

结合需求分析，就可以进行系统设计。

那时我除了用别人帮忙搭好的框架写过几个 .vue 文件之外，其实对 B/S 的各项技术都不了解。当时只是想选一些比较热门的，这样之后有问题也方便互相交流。关于技术选型的体会，稍后我将在每个模块的部分详细展开，此处暂且不表。

按照对需求的理解，我初步设计出了数据表和接口——当然这些在之后的开发过程中都

根据实际需要发生了变化。在形成《需求规格和系统设计说明书》之前，我其实已经进行了一些探索，形成了基本的前端静态页面，这也直接在文档的系统界面原型部分体现了出来。

概括说来，这一步进展的是比较顺利的，需求很明确，我自以为理解得还是比较充分，在后续的开发过程中没有发生什么大的需求变更。

03.前端开发

我坚定地选择了曾经“合作”的 Vue 框架、Element Plus 组件库和 ECharts 可视化库。

虽然有过一些使用经验，但是之前主要还是别人帮忙搭搭框架，然后自己负责一些.vue文件罢了。但这次来说，可以算是比较系统地学习了这个框架。参考网上的某个教学视频，我从 Vue CLI 开始创建项目，使用了 pinia 这一状态管理库（它好像比 vuex 会再简单些），选择 Vue router 来进行路由管理。当时看的教学视频其实是使用 Vue2 的，可能出于时效性的原因，那时 Vue3 在整个技术栈中的兼容性还不是很好，配置起来会比较麻烦——但是我还是觉得，一方面要拥抱新事物、了解新技术，另一方面是这学期在写的另一个项目在使用 Vue2，两者对比起来我可能会理解得更加透彻些——于是就大胆地选择了 Vue3。整体还是比较顺利的，初期做一些修改把整个系统改崩溃了，就直接重新创建一个项目，后面改一些看一下情况，没有出现特别大的高严重性问题。

这次系统学习后，才发觉自己之前写的前端代码是极其混乱的。这次除了调用地图 API 的部分相对比较复杂、还是要参考文档才能配置得比较流畅之外，其余均使用 Vue 选项式语法。项目创建后，最开始做的事情就是配置路由，先把主页面的导航栏和每个页面的路由配置好，再开发每个页面——也算是一种自顶向下的结构了。

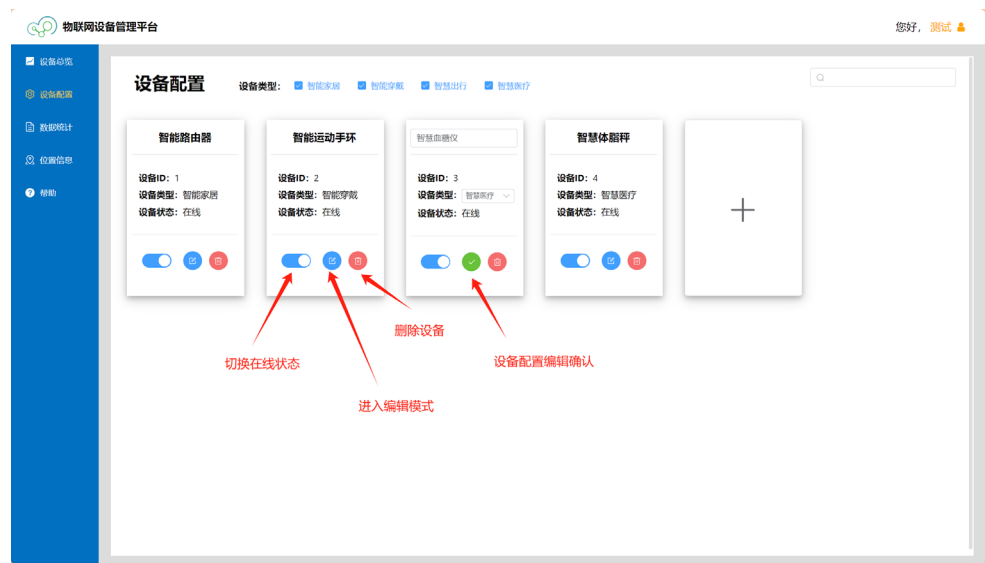


图 2 设备配置界面

最先实现的是设备配置页面，我觉得这也是我最自豪的一个页面了——卡片式显示风格一方面简洁美观，更重要的是可以实现对不同设备的独立管理，在修改一个设备的配置时允许将其他设备的修改状态暂时“挂起”，我认为这样在很大程度上提高了用户友好性。同时，在线状态修改简单，新建设备卡片页十分符合我们日常直觉上对新建设备按钮的期待。在这里，包括之后的设备总览部分，下了一个很重要的功夫是，卡片数量是可以根据屏幕宽度而灵活调整的，我将卡片的尺寸固定为绝对宽高，通过限制卡片数量来进行布局，而不是通过将卡片的尺寸设置为相对宽高，这样不至于在屏幕宽高比发生变化时出现比较奇怪的效果——这也在事实上为移动端适配奠定了基础。

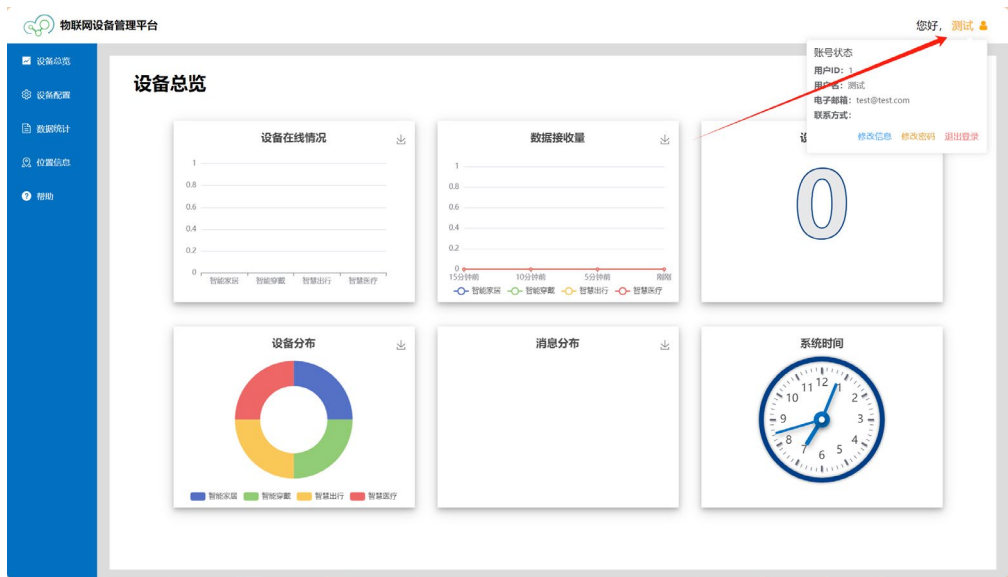


图 3 账号状态管理

之后实现的是账号状态管理框，这个感觉也是用户友好性比较高的。相比于单独设置一个账号管理页面——不难想象，我们所维护的用户信息只有几个属性，单独使用一个页面可能有些“大材小用”，页面会没有什么东西可放——这样的实现会更加轻量级，用户将鼠标悬浮在指定位置即显示。当然，选择“悬浮”这一动作一开始是有些顾虑的，我专门去移动端验证了一下发现是可以点击来弹出的。

设备总览部分主要是调用 ECharts 可视化库中的内容，当时在思考如何安排这个页面，最后感觉放 6 个图表是比较美观的，因为 6 是 2 和 3 的倍数，在屏幕宽度缩放时总能填满最后一行，不至于出现第一行很满第二行只有一个的情况。但是在选择图表的时候还是比较头疼的，好像没有那么多数据给我来表示，图表之间也会比较重复。经过反复思考，最后选择了这六个图表来进行显示，每个图表都各有特色且没有重复，还是比较开心的。

数据统计页面比较简单，考虑到平台数据规模较小，我选择不分页但是置入筛选和搜索功能，相当于把筛选工作由后端转移到了前端，这个感觉仁者见仁智者见智。

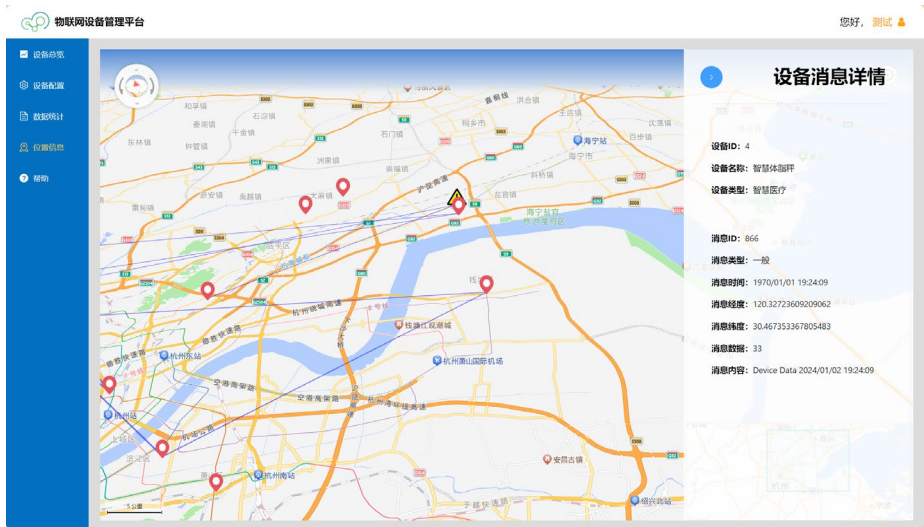


图 4 位置信息显示

位置信息界面主要是调用地图 API,在和同学交流了使用经验后我选择了高德地图 API,整体使用比较顺利,包括地图标点连线等工作做得都比较快。只是单放一个地图似乎显得有些单薄,我设计了一个点击地图点后展开侧边栏显示详细信息,这样页面内容会更加丰富些。



图 5 用户登录界面及用户注册界面入口

登录注册界面可以说是之前的“老本行”了，毕竟我最开始接触前端就是负责这一部分。这次主要是在选 logo、调背景、尺寸、透明度等等方面下功夫，同时还学到了过渡显示用法。这一部分是在移动端显示最偏离预期的部分，为此我选择的逻辑是“仅在横屏时显示 logo”，在一定程度上在移动端显示得更好。

在完成所有功能及测试之后，我开始撰写用户使用手册，突发奇想将它嵌入到了帮助页面，没有增加很大的工作量但是提高了用户友好性。

我还记得，在上学期的软件工程基础课程中，我第一次参与前端开发，那时在一周之内突击完了 HTML、CSS、JavaScript 等语法，用起来也不是很熟练，甚至调一个居中都要在各个博客之间颠沛流离。到现在，已经算是得心应手，尽管仍有重重阻碍，但已经基本能够用这些工具将自己心中的蓝图擘画出来。

此外，在前端还进行了一些路由管理。在之前的系统中，每当刷新就要重新登录，用户体验很差。在这个项目中，我学到了 Session 的使用，可能只是一小步，但对我来说确实是一个很重要的突破。这也是我的 URL 守卫能够比较好发挥功能的基础——在未登录状态刷新会回到登录页面，使用登录后的界面的 URL 或其他非法 URL 均会跳转到登录页面，在已登录状态刷新会来到设备总览页面，使用预期外的 URL 也是一样。

同时，前端也使用 SHA256 对密码进行了初步加密，使得向后端发送的网络请求中不含明文密码，一定程度上提高了安全性。

04.后端开发

在后端，我选择了 SpringBoot 框架和 MyBatisPlus 持久层。相对于前端，我在后端就是一个纯粹的“小白”了。所幸本次项目的后端要求其实并不高，比较简单，在进行一些学习后，我在 Controller 中就基本实现了处理网络请求的逻辑。

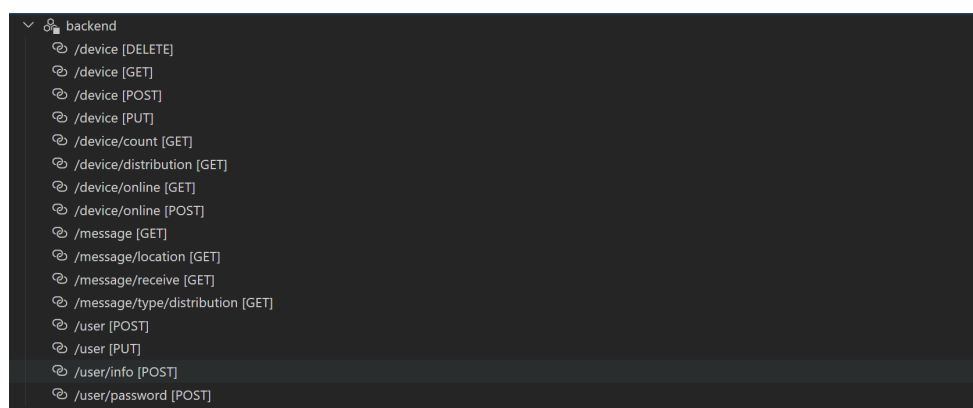


图 6 前后端接口

后端基本就是对数据库进行增删改查操作，MyBatisPlus 确实在很大程度上方便了开发。在之前的数据库系统课程中，我使用过一次 JDBC，那里确实比 MyBatisPlus 要复杂很多。

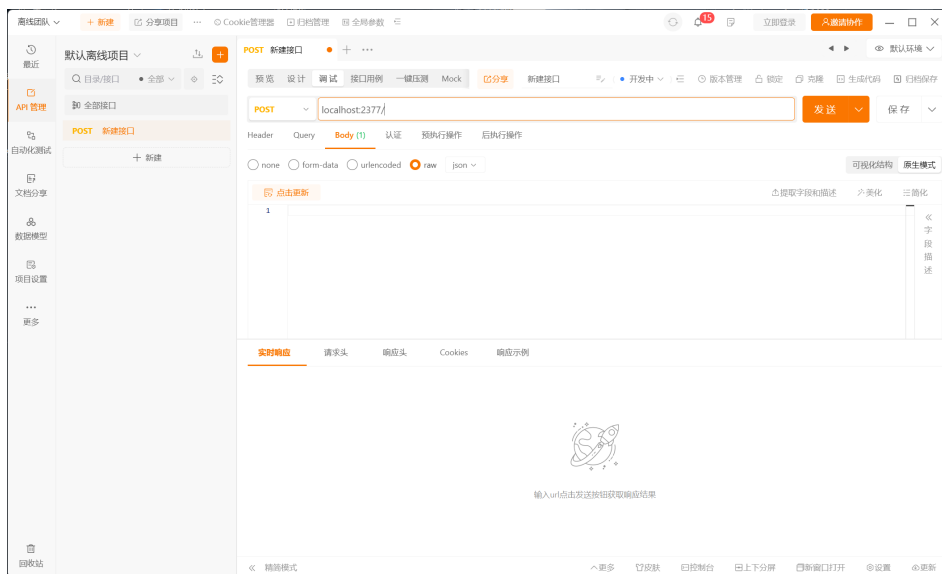


图 7 ApiPost 界面

虽然之前没有写过后端,但是在开发前端的时候也是要用到网络请求的.当时为了方便,我们采取“一切请求都用 POST”的原则,这次也学会了使用 ApiPost 这一工具来直接测试后端的有效性.而更重要的是,这次我选择了使用 RESTful 风格,更加合理地区分了 GET、POST、PUT、DELETE 请求,将 URL 中的动词基本隐去,很大程度上提升了规范性。

后端遇到的一些障碍,有相当一部分集中在 Maven 依赖的配置上,它们确实还是存在一些兼容性问题,所幸,这些经过查阅文档和博客最后得到了比较好的解决.还有一个需要提到的是,跨域问题.在这里也是下了一些功夫.对于密码,我采用了前端后端两次加密的模式,前端对密码进行 SHA256 加密后发到后端,后端再进行 MD5 加密存储到数据库。

真心感觉,虽然之前做过前端开发而没有做过后端开发,但无论如何 JAVA 还是比 JavaScript 用起来会亲切地多。

05.数据库管理

数据库使用 MySQL,这也是老朋友了——我对数据库一直都是很有好感的.在开发的过程中,数据表经过了多次修改,最后确定了一个还不错的版本.这个项目也让我学到了一些新的数据库语法,如 CREATE TABLE IF NOT EXISTS 等,也是我第一次使用建表脚本。

其中一个值得提及的是一个比较关键的数据列的去留问题.我曾思考是否要将 user 表中的 id 作为 message 表的一个外键,如果作为的话这个表可能会显得有些笨重,但省去了多表查询的环节.从实体关系上思考,我还是觉得只用 device 的 id 作为 message 的外键在逻辑上会更加合适一些。

06.MQTT 服务器使用

MQTT 服务器我使用了 EMQX。尽管没有接触过后端开发，但在小组合作中多少还是接触过一些后端的内容的，但这个 MQTT 确实是新面孔。一开始其实并没有搞懂它的作用是什么，但是在仔细阅读老师的测试代码之后，基本有所了解。

考虑到数据表的差异，我将测试代码进行了一些本土化，并将其内嵌到了我的后端代码里。后端采用多线程模式，在启动时会分出一个子线程用于订阅 MQTT 服务器消息，分出多个子线程模拟物联网设备向 MQTT 服务器发送消息。

将测试代码嵌入后端代码的目的是让设备消息发送更符合逻辑。在后端启动时，将会向数据库发送查询请求，得到目前在线的所有设备，并且为这些设备创建子线程，让他们开始向用户发送信息。当设备状态切换成在线时，为其创建子线程；而切换为离线时，终止其子线程。也就是说，在后端开启状态下即可接收模拟物联网设备消息，即使某个用户没有登录但他有在线设备，那些设备也会向 MQTT 发送消息。

当然，这一逻辑只是为了模拟测试。在发布正式版本时，模拟部分将被删去，后端和 EMQX 在代码上不会有很强的耦合关系。

07.测试

由于在本学期修读了软件质量保证与测试课程，这次的测试在我看来相比之前会更具专业性。在提出测试计划时，我聚焦正确性、可靠性、性能、安全性、兼容性等多个因素，结合等价类划分法等测试方法，设计了比较详尽的测试用例。在那门课程上，我们也实现了 AETG 组合测试算法，我也将其应用到了我的测试用例设计中。

其实很多缺陷在单元测试部分就解决了，这次专门性的系统测试所面向的对象其实已经是非常成熟的版本了，在设计的 125 个测试用例中也只发现了 1 个一般缺陷。

移动端适配的效果比想象中好很多，并没有做非常专门的修改。

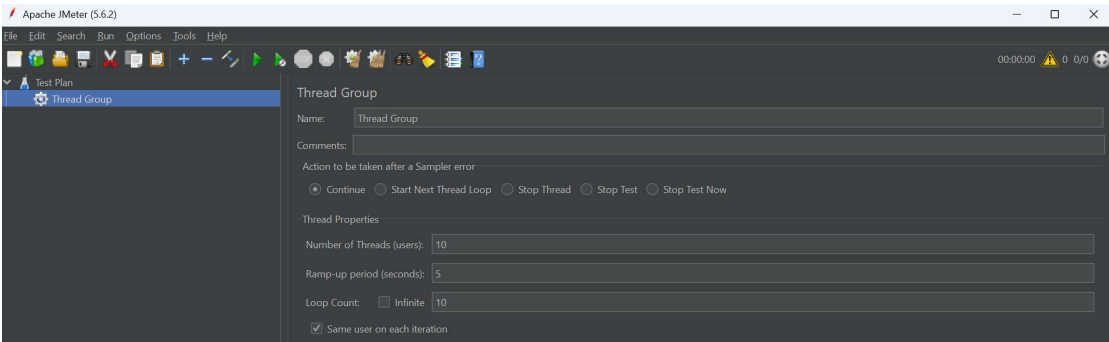


图 8 性能测试线程组配置

一个学到的新东西是 JMeter，掌握了一些压力测试技巧。

此外，我也积极地邀请了很多朋友参与测试，总体反馈良好。

08.Docker 部署

毫无疑问，Docker 部署是本次开发环节中最难的部分。

这一环节高度结合了同学、GPT 和各大网站的支持。

自本学期初步学会使用 Makefile 之后，本次了解 Dockerfile 是又一重大突破。在部署过程中，遇到的阻碍主要有不同镜像间的网络问题、建表问题等。

关于网络问题，后端在连接 MySQL 和 EMQX 镜像时需要将连接地址 `tcp://localhost:1883` 替换为形如 `tcp://emqx:1883`，否则无法正常连接。数据库建表的脚本如何置入，也是一个研究了许久的问题。此外，还有时区的配置问题等等，总之是过关难过关过。

09.总结与展望

本次开发进展比较顺利，在此过程中我学习了一套非常有用的技术栈，将很多零散的知识系统地整理了起来，对很多东西有了更加深入的理解。

但实事求是地说，项目中还有很多值得更好的地方，有些地方自己还是会偷一些懒，甚至部分功能只是将代码移来移去就莫名其妙地跑了起来，却由于学业压力没能去深入的理解它们为什么是那样——这种不求甚解的态度是需要改进的。像后端其实本来也是要分好几层的，而这次保障功能的实现而简省了一些规范。

希望在之后的学习中能够不断完善自己，为成为一个有用的人而努力。

10.对课程的建议

本课程安排非常灵活，允许学生根据自己的偏好选择技术栈，我觉得这是非常好的。

在开发的过程中是真的能够学到很多东西的，也值得我们在此之上花费很多精力。

只是如果提供的模拟物联网设备测试代码能够更加清晰一些、并且适当引入一些过程管理可能会更好。在过程管理上，作为一个长学期课程，对作业的要求感觉还是太灵活了，除了中期的一次设计报告之外，其实可以再安排一个阶段来把控进度。

同时，希望能够对 Docker 进行一些讲解。

11.致谢

这是我独立完成的第一个全栈开发项目，完成之后，心中无比自豪。虽是东施效颦，却仍收获良多。一步步地，我参与开发了一个又一个项目，加之这学期学到的一些操作系统和网络知识，我愈发觉得自己在叩响计算机专业领域的大门。所学从空中楼阁变成了落地的项目，这是很开心的事情。作为软件工程专业的学生，这一点尤为显著。

在我们的努力下，我们将建设出更多伟大的工程。

感谢在本项目开发过程中无私提供帮助的同学们。

感谢一路以来始终陪伴、提供建议的朋友们。

感谢老师一学期以来的付出和包容。