

中国大学生计算机设计大赛



软件开发类作品文档简要要求

作品编号： 2022057545

作品名称： R&D Management

作 者： 李帅 黎治跃 杨跃

版本编号： v1.0

填写日期： 2022.5.3

填写说明：

- 1、本文档适用于**所有**涉及软件开发的作品，包括：软件应用与开发、大数据应用、人工智能应用、物联网应用；
- 2、正文一律用五号宋体，一级标题为二号黑体，其他级别标题如有需要，可根据需要设置；
- 3、本文档为简要文档，不宜长篇大论，简明扼要为上；
- 4、提交文档时，以 PDF 格式提交本文档；
- 5、本文档内容是正式参赛内容组成部分，务必真实填写。如不属实，将导致奖项等级降低甚至终止本作品参加比赛。

目录

第一章 需求分析.....3

第二章 概要设计.....3

第三章 详细设计.....5

 第一节 系统框架设计5

 第二节 医院网站设计6

 第三节 药品管理系统设计8

 第四节 数据库设计9

 第五节 关键技术 13

第四章 测试报告..... 13

第五章 安装及使用 17

 一、前端打包 17

 二、后端打包 18

 三、上传到服务器 19

第六章 项目总结..... 23

第一章 需求分析

根据客户需求，我们的产品主体分为俩大模块，分别是医院网站以及医院的药品管理系统。下面是这两大模块内容的相关简述：

（一）医院网站

1. 首页：首页模块主要介绍了医院的服务项目以及医院的服务理念。
2. 动态分享：应客户需求，这一模块，主要对医院的动态进行实时展示，分别有医院的最新动态、工作招聘信息以及有关医院的最近的媒体新闻报道。
3. 医院党建：该模块主要介绍了医院内每月相关的党建活动。
4. 数据中心：为方便客户一览医院各方面信息，该模块设置成了数据大屏，主要介绍了医院的概况、科室的组成、医院管理以及医院评价。
5. 登录：该模块提供系统登录界面，方便客户访问医院药品管理系统。

（一）药品管理系统（仅向医院内部工作人员提供）

1. 用户管理：该模块仅向管理员提供，用于增加，删除，修改管理员或普通用户的信息。
2. 药品库存管理：该模块分为登记出入库信息、问题药品记录、药品保质期检查三个功能，都可对药品进行相关增加、删除及修改操作。
3. 药品销售管理：该模块可对药品销售记录、收到退货信息进行增加、删除及修改操作。
4. 药品明细查询：该模块可编辑以及删除相关的药品信息、供应商信息、退货返厂信息、账单信息。
5. 系统设置：该模块用于展示个人信息及、个人操作数据的日志以及退出系统功能。

第二章 概要设计

1. 模块功能层次图，如图 1 及图 2 所示：

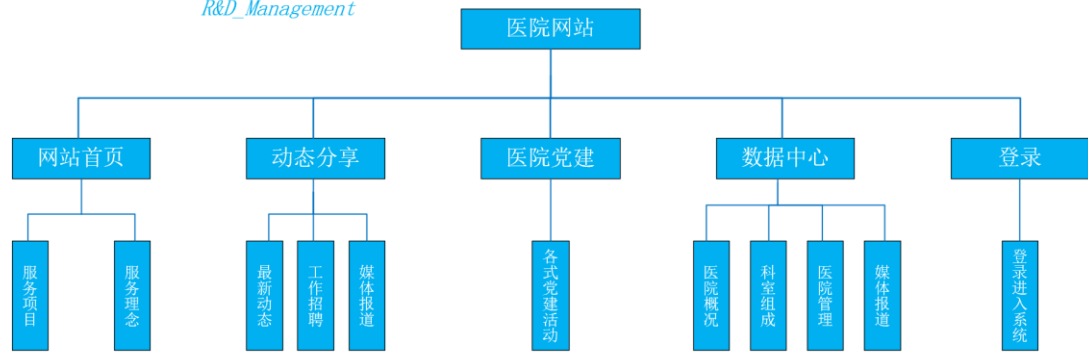


图 1-医院网站

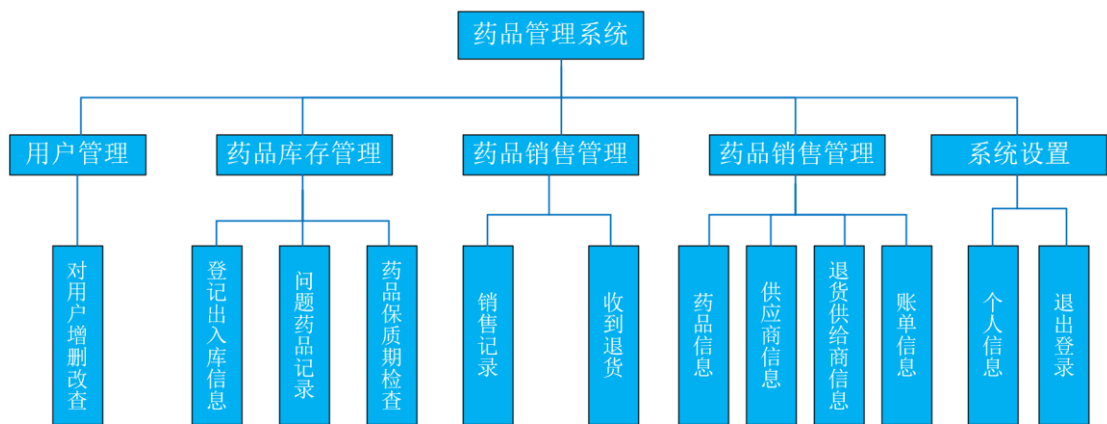


图 2-药品管理系统

2. 关系结构图，如图 3 所示：

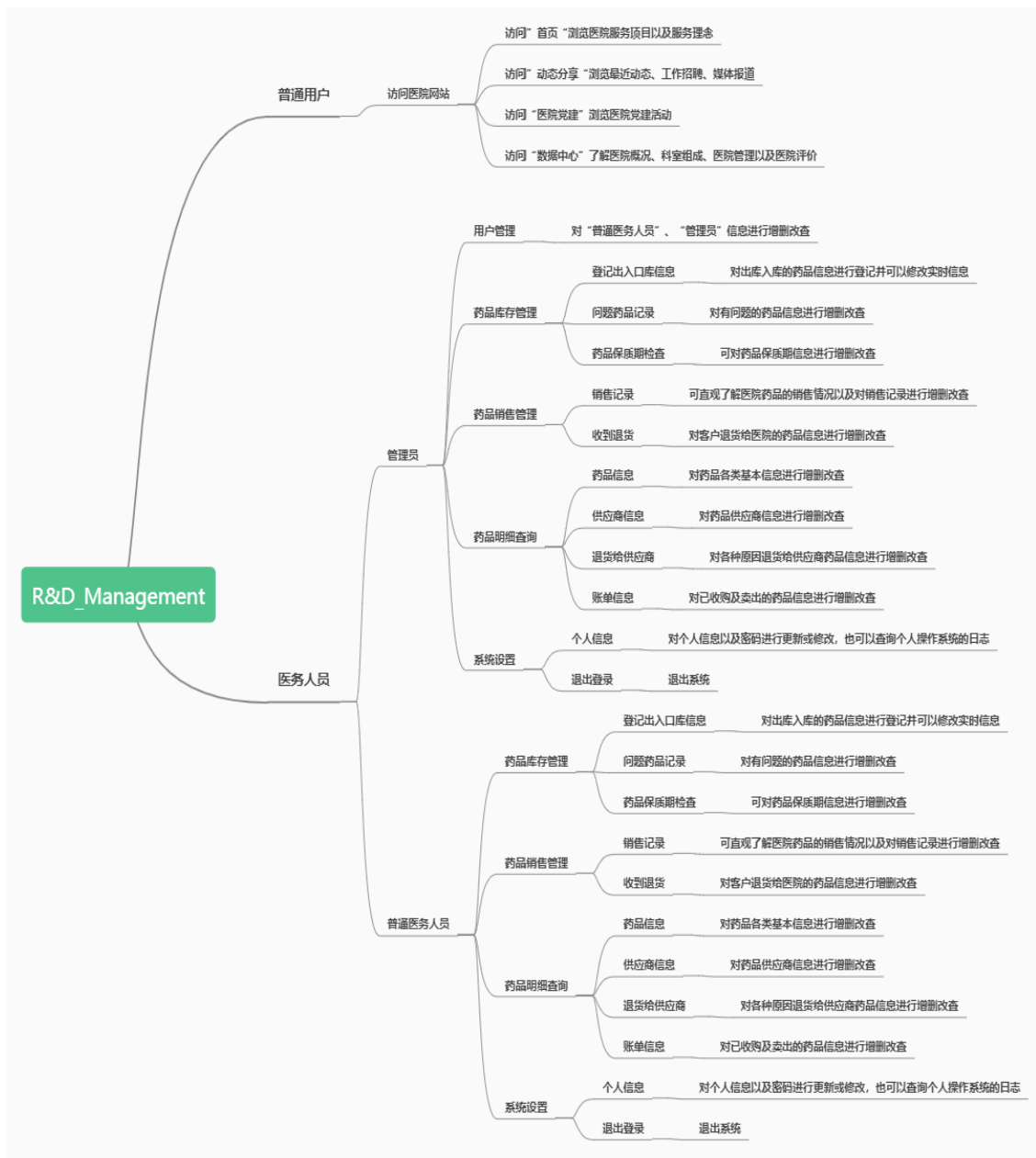


图 3-关系结构图

第三章 详细设计

本部分内容从医院网站设计、药品管理系统设计、数据库设计、关键技术等方面简要阐述本系统的实现过程。

第一节 系统框架设计

本系统前端展示为 web 页面，该部分提供网站页面供用户浏览；后端采用 Spring boot

框架整合 MongoDB、MySQL、Redis 数据库。如下图所示：

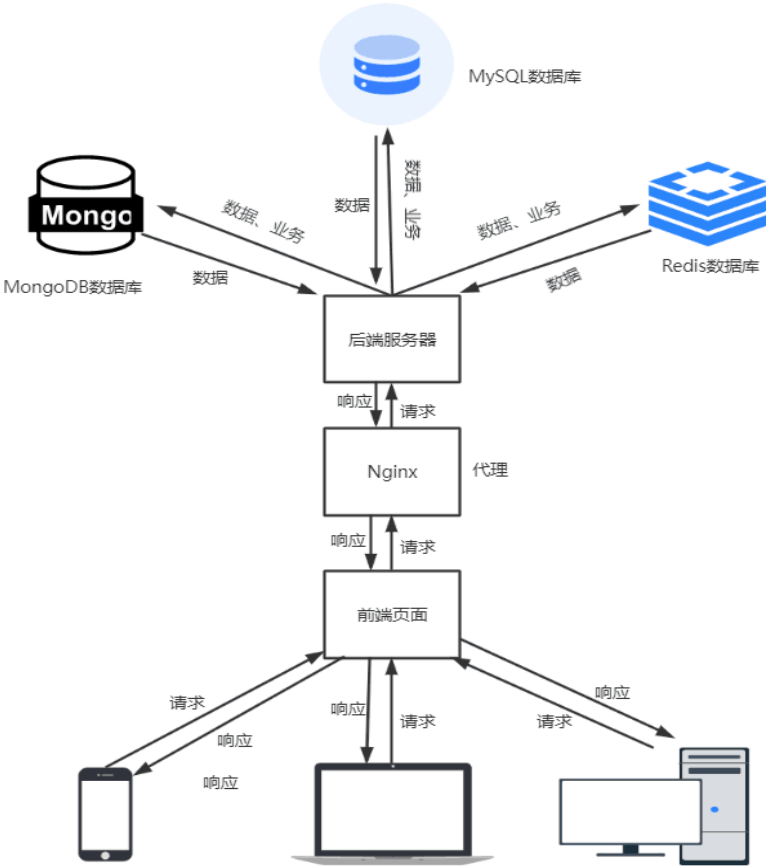


图 3-1 系统前后端交互图

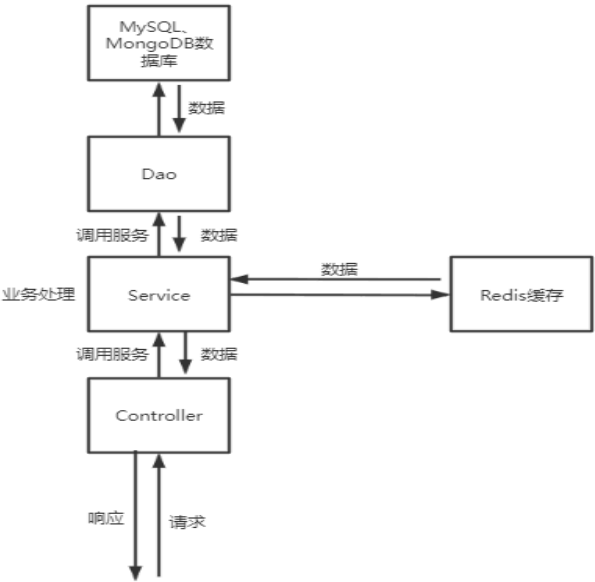


图 3-2 后端业务处理图

第二节 医院网站设计

用户可在首页中了解医院的服务项目以及服务理念如图 4 所示。转到动态分享中，用户可以浏览医院的最新动态、工作招聘以及媒体报道如图 5 所示。在数据中心界面，是比较有个性的，我们通过利用堆叠图、南丁格尔玫瑰图、漏斗图、雷达图以及仪表盘等数据图将医院概况、科室组成、医院管理以及医院评价组成数据大屏，让用户直观了解相关信息，如图 6 所示。转到医院党建界面，用户能够了解医院近期举办的所有红色活动，如图 7 所示。



图 4 首页



图 5-1 最近动态



图 5-2 工作招聘



图 5-3 媒体报导

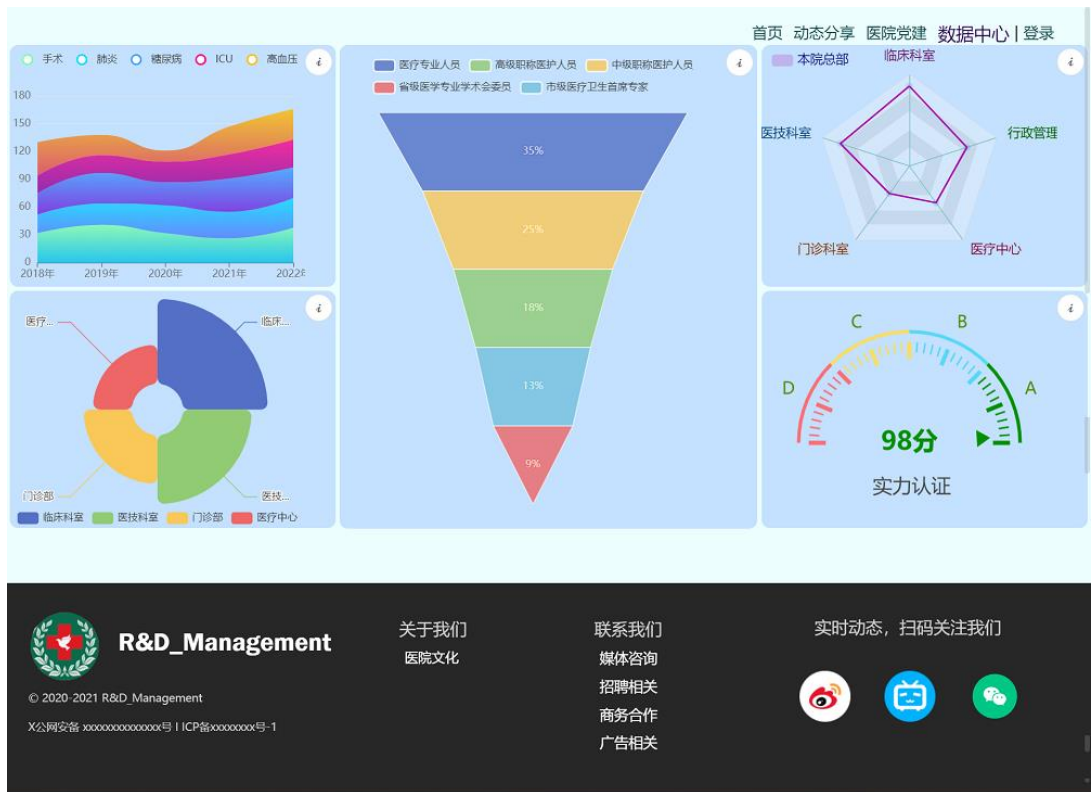


图 6 数据中心



图 7 医院党建

第三节 药品管理系统设计

药品管理系统面向的用户主要是医院内部的工作人员，分为两个类别，分别是管理员与普通医务人员。管理员可对管理员以及普通医务人员信息进行增删改查，如图 8 所示。其次管理员与普通医务人员在药品库存管理、药品销售管理、药品明细查询、系统设置中操作

权限是一致的，如图 9 所示。

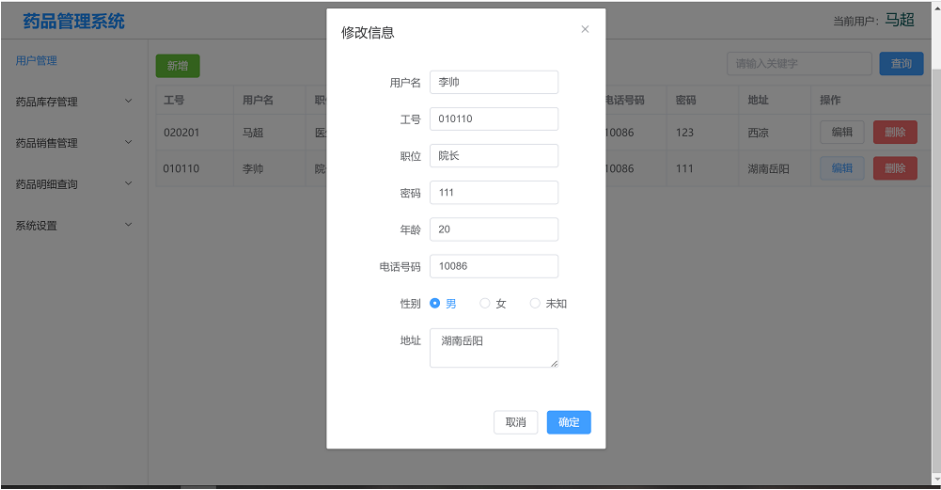


图 8

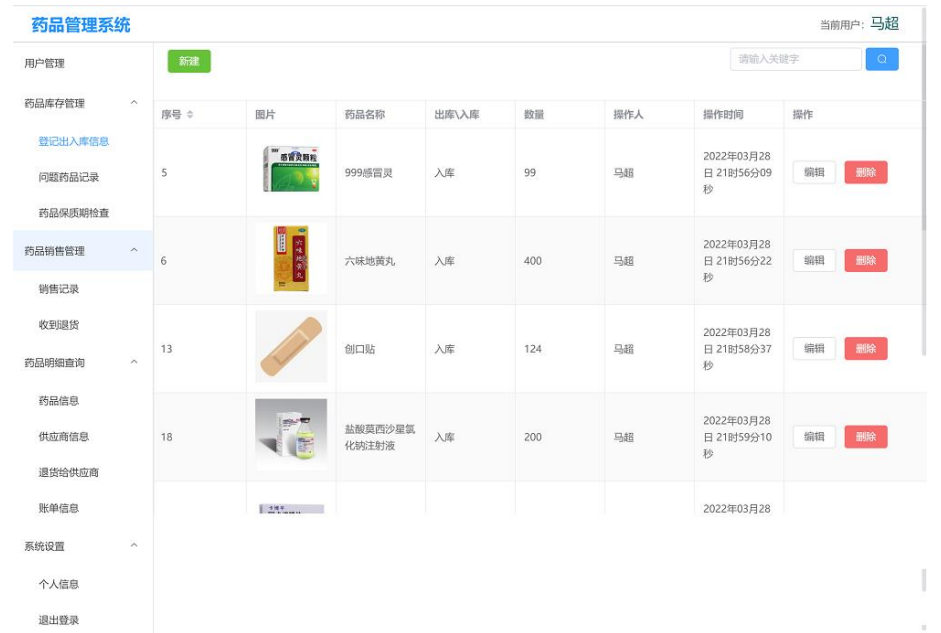


图 9

第四节 数据库设计

- 1) 医院网站：医院网站相关数据是利用 MongoDB 数据库进行存储，利用 BSON 数据结构灵活处理各类数据，如实例图 10，图 11 所示。

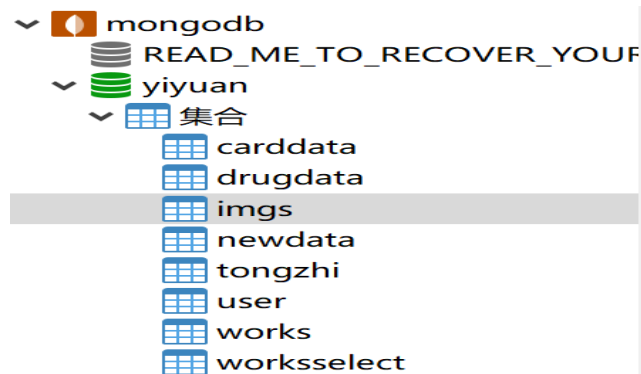


图 10

开始事务	文本	筛选	排序	全部折叠	类型颜色	导入	导出	分析
_id	url	text	_class					
623d94083b80a255b71dc316	http://124.222.131.59:8090/files/重点专科建设专题	com.example.entity.menhu.dongtaiimg.Imgs						
623d94423b80a255b71dc317	http://124.222.131.59:8090/files/呼吸与危重症医学	com.example.entity.menhu.dongtaiimg.Imgs						
623d94543b80a255b71dc318	http://124.222.131.59:8090/files/深度打磨专业教师	com.example.entity.menhu.dongtaiimg.Imgs						
623d946a3b80a255b71dc319	http://124.222.131.59:8090/files/学术交流	com.example.entity.menhu.dongtaiimg.Imgs						

图 11

2) 药品管理系统数据库设计:

a. ER 图设计:

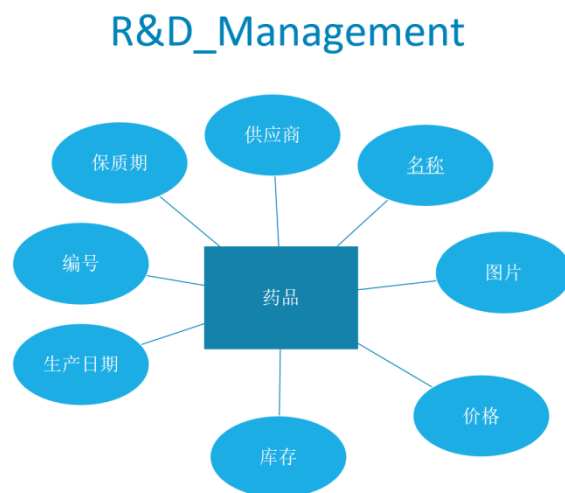


图 1-1 药品实例图

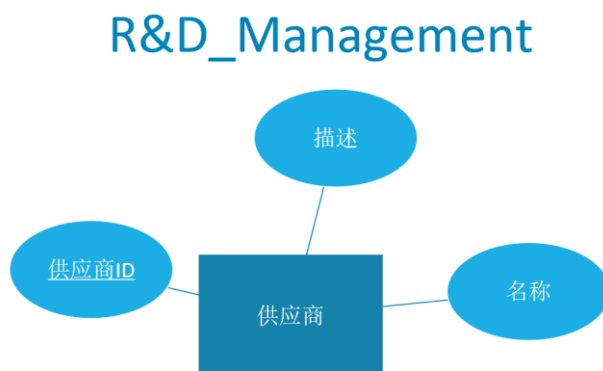


图 1-2 供应商实例图

R&D_Management

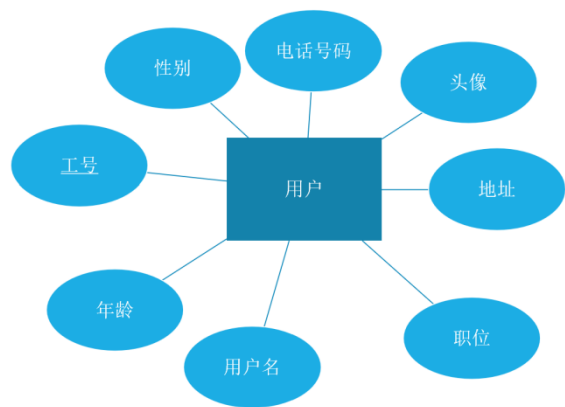


图 1-3 用户实例图

b. 数据库表:

表 1 bill

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
num	varchar(255)	not null	药品编号
drugname	varchar(255)	primary key, not null	药品名称
count	int(11)	not null	药品数量
tupian	varchar(255)	not null	图片
price	double(10)	not null	单价
createtime	date	not null	创建时间

表 2 drug

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
tupian	varchar(225)	not null	图片
drugname	varchar(225)	not null	药品名
outin	varchar(225)	not null	出入库
nums	int(255)	not null	数量
druguser	varchar(225)	not null	操作人
ctime	varchar(225)	not null	操作时间

表 3 druginfo

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
tupian	varchar(255)	not null	图片
drugname	varchar(255)	primary key, not null	药品名称
suppliername	varchar(255)	not null	供应商
producttime	date	not null	生产日期
warrenty	int(255)	not null	保质期

number	varchar(255)	not null	药品编码
price	double(10)	not null	价格
stock	int(255)	not null	库存
guo	tinyint(255)	not null	是否过期
guotime	varchar(255)	not null	过期时间

表 4 drugmatter

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
img	varchar(255)	not null	图片
drugname	varchar(255)	primary key, not null	药品名称
matternum	int(11)	not null	问题数量
price	int(10)	not null	单价
ctime	varchar(255)	not null	操作时间
miaos	varchar(255)	not null	描述

表 5 returngoods

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
drugname	varchar(255)	primary key, not null	药品名称
count	int(11)	not null	退货数量
reason	varchar(255)	not null	退货原因
money	double(10)	not null	总金额
operatetime	date	not null	操作时间
tupian	varchar(255)	not null	图片

表 6 returnsupplier

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
count	int(11)	not null	数量
tupian	varchar(255)	not null	图片
drugname	varchar(255)	not null	药品原因
reason	varchar(255)	not null	退货原因
createtime	date	not null	退货时间
pay	varchar(255)	not null	支付方式

表 7 saleinfo

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
drugname	varchar(255)	primary key, not null	药品名称
number	varchar(255)	not null	编号
count	int(11)	not null	数量

money	double(10)	not null	金额
operator	varchar(255)	not null	操作人
operatetime	date	not null	操作时间
tupian	varchar(255)	not null	图片

表 8 supplier

字段名	类型（长度）	约束	备注
id	int(11)	primary key, not null	id 号
name	varchar(255)	primary key, not null	供应商名称
content	varchar(255)	not null	供应商描述
createtime	date	not null	创建时间

第五节 关键技术

- 1) 医院网站：
 - a. 前端使用 Vue 框架整合 Echarts、Element Plus 第三方库。
 - b. 后端采用 Spring boot 框架整合了 Mongodb、Redis 数据库，第三方工具 Hutool、Lombok。
 - c. 对整个医院网站使用了 Redis 缓存技术。
- 2) 药品管理系统：
 - a. 前端使用 Vue 框架整合 Echarts、Element Plus 第三方库。
 - b. 后端采用 Spring boot 框架整合了 Mongodb 数据库，Mybatis Plus 持久层框架，第三方工具 Lombok、Hutool。
 - c. 在药品搜索功能使用了 Mybatis-Plus 的模糊查询。
 - d. 我们的信息管理系统，使用了 Mongodb 做的日志记录，操作人若是对数据操作错误，后期可对数据进行恢复
 - e. 前端加入了多重权限验证，在页面加载时以及发起请求时验证权限。

第四章 测试报告

1.测试方案

项目在开发过程中采用分阶段的单元测试的方法进行测试工作，全面覆盖整个项目开发过程。在每个模块开发完成后分别进行了黑盒测试和白盒测试。黑盒测试主要针对网站的页面布局，功能实现，逻辑判断进行测试；白盒测试主要以代码对网站各个业务需求进

行逻辑覆盖以及接口测试等等。另外还要进行集成测试和验收测试以此来排查网站开发过程中的不足之处。

2.测试项目

主要围绕界面测试，安全性测试，功能测试和兼容性测试展开。

界面测试包括：

- 页面布局
- 控件使用
- 图片展示
- 文本信息

表-2.1 界面测试

序号	测试用例	操作	预期结果	实际结果
1	页面布局	任意缩小放大网站界面	页面响应式改变，同时各个模块的功能能够正常使用	模块能正常使用，界面比列会失调
2	控件使用	对任意功能按钮进行操作	功能按钮正常使用，布局合理，简单以操作，部分输入框对应的提示信息	控件使用正常，操作功能良好，交互性可观
3	图片展示	访问网站的各个界面	图片素材能够及时加载出来	受服务器带宽和网络影响，有时候会出现显示不及时
4	文本信息	访问网站的各个界面	文本信息显示正常，且无错别字信息，用户体验良好	文本信息显示正常，无错别字情况，交互体验良好

安全性测试包括：

- 管理员登陆
- 管理员信息查看

序号	测试用例	操作	预期结果	实际结果
1	管理员登录	输入正确的账号密码进行登陆操作	管理员登陆成功并可以天转到后台管理界面	管理员登陆成功并可以天转到后台管理界面
2	管理员信息查看	登陆成功后进入查看详情页	登陆成功的管理员可以访问后台信息，否则会提示重新进行登录	登陆成功的管理员可以访问后台信息，否则会提示重新进行登录

表-2.2 安全性测试

功能测试包括：

- 新建药品
- 查询药品
- 修改个人信息
- 删除目标信息

序号	测试用例	操作	预期结果	实际结果
1	新建药品	输入正确的相关信息点击新增	在展示页面能看到相关药品信息，且同步到数据库上	在展示页面能看到相关药品信息，且同步到数据库上
2	查询药品	输入想要查询的药品名或部分字段模糊查询	在展示页面能看到查看目标药品的信息	在展示页面能看到查看目标药品的信息
3	修改个人信息	正确输入修改个人信息的修改项	修改保存后个人信息成功被修改，且个人信息能及时同步到数据库上	修改保存后个人信息成功被修改，且个人信息能及时同步到数据库上

序号	测试用例	操作	预期结果	实际结果
4	删除目标信息	进行删除操作	点击删除按钮后有相关的提示信息，并且正确操作后结果能同步到数据库上	点击删除按钮后有相关的提示信息，并且正确操作后结果能同步到数据库上

表-2.3 功能测试

兼容性测试包括：

- 浏览器兼容性
- 系统兼容性

序号	测试目标	测试用例	预期结果	实际结果
1	浏览器兼容性	分别使用 FireFox, Chrome, 360, IE 等浏览器访问 yiyuan 网站	在各个浏览器中兼容性良好	360、Chrome 等浏览器兼容性良好 IE、会出现部分模块不兼容情况
2	系统兼容性	分别使用 Windows10, Windows11, MacOS, Android, Linux 等不同的操作系统访问 yiyuan 网站	在各个操作系统中兼容性良好	Windows10, Windows11, MacOS, 系统中访问良好，但是在 Android 上由于不是响应式界面，模块分布很不均匀，Linux 系统上由于服务器带宽较低的原因也有一定不兼容的情况

表-2.4 兼容性测试

3 .测试中遇到的问题及解决方案

序号	问题描述	影响	解决方案
1	在信息管理系统中，点击退出以	信息安全受到了影响，其他	在页面初始化的时候，先查看缓存中是否有当前用户信息，在发起请求的时候也需要判

序号	问题描述	影响	解决方案
	后，点击浏览器的回退按钮，又回到信息管理界面	人有机会恶意操作数据	断缓存中是否有用户信息，用户退出的时候，清除缓存
2	数据库中的数据改变后，界面上的数据还没有改变	后端数据和前端数据没法得到同步，用户无法查看到最新的数据	这一原因是因为我们引入了 Redis 缓存造成的，数据库的数据修改了，没有更新缓存的数据，我们在业务层对数据修改的操作上都会将 redis 缓存的数据删除，并且，我们也修改了缓存的机制，每一条数据都设置了过期时间，最大限度保证了 MySQL 或者 MongoDB 的数据与 Redis 缓存数据同步

第五章 安装及使用

当前软件编写基本都是用 java 语言来完成，本系统也是是用 java 语言来完成的。由于 java 语言的高可移植性和跨平台能力，从而使得了应用程序能够“一次编译，处处运行”。由于本项目最终是运行在云服务器上的，所以先做一些安装前的准备。

需要用到的软件：Navicat Premium：，用这个软件可以方便我们将数据库里面的数据添加到云服务器的数据库上。FinalShell：用于连接云服务器，方便上传文件和操作云服务器。服务器环境：jdk8: java 环境 nginx: 用于部署前端项目 MongoDB: 数据库 MySQL: 数据库 Redis: 缓存数据库

一、前端打包

1.1 测试 node 和 npm 是否成功 打开黑窗口（win 键+R）点击:命令提示符管理员（必须以管理员运行）。输入 `node -v npm -v`，出现版本号为成功。

1.2、安装完 node 以后，黑窗口进入 vue 项目根路径下。

运行命令：`npm install`（下载必须的依赖，如果报错了就是 node 环境有问题，查百度重新搭环境）运行完上面的命令以后在执行：`npm run build`（这个命令是打包）出

现这个说明已经打包成功：

```
static/plugins/ueditor-1.4.3.3/dialogs/scrawl/images/eraser.png 43.3 kB
static/plugins/ueditor-1.4.3.3/dialogs/scrawl/images/empty.png 519 bytes
static/plugins/ueditor-1.4.3.3/dialogs/scrawl/images/emptyH.png 657 bytes
static/plugins/ueditor-1.4.3.3/dialogs/scrawl/images/delimgH.png 578 bytes
static/plugins/ueditor-1.4.3.3/dialogs/scrawl/images/delimg.png 516 bytes
static/plugins/ueditor-1.4.3.3/dialogs/scrawl/images/brush.png 608 bytes
static/plugins/ueditor-1.4.3.3/dialogs/scrawl/images/adding.png 628 bytes
static/plugins/ueditor-1.4.3.3/dialogs/music/music.js 8.01 kB
static/plugins/ueditor-1.4.3.3/dialogs/preview/preview.html 1.22 kB
static/plugins/ueditor-1.4.3.3/dialogs/music/music.html 981 bytes
static/plugins/ueditor-1.4.3.3/dialogs/music/music.css 1.71 kB
static/plugins/ueditor-1.4.3.3/dialogs/map/show.html 5.08 kB
static/plugins/ueditor-1.4.3.3/dialogs/map/map.html 6.15 kB
static/plugins/ueditor-1.4.3.3/dialogs/link/link.html 4.51 kB
static/plugins/ueditor-1.4.3.3/dialogs/internal.js 2.68 kB
static/plugins/ueditor-1.4.3.3/dialogs/insertframe/insertframe.html 4.38 kB

Build complete.

Tip: built files are meant to be served over an HTTP server.
Opening index.html over file:// won't work.

[13:23:46] Finished 'build' after 41 s
[13:23:46] Starting 'create:versionCatalog'...
[13:23:47] Finished 'create:versionCatalog' after 721 ms
[13:23:47] Starting 'replace:cdnUrl'...
[13:23:47] Starting 'replace:version'...
[13:23:47] Finished 'replace:version' after 12 ms
[13:23:47] Starting 'concat:config'...
[13:23:47] Finished 'concat:config' after 222 ms
```

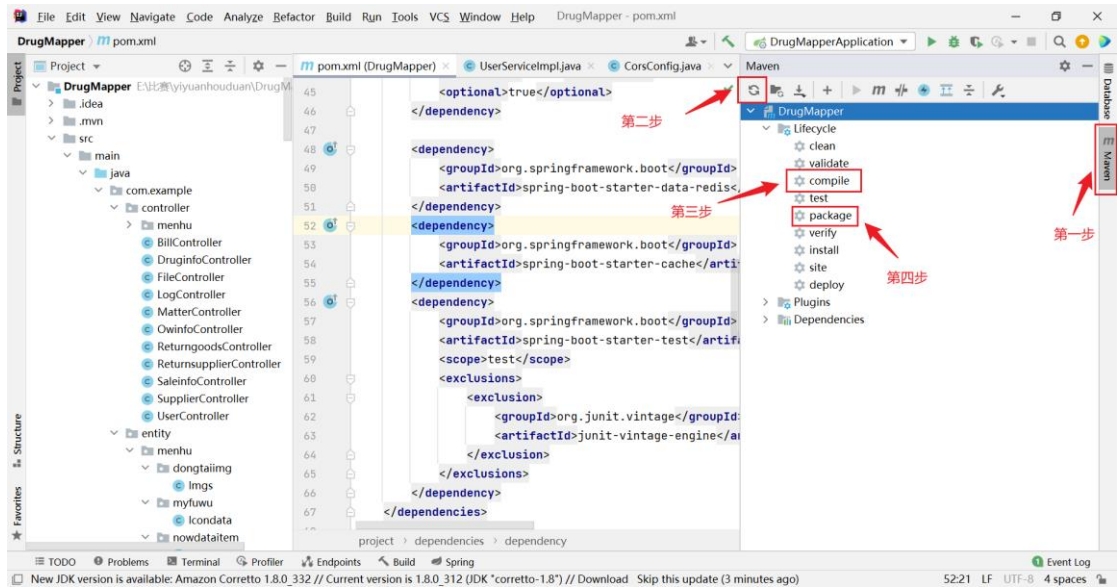
成功以后你的 vue 项目下会出现一个 dist 的文件夹。

名称	修改日期	类型	大小
.git	2018/8/1 13:43	文件夹	
build	2018/7/31 10:12	文件夹	
config	2018/7/31 10:12	文件夹	
demo-screenshot	2018/7/31 10:12	文件夹	
dist	2018/8/1 13:23	文件夹	
node_modules	2018/7/31 11:15	文件夹	
src	2018/7/31 10:12	文件夹	
static	2018/7/31 10:12	文件夹	
test	2018/7/31 10:12	文件夹	
.babelrc	2018/7/30 17:23	BABELRC 文件	1 KB
.editorconfig	2018/7/30 17:23	EDITORCONFIG ...	1 KB
.eslintignore	2018/7/30 17:23	ESLINTIGNORE ...	1 KB
.eslintrc.js	2018/7/30 17:23	JS 文件	1 KB
.gitignore	2018/7/30 17:23	文本文档	1 KB
.postcssrc.js	2018/7/30 17:23	JS 文件	1 KB
CHANGELOG.md	2018/7/30 17:23	MD 文件	2 KB
gulpfile.js	2018/7/30 17:23	JS 文件	3 KB
index.html	2018/7/30 17:23	360 Chrome HT...	2 KB
LICENSE	2018/7/30 17:23	文件	2 KB
package.json	2018/7/30 17:23	JSON 文件	4 KB

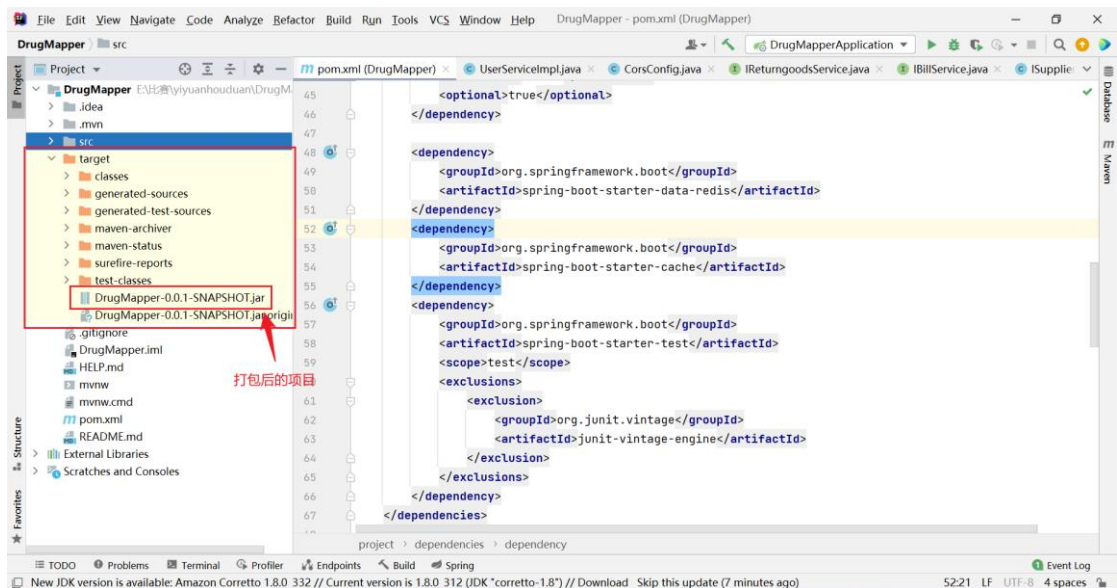
打包好 vue 项目以后，留着 dist 文件一会用到。

二、后端打包

打开 IDEA，点击右侧的 Maven，防止依赖没有加载上，所以先刷新依赖，再点击 compile 进行编译，依赖加载完成以后点击 package 进行打包。



完成上面步骤以后，在项目会出现一个 target 目录，target 以 .jar 后缀结尾的文件就是已经打包完成的项目。

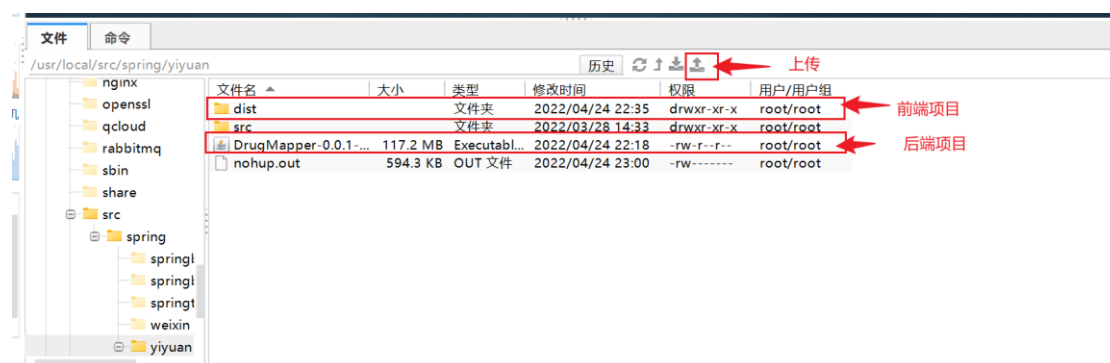


三、上传到服务器

使用 FinalShell 连接服务器（我们组自己的服务器是腾讯云服务器，Linux 版本是 CentOS7.8，控制面板使用的是宝塔控制面板，接下来的演示都是在腾讯云服务器上进行），在服务器上使用 mkdir 命令在合适的地方建一个文件夹用于存放项目。

```
连接主机...
连接主机成功
Last failed login: Thu Apr 28 12:48:41 CST 2022 from 223.71.51.178 on ssh:notty
There were 3710 failed login attempts since the last successful login.
Last login: Tue Apr 26 16:44:58 2022 from 223.104.20.54
[root@VM-16-7-centos ~]# cd /usr/local/src/spring/
[root@VM-16-7-centos spring]# mkdir yi yuan
```

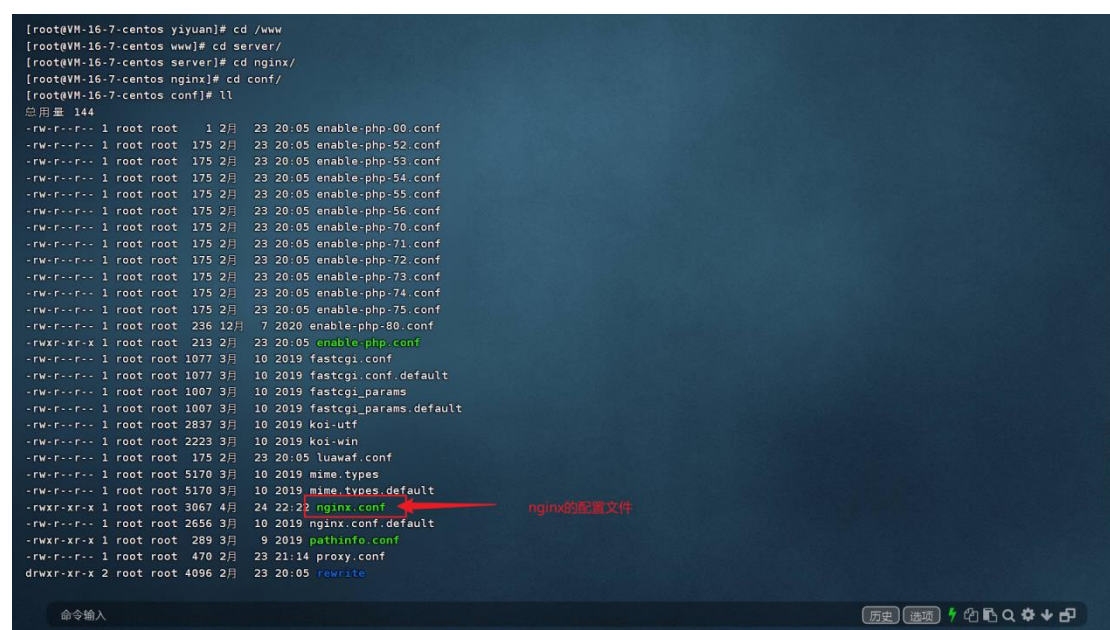
使用 cd 命令打开该目录，使用 FinalShell 的上传文件功能，将前端项目和后端项目上传到服务器。由于我已经提前部署过，所以该目录下有其他文件



使用 Navicat Premium，把 SQL 文件导入到服务器的数据库中，接着使用 nohup java jar 命令启动后端，若出现下面的结果则启动成功。（nohup：表示我们使用的是后台运行的方式启动的后端）

```
[root@VM-16-7-centos yiyuan]# nohup java -jar DrugMapper-0.0.1-SNAPSHOT.jar
nohup: 忽略输入并把输出追加到 "nohup.out"
```

接着找到 nginx 的配置文件



使用 vim 命令打开，nginx.conf 文件，在 http 节点下新增一个 server 节点


```

http
{
    include mime.types;
    #include luawaf.conf;

    include proxy.conf;

    default_type application/octet-stream;

    server_names_hash_bucket_size 512;
    client_header_buffer_size 64k;
    large_client_header_buffers 4 32k;
    client_max_body_size 50m;

    sendfile on;
    tcp_nopush on;

```

```

server {
    listen 8877;
    server_name 124.222.131.59;

    location / {
        root /usr/local/src/spring/yiyuan/dist;
        index index.html index.htm;
        try_files $uri $uri/ /index.html;
    }
    location /proxy{
        proxy_pass http://localhost:8090/;
    }
}

```

前端端口号

前端项目的路径

前端项目的入口文件

代理的IP地址和端口号，也就是后端服务器

这里配置完了以后，进入宝塔控制面板，重启 nginx



在宝塔控制面板中开放前端端口号。

第六章 项目总结

在准备此项目前期，寒假中我们小组成员与指导老师在线上多次讨论，集中头脑风暴。在指导老师的协助下，我们定好了此项目的主题。接下来我们开始去做需求分析。首先由李帅去做市场调研。他走访了他所在地方的几个医院，参考了这几个医院中的管理系统，同时重点放在了药品管理的部分上面，经采访得到了不错的学习。由于我们项目准备的时间较早，所以为了提高项目软件的质量与效率，我们采用了瀑布模型进行开发。

黎治跃作为组长负责分解任务，并把任务下达给小组的其他成员。李帅管理计划时期、黎治跃管理开发时期、杨跃管理运行维护时期。我们互相合作，共同进退，很深刻的体会到了团队的力量。在项目前中期处于寒假的时候，我们会每隔三天进行一次视频会议，线上讨论最近的进展和遇到的难题，开学后我们便进入了编码阶段。以为团队成员都是同班同学，在每天空闲时间都会聚在一起讨论项目的内容，比如说采用什么技术，使用什么第三方的工具，有时候也会相互解决 bug。一是营造团队合作的氛围，增加小组成员的项目实战经验，二是有效推动项目的有序开展。

此次开发遇到过诸多实际困难，而且这些问题是书本上没有讲到的，需要我们去查阅资料，去摸索解决办法，有一些难题还是网上没有相似案例的。例如：MySQL 数据库与 redis 缓存数据库之间的数据同步，我们为了提高网站的响应速度，使用了缓存技术，但是没有想到使用缓存之后带来了个新的问题，数据库数据更新了，但是界面的数据没有更新，在网上查阅一些资料之后，也找到了很多的解决办法，比如使用一个消息中间件，控制缓存的刷新，或者是延迟双删，还有一种就是使用人工删除。我们的解决方案是，在缓存数据库中，我们设置了数据过期时间。在业务层，当触发数据更改方法之后，同时也会触发缓存数据库刷新的方法。

用户登录系统之后，然后再点退出，用户将会回到登录页面，但是若用户点击浏览器之类的回退按钮，又会回到用户未退出的页面。这是一个非常严重的安全问题。若是有人恶意操作数据的话，会造成数据丢失。我们也是第 1 次做 Spring boot+vue 的前后端分离项目，所以就没有考虑到用户信息的过期处理。对于这个问题，我们最先想到的解决方案是在后端使用 redis 缓存来保存用户信息，当用户退出的时候，清除缓存中的数据，在页面中页面的创建会检测用户的信息，若没有用户信息会强制跳转到登录界面。但是这种方法也有很多缺点，在这大数据的时代，任何服务器都会承担很大的压力，我们为了尽可能的减少服务器的压力，所以我们使用了浏览器的缓存区，存储用户信息。当用户退出的时候，将会清除缓存中的数据。页面每一次创建都会检测用户信息是否存在，若是不存在，则跳到登录界面，并且在每一次发起请求的时候，都会检测用户信息是否存在，若不存在，则强制跳转到登录界面。

经过此次开发我们更加坚定了一个信念，团队的力量是巨大的，团队成员互相信任、互相沟通、互相合作是项目成功的钥匙。