

南京邮电大学

专业课程设计 II 报告

(2022 / 2023 学年 第 一 学期)

题 目： 高校学生家长关注
信息推送系统

专 业	计算机科学与技术
学 生 姓 名	
班 级 学 号	
指 导 教 师	张琳
指 导 单 位	计算机科学与技术系
日 期	2022.10.24-2022.11.4

课程目标	评价准则	计分（每项 10 分）
课程目标 1：通过课程设计提高学生的理论联系实际、查阅资料、自学和独立分析与解决复杂工程问题的能力（30 分）	1、能够掌握计算机软件设计的相关基础知识，并能够针对求解的工程问题，进行合理的分析与设计。	
	2、能够给出数据结构和算法的设计描述，给出关键算法的流程图或伪代码，并给出各算法之间的结构关系描述。	
	3、具备一定的人机交互设计意识，人机交互设计合理、友好，操作简便。	
课程目标 2：要求学生在深入理解软件系统的相关知识基础上，综合运用所学的方法和理论自主地完成课程设计。要求学生通过软件课题的开发实践活动熟悉一种主流的数据库管理系统产品和软件开发工具，掌握软件设计和开发的理论方法与实施步骤。要求学生在软件开发的过程中考虑到可持续发展的因素，例如安全以及环保等。（30 分）	4、能够结合计算机软硬件资源，合理选用算法、数据结构、数据存储方式等技术手段，对求解的工程问题进行有效建模和求解。	
	5、能够选择合适的程序设计语言与软件开发平台，对求解的工程问题进行编程实现。掌握调试方法与工具，对程序开发过程中出现的问题进行分析、跟踪与调试，并能够进行充分测试。	
	6、在完成课题要求的各项任务和指标时能够充分考虑可持续发展因素，例如安全和环保等。	
课程目标 3：要求学生完成一次项目设计与开发的全过程，掌握软件需求分析、软件设计、软件测试和团队协作开发的基本理论和技术实践。应用工程项目管理原理和经济决策方法划分和管理课题任务，培养项目组织管理能力。（20 分）	7、能够正确理解课题目标任务，并做好需求分析、软件设计、软件测试等关键环节的实践工作，具备团队协作、协作和组织管理能力。	
	8、具备一定的语言表达能力与文字处理能力，能够结合复杂工程问题撰写报告，报告内容和实验数据详实，格式规范。	
课程目标 4：要求学生能够通过参与项目设计与开发的全过程，建立解决实际问题需要不断探索和学习的主动性意识，能够结合实际问题的特征需求，通过自主学习、终生学习，拓展自身的知识和能力。（20 分）	9、具备一定自学能力与探索创新意识，能够充分利用教科书及其资源（如网络等）自学新知识与新技能，以解决实际问题。	
	10、能够正确、完整地回答指导教师关于课题开发全过程的问询，能够掌握解决具体问题的一般性规律和方法，具备自主学习、终生学习的意识。	
专业课程设计 II 能力测评总分		
指导教师：_____年__月__日		
备注		

高校学生家长关注信息推送系统

一、课题内容和要求

主要功能：

实现学生在校个人信息以及与该生相关的专业建设、学科竞赛、国际交流以及各类考试等重要通知、热点事件和重要新闻等自动推送给学生家长。系统的主要功能包含：

(1) 用户管理与登录模块：用户注册，设置与修改密码，系统管理员、部门管理员以及学生及其家长具有不同的权限和功能；

(2) 人员信息管理：学生及其家长的基本信息；

(3) 推送消息管理：包括消息的编辑与审核；个人消息的推送与查询；或者通过微信公众号进行推送。

(4) 部门消息的审核与发布：各类考试、学科竞赛和专项工作的审核（可区分后台自动审核和人工审核）、发布和定向推送；

(5) 学校公共信息的审核与发布：重大新闻和热点事件等。

(6) 统计每条热点的关注度，将关注度较高的多条进行群发推送。

(7) 要求：根据实际情况，自行展开需求分析，完善上述未列出的其他功能；建议用报表的形式完成统计功能。

二、需求分析和总体设计

2.1 本课题的主要功能包括：

(1) 登录功能，不同角色（学生，家长，部门管理员和系统管理员）登录对应进入不同的页面。

(2) 学生、家长功能模块，登陆后进入可以进行对账号信息（包括个人身份信息，登录密码）的修改，查看学校推送的单发和群发消息以及对消息的点赞功能。

(3) 部门管理员功能模块，登陆后进入可以进行对账号信息（包括个人身份信息，登录密码）的修改，以及管理学生家长账号信息的修改，增加，删除；可以申请推送信息，查看自己申请推送信息的状态，修改以及删除推送信息。

(4) 系统管理员功能模块，登陆后进入可以进行对账号信息（包括个人身份信息，登录密码）的修改，以及所有账号信息的修改，增加，删除；可以审核部门管理员申请的推送信息。

用例图如图 1 所示：

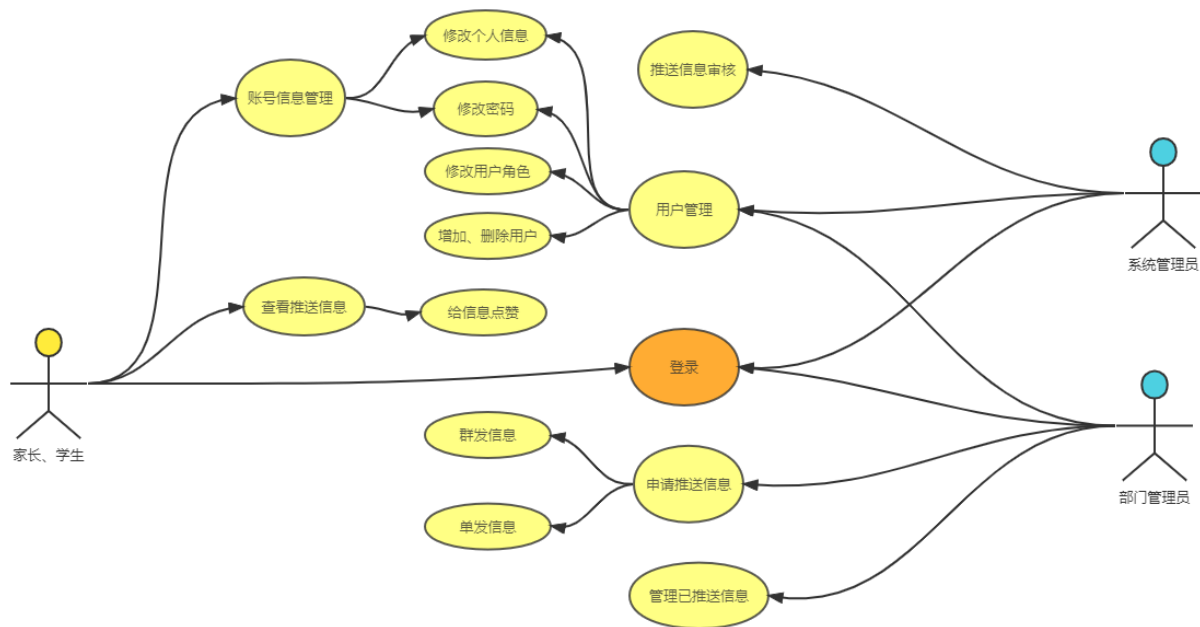


图 1 系统用例图

2.2 本课题的数据表单设计

关系模式集为：

学生和家長（用户 id，用户名，登录密码，邮箱，电话）

管理员（用户 id，用户名，登录密码，邮箱，电话，角色）

推送消息（消息流水号，作者，编辑时间，标题，内容，类型）

申请审核（消息流水号，用户 id，推送模式，推送对象，审核状态，审核理由）

关系模式对应 ER 图如图 2 所示：

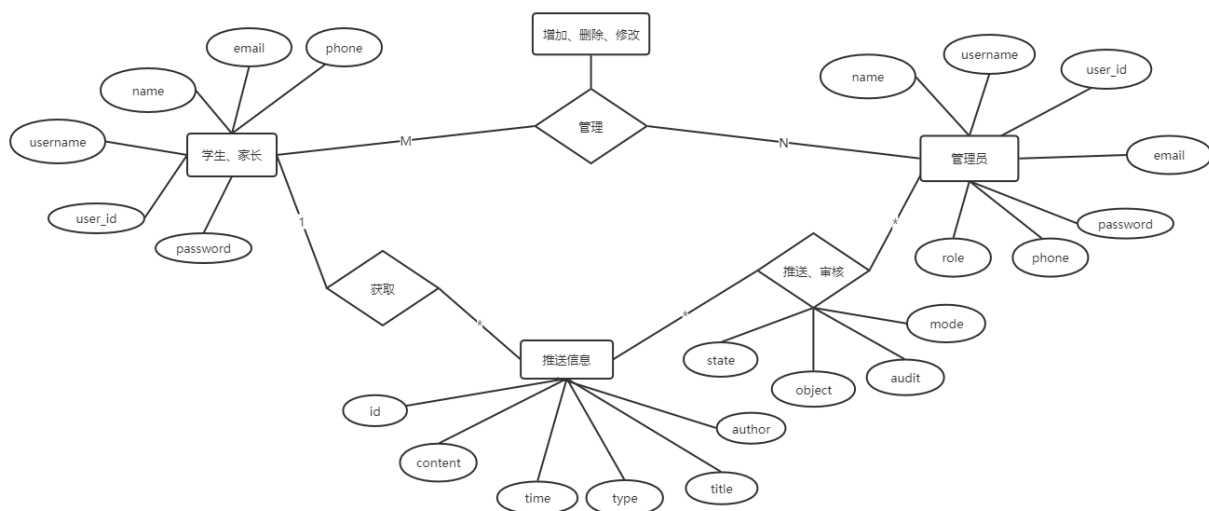


图 2 数据库设计 ER 图

2.3 本课题的体系结构设计

本系统形式为 Web 应用，采用 MVC（Model, View, Control）架构，分为模型层，视图层以及控制层。模型表示业务规则，有最多的处理任务。被模型返回的数据是中立的，模型与数据格式无关。在本项目中主要负责数据库的操作，为控制层提供数据存取方法。控制器是指控制器接受用户的输入并调用模型和视图去完成用户的需求，它只是接收请求并决定调用哪个模型构件去处理请求，然后再确定用哪个视图来显示返回的数据。在项目中即为根据业务需求对数据库中的数据进行处理返回给视图层以给用户交互。视图是用户看到并与之交互的界面，在本项目中就是处理用户的交互操作，将用户的请求传递至控制层，并呈现请求处理结果样式。

本系统中数据库采用 MySQL；模型层和控制层采用 java 结合 Spring Boot 框架实现，采取基于内存实现的键值型非关系（NoSQL）数据库 Redis 实现对 session 登录的 token 验证，以保证不会有多人同时登陆同一账户，利用 spring security 实现开放端口的权限认证以保证端口的访问安全，这部分被称作后端；视图层采取 HTML，以 Vue 前端 JavaScript 语言框架结合 Vuex 状态管理模式库，采取 MVVM（Model, View, View-Model）页面架构，以实现数据变化自动更新绑定的 View 以及 View 的操作更新绑定的数据，利用 Vue Router 实现页面的动态路由挂载，实现页面和用户权限的分离，进一步保证用户权限和用户视图的控制。用 Element UI 组件库以实现页面组件的可重用性，这部分被称作前端。前后端实现分离，利用 Axios 的 http 封装请求库以 http 网络通信协议的方式传递 json 数据文件，实现前后端的信息传递和交互。

本系统框架结构如图 3 所示：

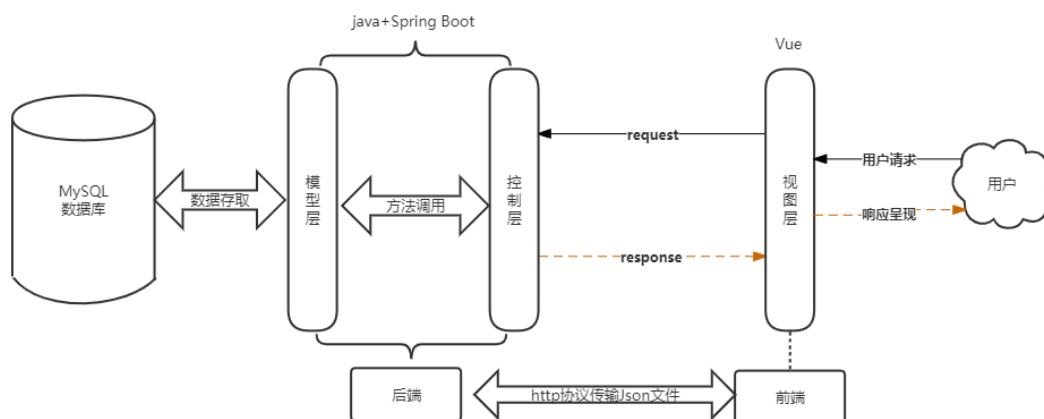


图 3 系统架构

前后端的项目文件结构分别如图 4，图 5 所示

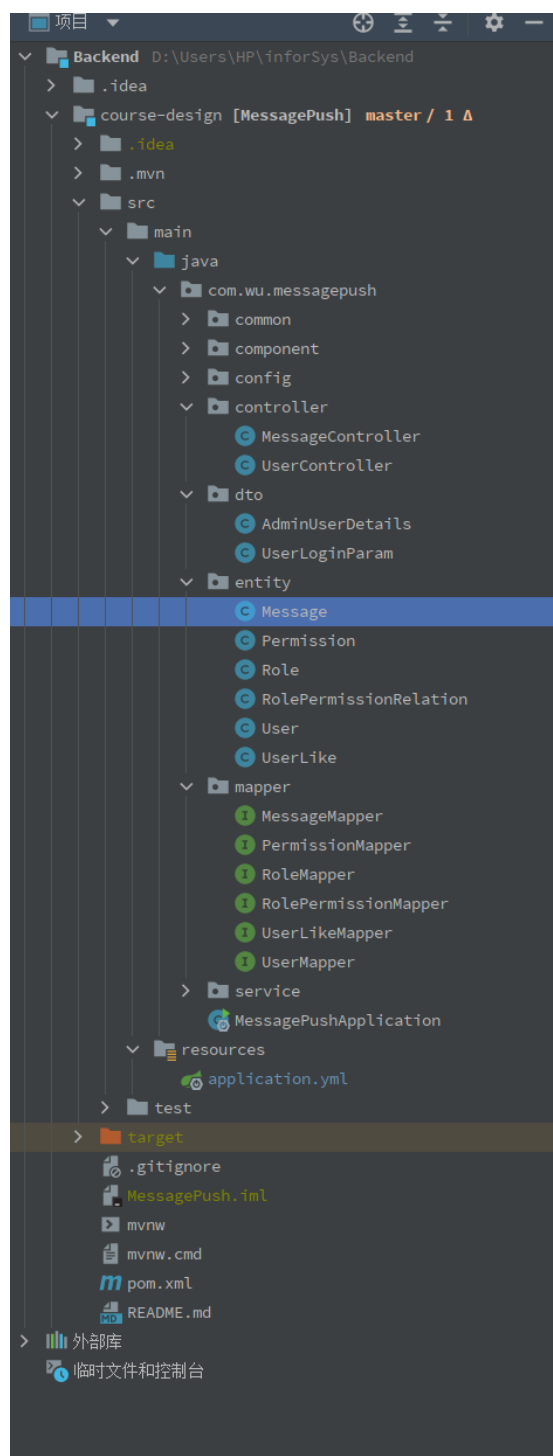


图 4 后端项目文件结构

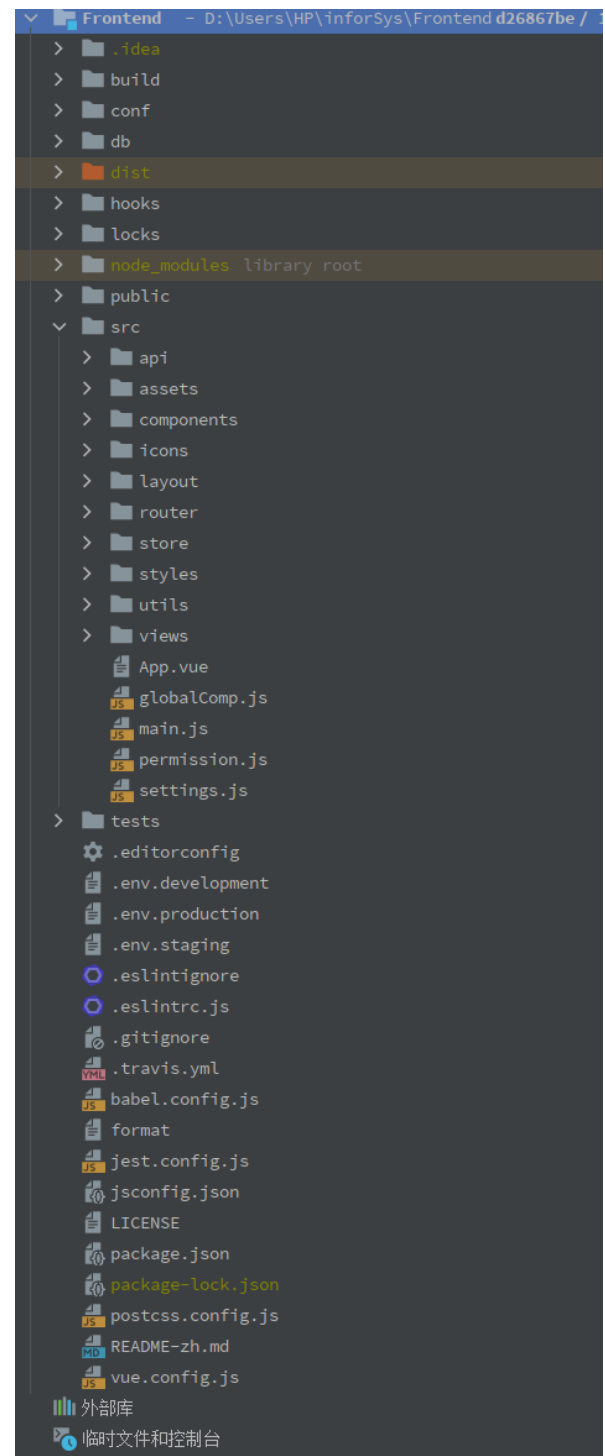


图 5 前端项目文件结构

2.4 本课题的主要功能界面设计

本课题登录界面，主要界面的 UI 设计原型如图 6，图 7 所示，主要页面形式详见第 5 小节软件测试与分析。

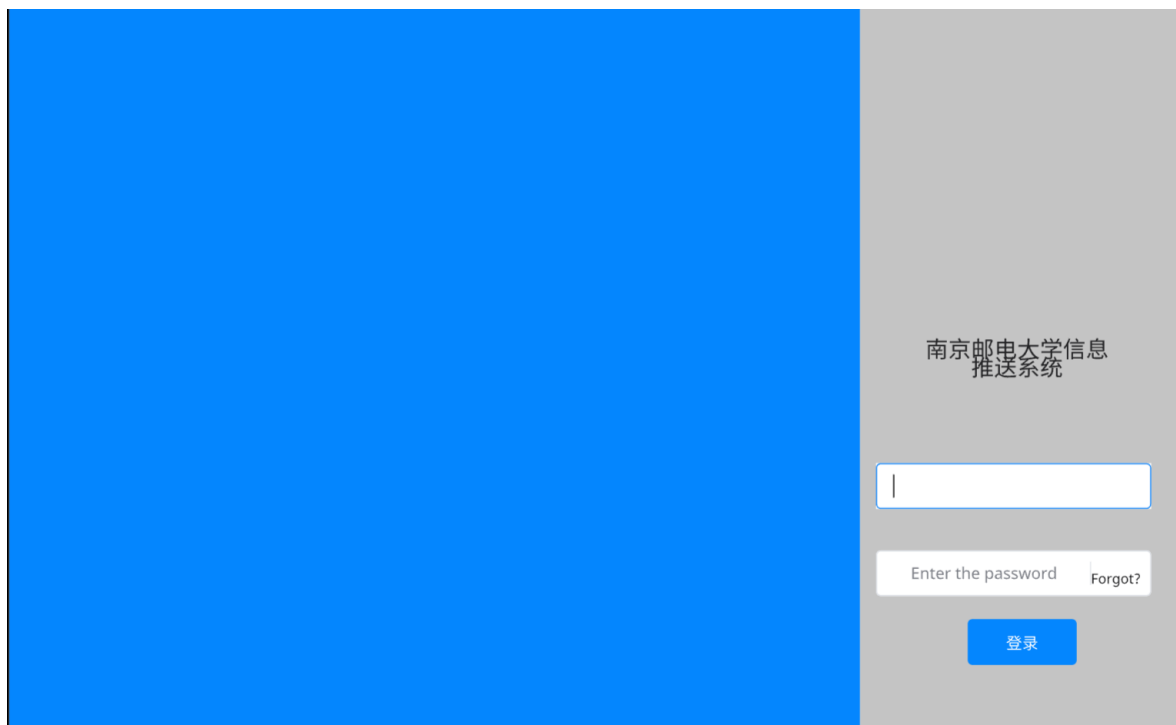


图 6 登录界面 UI 设计

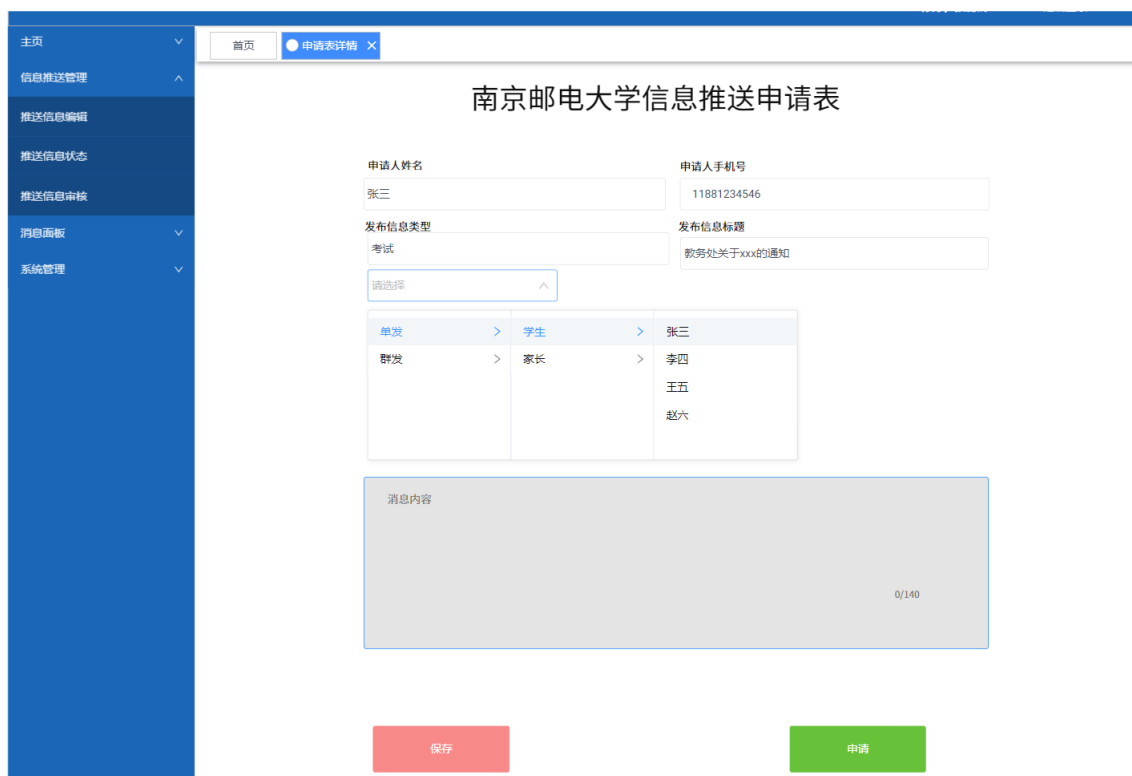


图 7 主界面 UI 设计

三、相关功能模块详细设计

1 模块内的数据结构设计

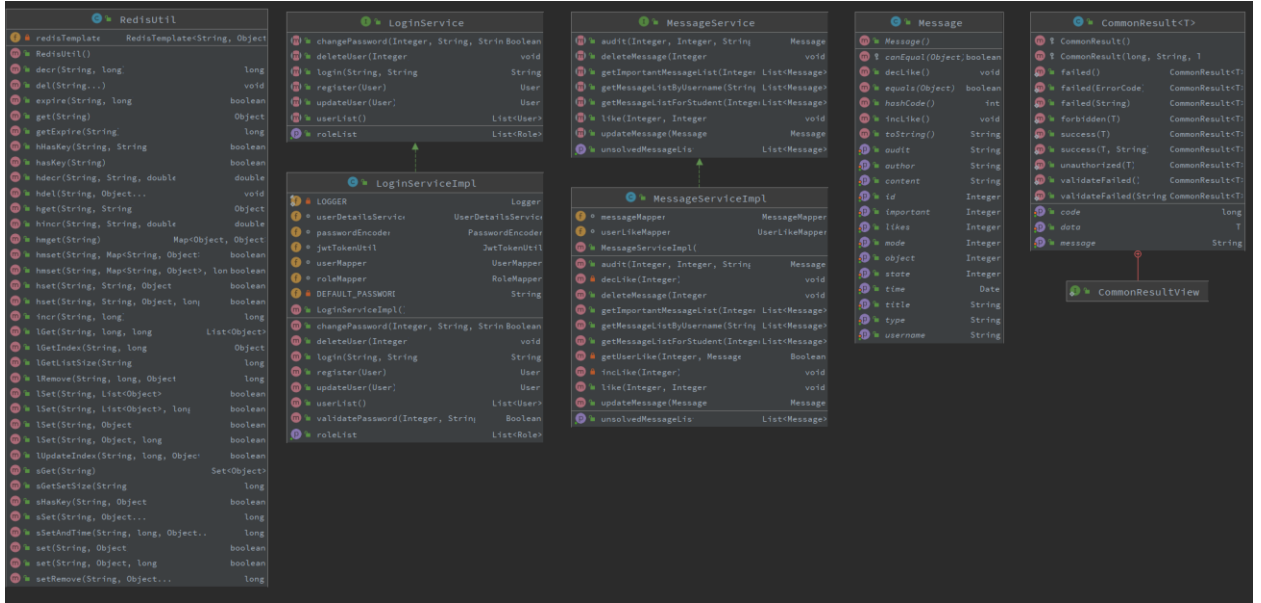


图 8 后端接口服务类图

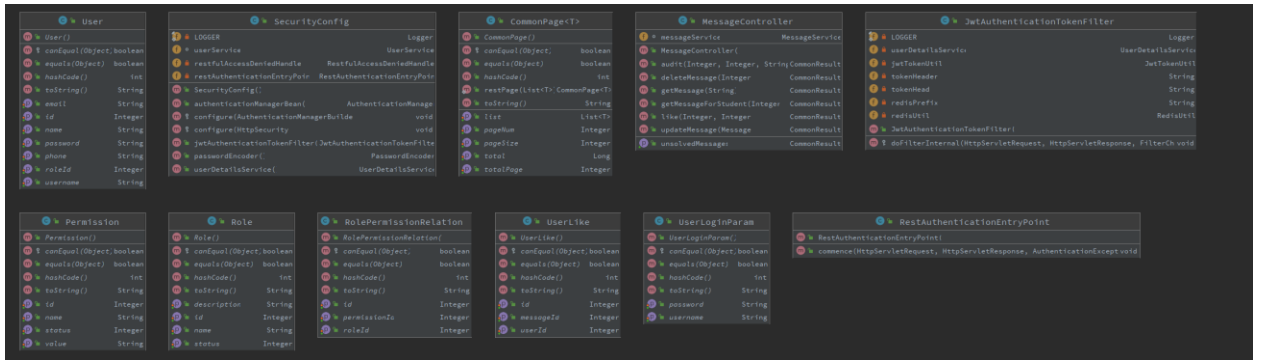


图 9 后端用户信息类图

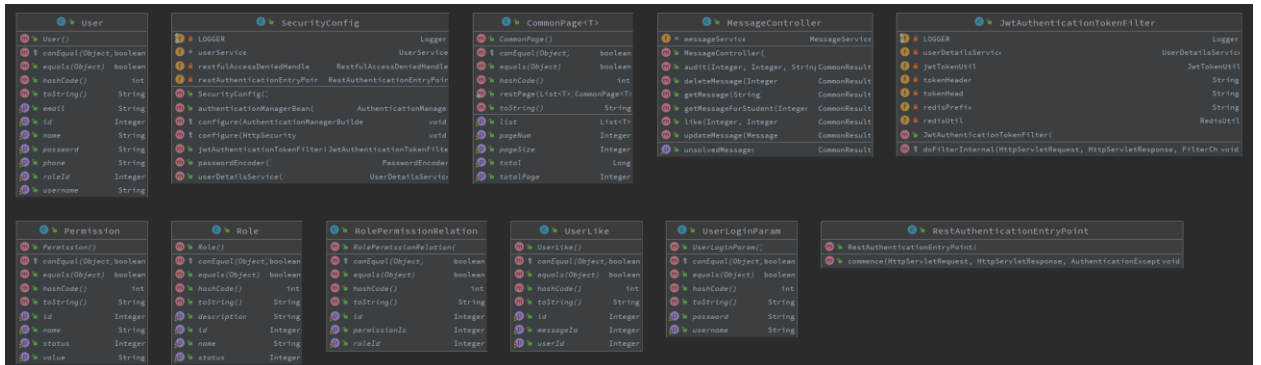


图 10 后端用户行为控制类图

在图 8, 9, 10 中展示了后端模块中的类图。具体可分为三大类, 前后端分离时向前端暴露的接口服务类, 用户信息类, 和用户行为控制类。

其中前端暴露接口服务类主要分为两类, 用户系统模块以及推送系统模块。具体又可细分出 4 个接口类, 即 LoginService (负责对登录的验证), MessageService (负责推送信息的处理和数据库的处理), Errorcode (负责向前端传递出错处理信息), UserService

（负责处理当前的登录用户的权限认证和用户信息的管理）。两模块对前端暴露的接口服务如图 11，12 所示。

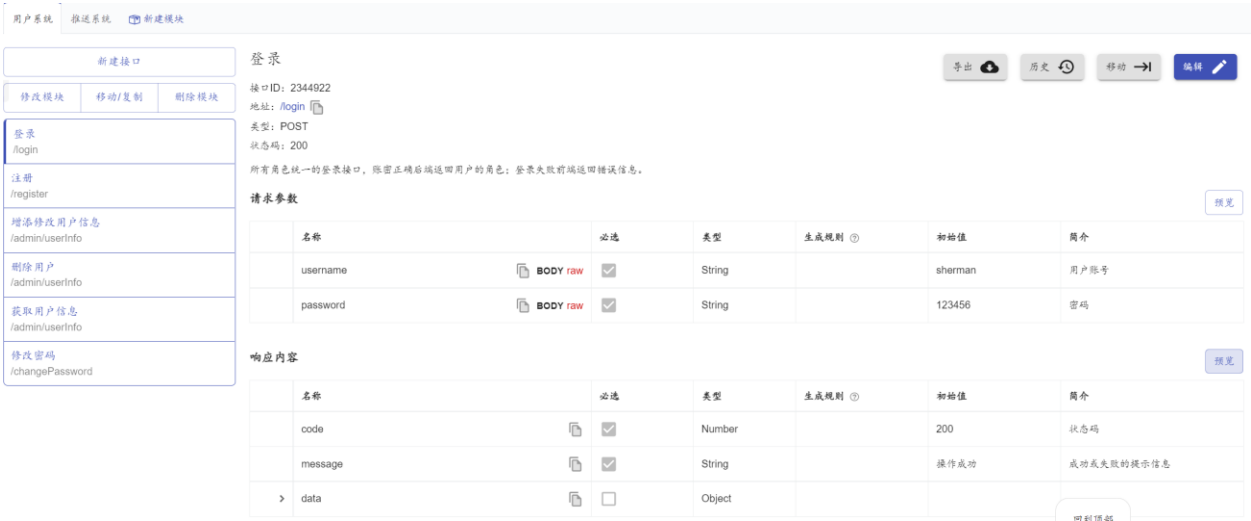


图 11 用户系统模块接口

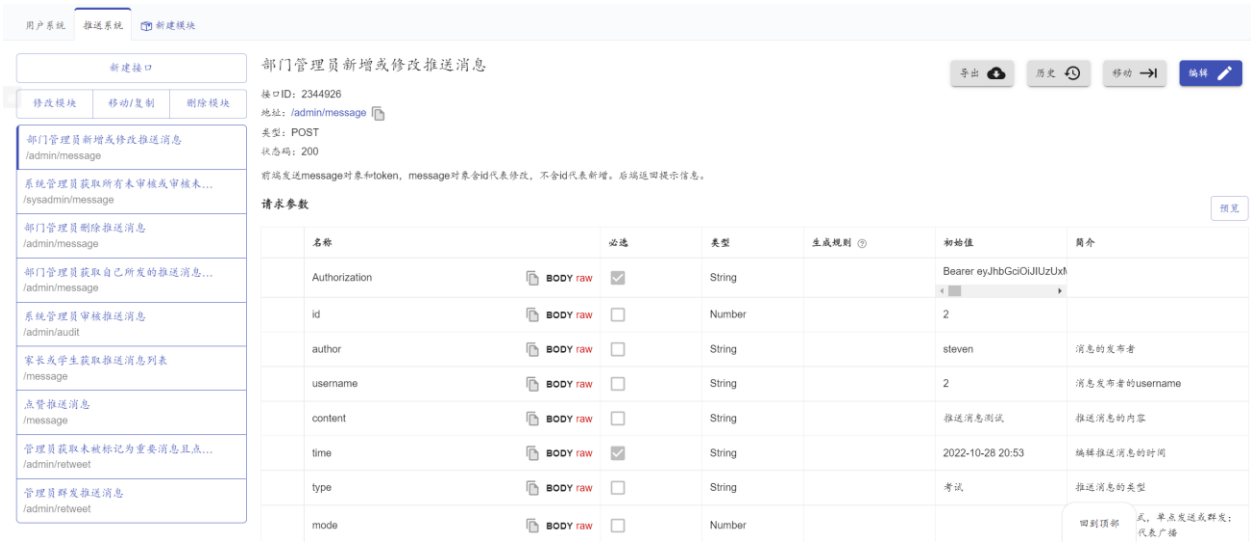


图 12 推送系统模块接口

后端用户信息类则主要涉及到整个系统对用户和信息存储的管理。主要处理的是用户角色信息匹配以及对消息的处理权限分配对应的操作处理。包括对前端传回端口的 token 验证匹配用户，对前端请求的行为验证是否符合权限，对信息的获取和审核以及推送申请的处理权限。这部分由于采取了动态权限绑定的方式，所以不同用户对权限的对应关系都可以相应改变，不同角色拥有的信息处理权限也可以相应改变。

用户信息行为控制类则是针对于用户请求行为的对应函数控制。主要涉及对数据库相应信息（如用户权限，信息内容等）的处理，即对数据库进行相应的增删改查。

在前端模块中，由于采取了 Vuex 状态管理库以及 cookies 缓存技术，故而也有针对于当前登录用户的存储设计。如图 13 所示，浏览器会在 cookies 中存储当前登录用户的

Token 以及相应的用户信息和用户权限以及生成的对应动态路由和页面权限控制信息。

```
const getters = {
  sidebar: state => state.app.sidebar,
  device: state => state.app.device,
  token: state => state.user.token,
  name: state => state.user.name,
  user: state => state.user.user,
  role: state => state.user.role,
  addRouters: state => state.permission.routes,
  permission_routes: state => state.permission.routes,
}
export default getters
```

图 13 浏览器 cookies 存储信息类

对于数据库的数据结构设计，系统共使用了 6 张表。如图 14 所示。其中 message 用于存储推送信息相关的数据，包括推送信息的发布者，信息内容，审核状态，编辑时间，发送方式，发送对象，热度；permission 用于存储当前系统中的用户行为权限，具体分为增添修改用户信息，删除用户，获取用户信息等；role 用于存储系统中的角色，在本系统中即为系统管理员，部门管理员，家长，学生；role_permission_relation 则用于存储系统中角色对应的用户行为权限，例如只有管理员可以进行删除用户和增添修改用户信息，只有系统管理员可以进行信息审核等；user 则是用户表，用以存储系统中的用户相关信息，比如用户名，密码，姓名，电话等；user_like 则单独记录用户对不同信息的点赞记录，可以基于此表得到关于某用户的所有点赞信息，或者每条信息的所有点赞用户。

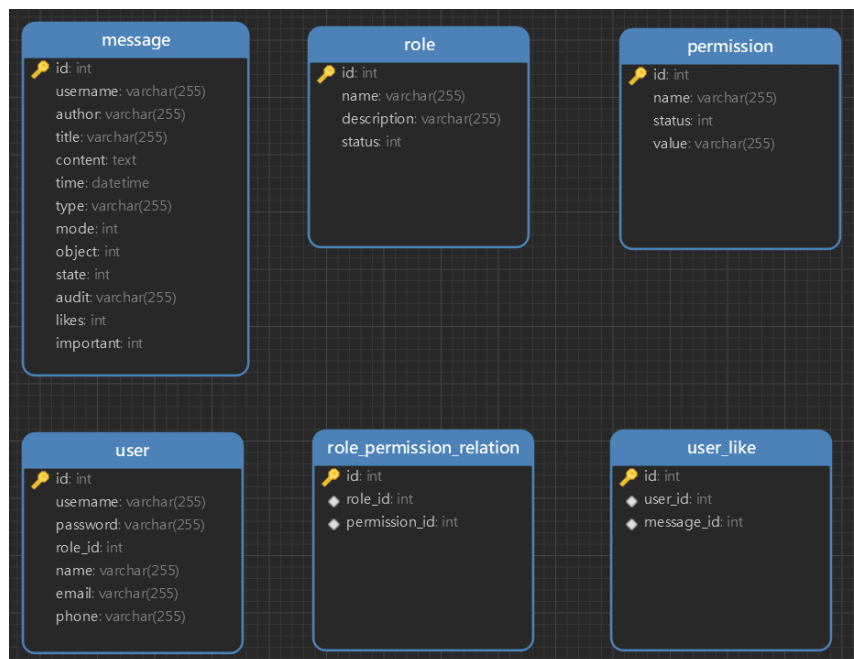


图 14 数据库表设计

从数据库的设计里可以看到将重点放在了用户和角色的分离，角色和权限的分离，以及之间的动态对应关系。这考虑到了现实生活中管理系统中的需要，因为所有的系统需求都在动态变化，用户的角色，角色对应拥有的权限都会随着业务的发展和系统的集成和进一步功能模块的扩张，需要新增角色或者某一个角色需要增加新的权限，这些都可以通过后端对数据库的更改在后台完成，亦可暴露给前端实现用户对于不同角色的切换。关于本系统的权限控制部分将在模块程序设计部分详细介绍，这里不再展开。

2 模块程序流程图设计

2.1 登录模块和 API 请求模块

前后端的信息传递由前文所言，是由 http 协议传递 json 文件数据，故以登录模块为例，介绍前后端的信息传递验证流程，流程图如图 15 所示。

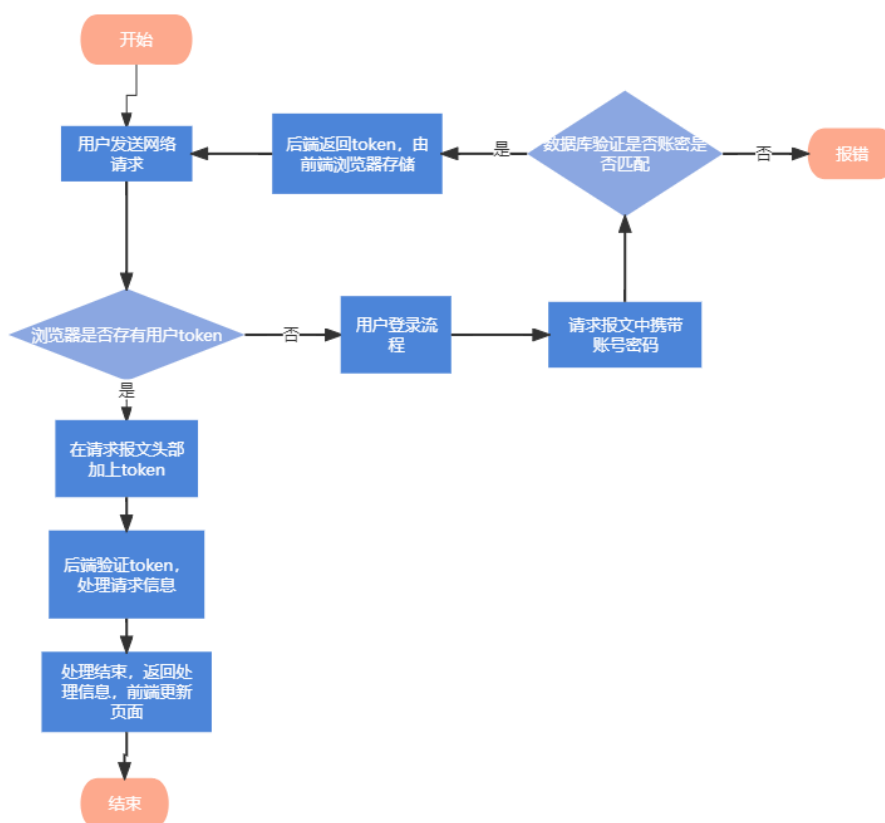
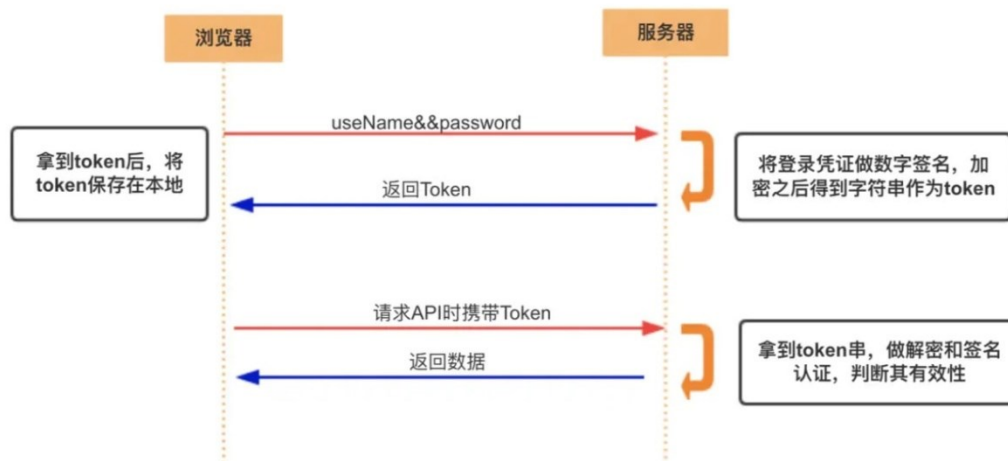


图 15 前后端请求交互流程

在本模块的设计中采取了利用 Token 来实现用户身份的验证。这里的 Token 是服务器产生的令牌，是用户访问资源接口（API）时所需要的资源凭证，包含 uid(用户唯一的身份标识)、time(当前时间的时间戳)、sign（签名，token 的前几位以哈希算法压缩成的一定长度的十六进制字符串）。

在这里后端采用了 Redis 数据库来做存储，设置了过期时间，每次客户端请求的时候取出来验证，除此之外还防止同一用户账号在多处同时登陆，保证了接口访问安全和用户的账户安全。此外，选择了中心化的 Redis-Token 方式，也是考虑到了去中心的无状态 JWT 服务对于 Token 泄露等安全问题的隐患的无能为力。而 Redis-Token 模式可以令本地数据库存储的泄露 Token 直接过期，以保障系统的安全性。对于 Token 实现的用户认证授权的流程看参照图 16 的时序图。



<https://blog.csdn.net/huaguo123>

图 16 token 验证模式

前端对于 Token 的处理则是将 Token 存储在浏览器本地 cookies 里而不是 Web 存储对象 localStorage 中。若将 Token 存储在 localStorage 中则可以通过同域的 JS 脚本访问，这样导致很容易受到 XSS（Cross-site scripting）攻击，在项目中引入很多第三方 JS 库的情况下，如果 JS 脚本被盗用，攻击者就很容易利用这种漏洞在网站上注入恶意的客户端代码，从而伪装成用户突破网站的访问权限。

图 17 的流程图展示了前端对于一般的 web 路由跳转的拦截函数流程。由于浏览器的 cookies 是有过期时间的，所以存储在 cookies 里的 Token 就需要时时刷新。且登录页面和接口仍需开放给所有未登录的访问用户，所以系统对于用户请求的路由跳转就需要时刻验证当前用户的身份以确保系统安全性。前端采取的方式就是通过每次用户请求路由跳转就对请求进行拦截判断当前请求的页面是由登录界面发出还是系统内页面，判断当前用户浏览器的 cookies 是否存储有 Token 来决定用户的页面反馈情况，以保证不会出现用户越权等安全问题。

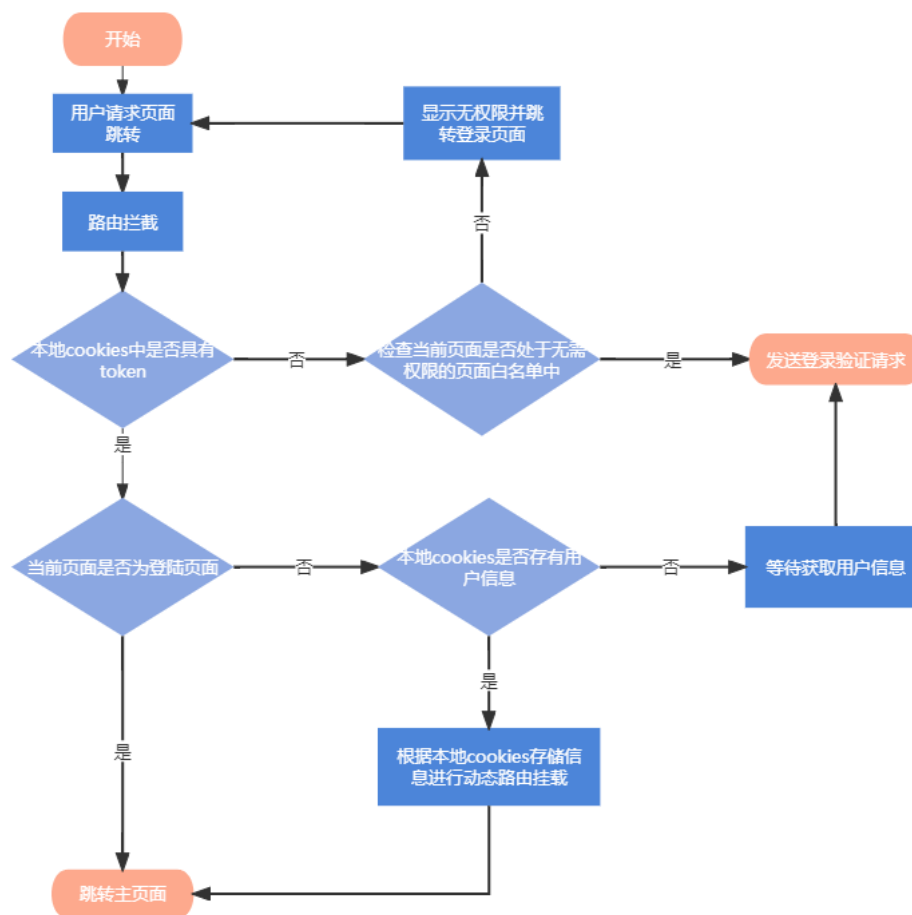


图 17 web 路由跳转拦截流程

2.2 权限控制模块

由于采取了 RBAC（Role-Based Access Control）权限控制方案来处理系统中的权限问题，前后端各自做了依据用户角色进行存取控制的权限处理。这一块也是本项目最为耗时和复杂的部分，但是选择这种设计方式就是为了使本系统贴近可上线的真实项目，更具有实际意义。

前端对于权限的设计是基于系统的路由表的权限设置，为用户展现相应的权限对应的页面。简单概括来讲就是通过获取当前用户的权限去比对路由表，生成当前用户具有的权限可访问的路由表，最终实现系统的动态路由挂载，采用的递归算法流程如图 18 所示。采用这种动态路由挂载的方式的好处在于，由于路由表是可更改的，故而对于未来系统的进一步集成和相应角色权限的页面分配也打定了框架基础。不会像传统系统设计模式为不同的角色设计不同的系统界面形成固定路由表，随着需求的更改和权限变化系统会被反复集成，页面多次进行修改最终使得系统难以维护和繁杂。

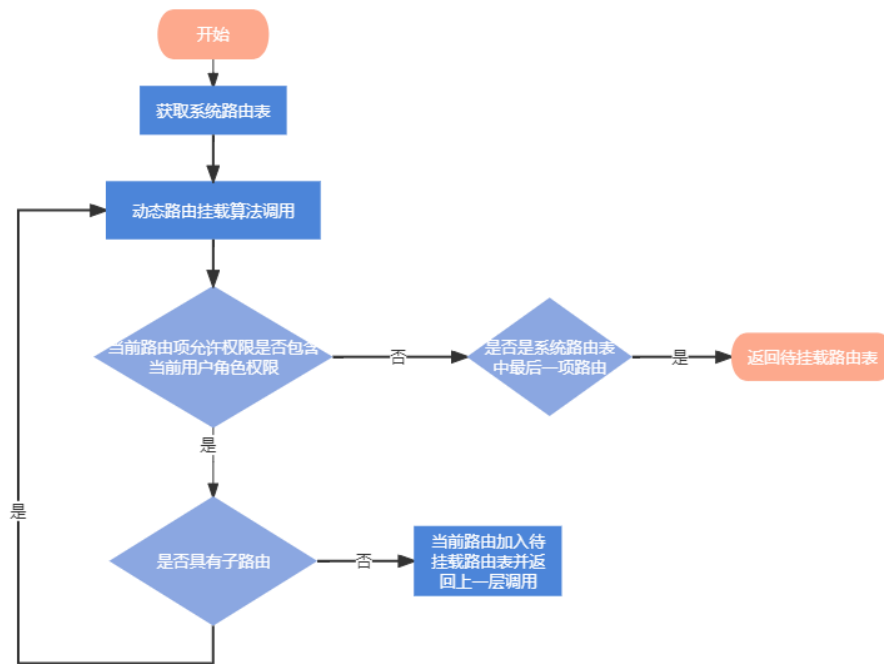


图 18 动态路由挂载的递归算法

后端对于权限的设计主要是基于数据库中的权限-角色表来设计的。由于系统中的角色并不是固定不变，且角色拥有的权限也并不是一成不变的，所以对于角色的动态权限设计就尤为重要。后端采取了 Spring Security 来实现权限控制。由于后端向前端暴露最重要的部分就是资源访问接口，所以对于接口的权限控制访问也异常重要。接口的权限控制逻辑如图 19 所示。

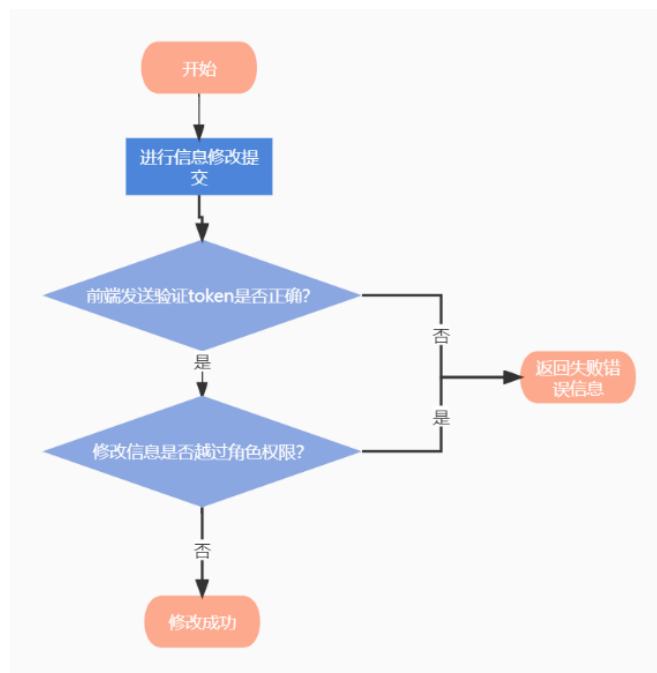


图 19 接口权限控制逻辑

虽然在前端已经针对于角色权限做了控制，可以确保权限不够的角色用户不会接触到相关接口服务，但是为了避免攻击者采取绕过前端页面通过地址路由直接对暴露的端口进行请求或攻击，因此前后端分别针对于角色权限做设计和校验是对于系统的安全性的重要保障。

2.3 系统业务模块

根据前述介绍，本系统实际业务流程为两个模块核心：用户账户管理和信息推送管理。其中用户账户管理为一般的账户管理流程，包括系统和部门管理员对于用户账号的增删改查，对于用户角色的分配，对于本账号附带用户信息的修改以及密码的修改。对于用户的角色和权限对应关系，如图 20 所示。

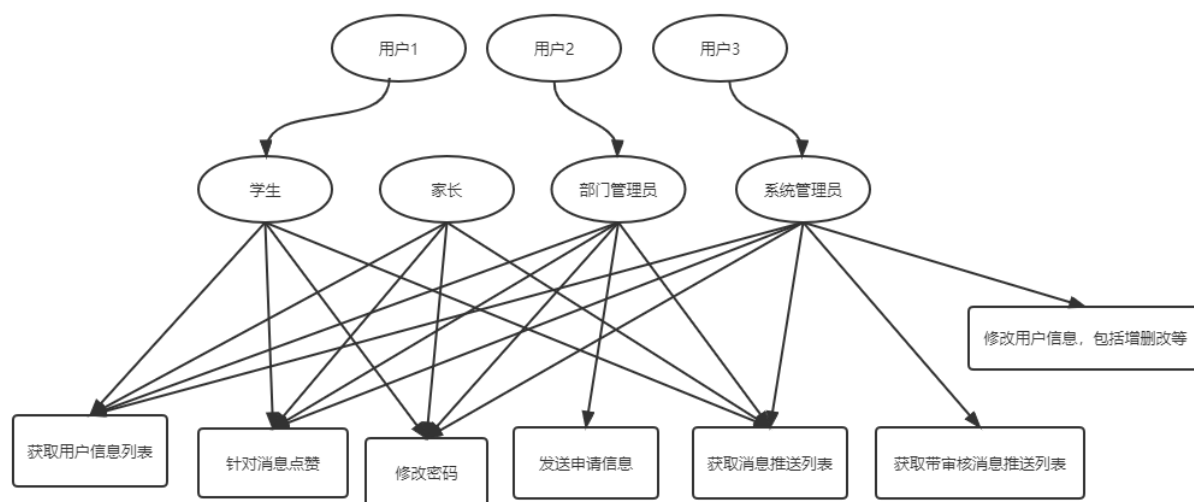


图 20 用户、角色和权限的对应关系

实际上由于本系统一开始采用的 RBAC 权限控制方案，这一对应关系都是可以根据不同的需求可以具体修改。比如为了简化认为学生和家長不能发送学校推送信息，但是如果需要，这一点也可以通过仅仅更改对应关系就可以做到实现，无需对代码进行修改。此外在本系统中角色和用户是一对一的，实际上也可以进行拓展为一个用户对应多个角色，比如如果学校教职工的子女恰巧也是本学校学生，那么就可以给他的管理员角色以外附带增加家长角色。不过出于一般生活实际情况，这样做没有太大意义，所有就没有这样设计。

关于推送信息模块，其工作流程可以见图 21。基本的流程为由教务处、学工处等有推送信息需要的部门管理员在推送消息申请页面填写待推送信息的相关信息，包括消息发送方式，发送对象等，然后该消息将被推送至系统管理员处由相关管理员进行审核。若审核通过则将相应的推送至待发送对象处的消息队列中。

在本系统中一切出于贴近生活实际的角度，为其设置了相应的页面配置，包括批量审核，用户推送的单发和群发，信息推送编辑采取的是富文本编辑，允许用户发布 HTML 页面而不仅仅是文字信息等相应细节。

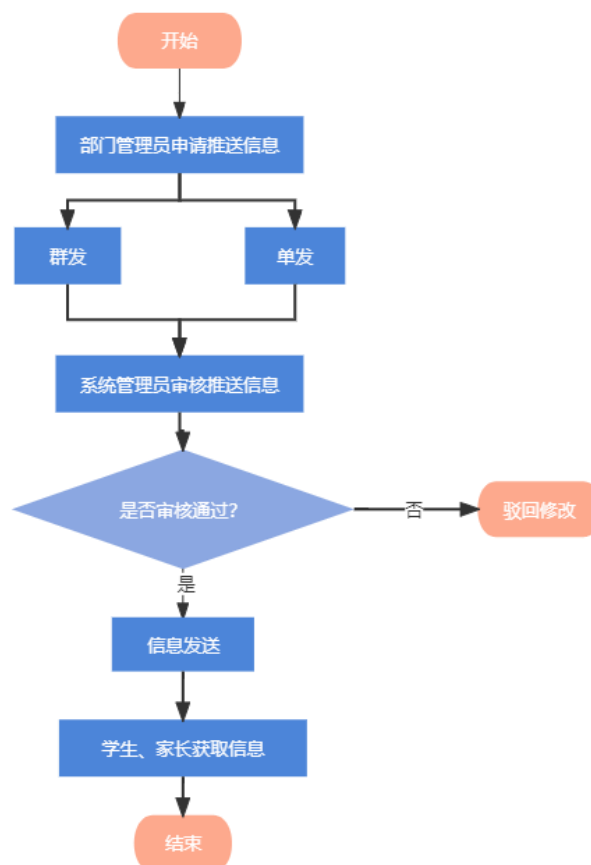


图 21 信息推送模块业务逻辑

除此之外，在系统的功能需求研究初期，还对于系统的其它功能做了思考，虽然出于精力和时间问题没有完成，也在这里一并提出。比如对于热门消息的多种判定维度，出于简化目的，我将热门消息的判定来源于用户对于消息的点赞数量。在后端设置了定时任务，每过一个固定时间段都会将所有消息中点赞数目最多的前十条消息设为重要消息，而后在前端展现。但是现实生活中可以进行分类考虑的是针对消息类型做个性化的热度分析，针对信息的浏览量结合点赞量给出重要消息；在信息的发送方式中除了针对个体的单发就只有针对于全体的群发了，事实上可能还有针对某个学院，某个班级的学生多发的需求；以及消息面板中针对于当前用户的相关度个性推荐，比如学校的新闻可能相关度没有学院发布的相关信息高，那么学院的消息就应该排在学校的信息前，然而这部分做的处理是直接按照消息的发送时间来做排序；此外就是家长针对于和孩子相关的消息推送，家长可能需要学生在校信息的个性化推送，比如自己孩子参加的比赛获奖

新闻也应当可以设置单独为家长置顶。

当然这部分随着实际生活需求的多变，相应的也有更多可以发挥和设计的空间，本项目的可改进空间也异常大。

五、软件测试及其结果分析

本系统经过测试验证可以正常运行所有功能，下面是页面功能展示介绍。



图 22 登陆界面



图 23 消息面板界面



图 24 消息详情界面



图 25 申请推送信息界面



图 26 信息状态界面



图 27 信息审核界面



图 28 账号管理界面

基本界面如上所示，基本流程和功能均可使用，前后端交互正常，针对于用户的不
同界面分布也正常显示。界面流程也如前文所介绍，这里不再赘述。