

学生信息管理系统文档

—Web 应用系统开发

学 号： 201910901114

姓 名： 尚若冰

年 级： 2019 级

学 院： 管理与经济学院

系 别： 管理科学与工程系

专 业： 信息管理与信息系统

指导教师： 汤睿

目录

学生信息管理系统文档.....	1
1 需求分析.....	3
2 概要设计.....	3
2.1 功能模块.....	3
2.2 角色权限划分.....	4
3 系统设计.....	4
3.1 用例图.....	4
3.2 数据库设计.....	6
4 主要功能展示.....	7
4.1 用户验证(authentication).....	7
4.2 权限控制(permission).....	7
4.3 限流(throttle).....	9
4.4 缓存(cache).....	9
4.5 数据校验 validator).....	9
4.6 其他界面.....	10
5 系统目前存在的问题.....	11
5.1 数据库设计问题.....	11
5.2 受时间限制未能实现的功能问题与性能优化问题.....	11
6 心得体会.....	11

1 需求分析

为实现学校对学生信息进行数字化管理,实现学生信息观念的系统化、科学化、规范化和自动化,开发一款学生信息管理系统,其主要任务是利用计算机对学生各种信息进行管理,如查询、修改、增加、删除,另外还考虑到为学生提供选课、成绩统计等功能。技术实现层次上,考虑用数据库服务器存储学生信息,用 Web 服务器提供业务服务,在浏览器界面访问系统门户。

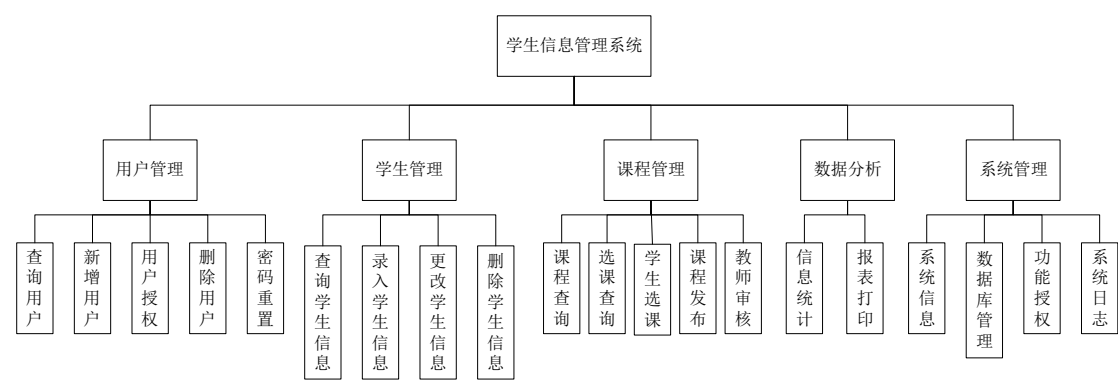
基础功能实现: 利用 Django-Rest-Framework 的 Serializer、APIView、Authentication、Permission、Throttle、Pagination 等模块实现序列化、用户验证、权限控制、访问频率限制、分页等功能,完成学籍管理系统,实现(老师、学生)两种用户的学籍操作功能。

高级功能实现: 将 Django 基于函数的视图(FBV)调整为 Django-Rest-Framework 基于类的视图(CBV), URL 符合 RestFul 规范,利用 Redis 进行 Caching,使用 JSON-Web-Token 进行用户验证、利用 Vue、Vuex、Vue-router 实现前端界面的开发。

2 概要设计

开发环境: Python 3.7.3、MySQL 8.0.26、Redis 5.0.14、Vue 3.2.26、Vue-router 4.0.26、Vuex 4.0.2、Element-plus 1.2.0、echarts 5.2.2
开发工具: Pycharm Professional 2021.3、WebStorm 2021.3、Visual Studio Code 1.6.32、Postman

2.1 功能模块



学生信息管理系统主要分为用户管理、学生管理、课程管理、数据分析、系统管理五大模块,系统内具有管理员、教师、学生三个角色,可依照不同的权限设置,开放模块内不同的功能供对应角色完成业务需求。

2.2 角色权限划分

各角色默认功能权限如下表：

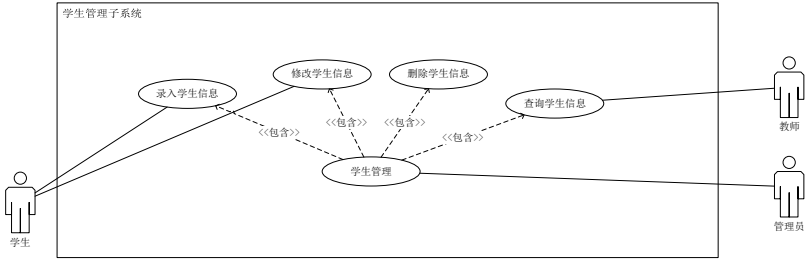
功能模块	具体功能	学生	老师	管理员
用户管理	用户查询			✓
	新增用户	✓	✓	✓
	用户授权			✓
	删除用户			✓
	密码重置	✓	✓	✓
学生管理	信息查询		✓	✓
	信息录入			✓
	信息更改	✓		✓
	删除学生		✓	
	成绩录入		✓	
	信息统计		✓	✓
	报表打印		✓	✓
课程管理	课程发布		✓	
	学生选课	✓		
	选课查询	✓	✓	✓
	教师审核		✓	
系统管理	数据库管理			✓
	系统日志			✓
	功能授权			✓
数据分析	报表打印	✓	✓	✓
	成绩分析	✓	✓	✓
	查看培养方案	✓		
	成绩排名	✓	✓	✓

3 系统设计

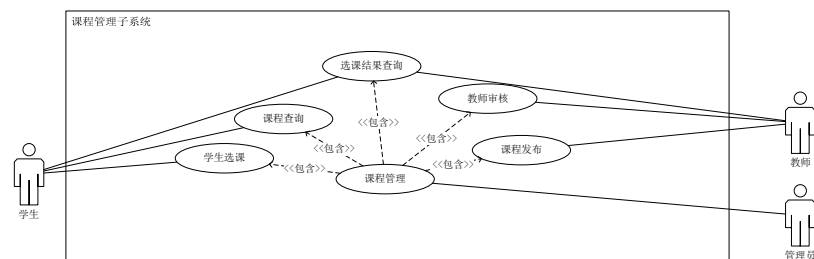
3.1 用例图

各子系统用例图见下：

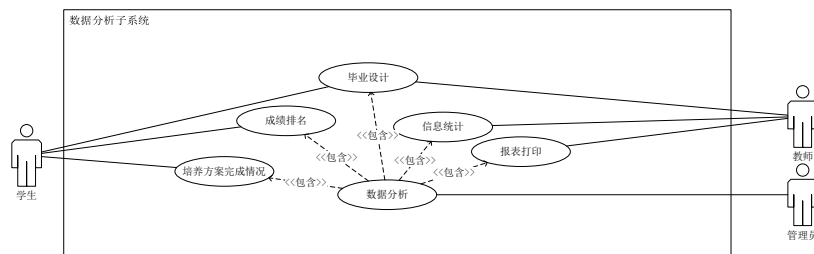
学生管理子系统：学生可以录入自己如手机、邮箱等补充信息，也可以修改自己的信息。老师具有查询学生基础信息的权限，而管理员具有管理学生所有信息的权限。



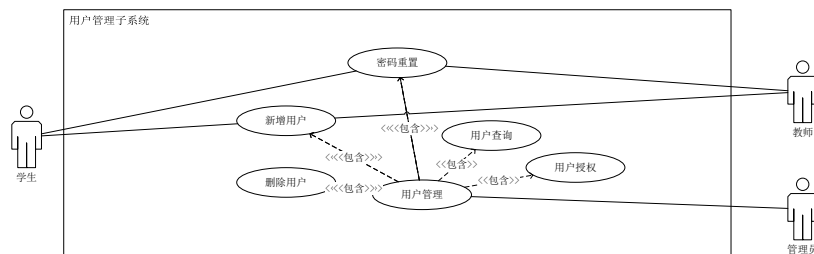
课程管理子系统：学生具有查询课程信息、选课以及查看选课结果的权限，老师可以进行课程的发布、可以通过教师审核功能通过/不通过学生的选课，同时可以查看自己课程的被选情况，管理员具有所有权限。



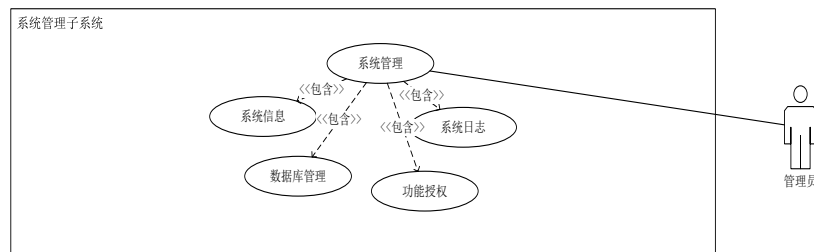
数据分析子系统：学生可以查看自己的培养方案完成情况、可以查看自己在班级中的成绩排名。教师可以通过信息统计查看班级各科的成绩情况，可以通过报表打印学生的成绩单。管理员具有全部数据分析的权限，可以查看所有的信息。



用户管理系统：学生和教师可以通过新增用户来注册新账号，也可以通过密码重置来修改自己的账户密码。管理员具有用户查询、删除用户、授权等高级功能。

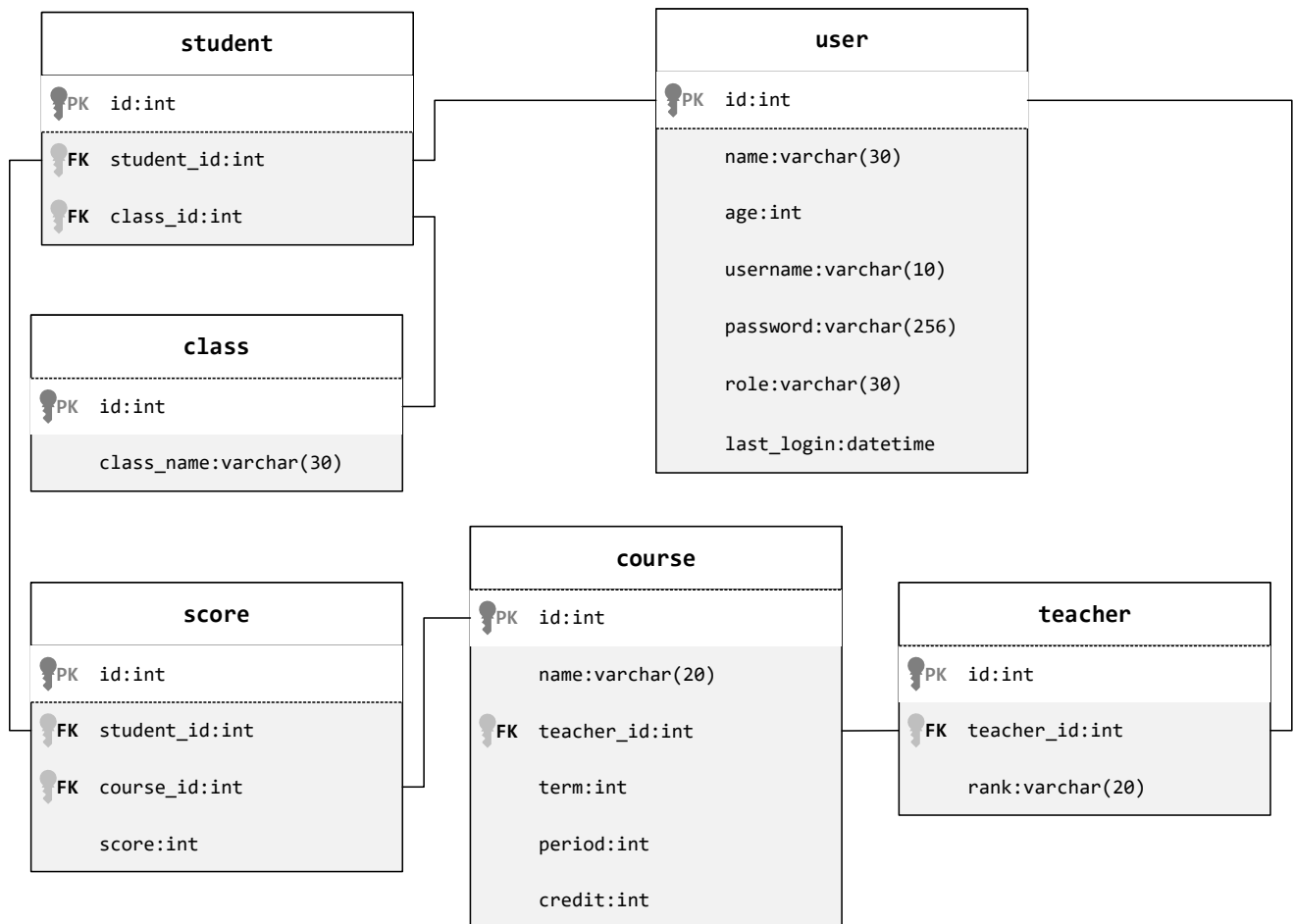


系统管理子系统：本系统只针对管理员开放，管理员可以在该系统实时监控运行状况、可以管理数据库、查看系统日志，通过系统日志锁定操作时间、操作者、操作类型等日志信息，可以进行功能授权、允许其他级别的用户使用某些功能。



3.2 数据库设计

3.2.1 逻辑模型



4 主要功能展示

4.1 用户验证(authentication)

用户验证采用 JSON-Web-Token，过期时间为 1 天，允许用户再次登录刷新 token，刷新后的 token 过期时间也为 1 天，登录成功后 Token 存储在 localStorage。

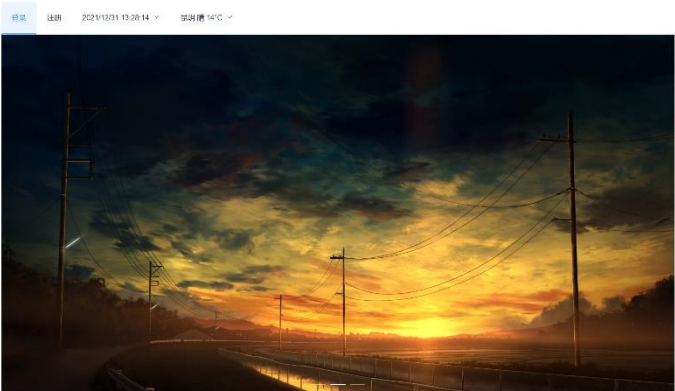


图 1：未登录状态

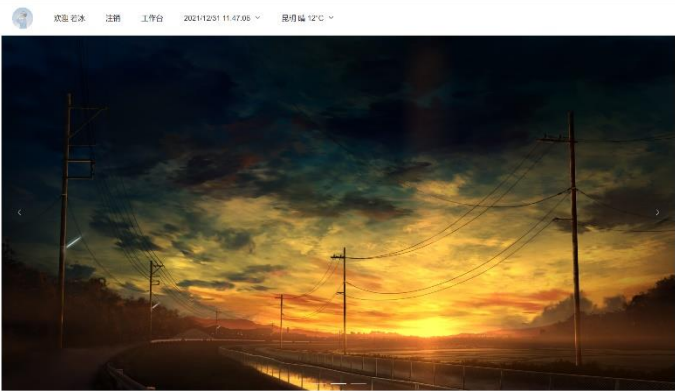


图 2：已登录状态

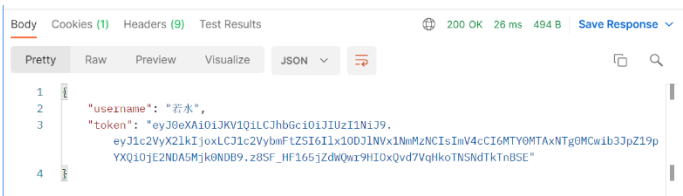


图 3：登陆成功返回 token

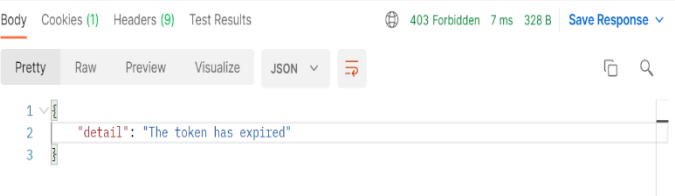


图 4：token 过期

4.2 权限控制(permission)



图 5：管理员可以对用户信息进行编辑和删除

当前用户:亮 切换登录		学生信息管理系统			
用户管理	学生管理	学生查询	成绩查询	删除学生	信息修改
统计功能					
		Student IDClassNameAge			
		1信管191尚若冰20			
		2信管201李亮20			
		4信管211杨文迪20			
		5信管191岳姗21			
		12信管191吴翰芃20			
		<12>			

图 6：学生用户(亮)只能查看学生信息，不可以进行更改

当前用户:若冰 切换登录		学生信息管理系统				
用户管理	学生管理	学生查询	成绩查询	删除学生	信息修改	统计功能
		IDCourseClassNameScore				
		6数据结构与算法信管201李亮90				
		7数据结构与算法信管191吴翰芃99				
		8数据结构与算法信管191岳姗99				
		<12>				

图 7：管理员用户(若冰)可以查看所有成绩

当前用户:亮 切换登录		学生信息管理系统				
用户管理	学生管理	学生查询	成绩查询	删除学生	信息修改	统计功能
		IDCourseClassNameScore				
		6数据结构与算法信管201李亮90				

图 8：学生用户(亮)只能查看自己的成绩

4.3 限流(throttle)

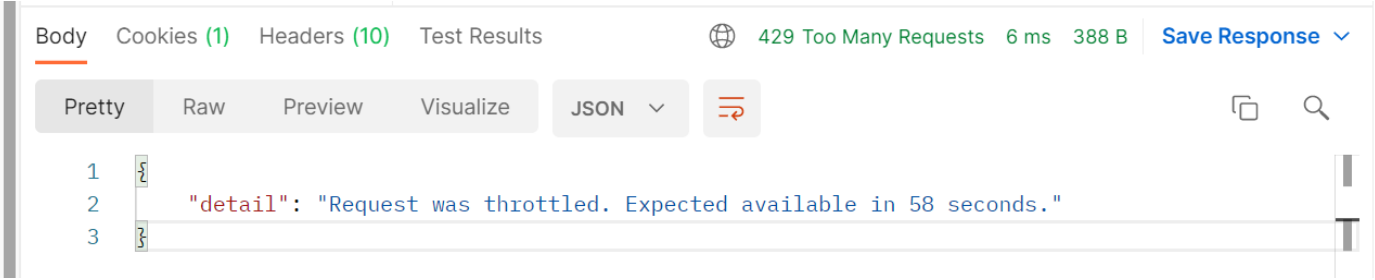


图 7:请求次数过多被限流

4.4 缓存(cache)

利用 Redis 存储缓存，缓存过期时间设置为 2 小时，由下述图 9、图 10 可以看到 Cache 可以提高 Response 的速度

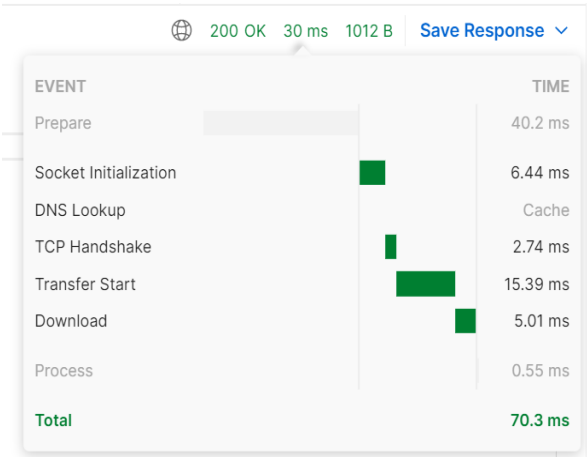


图 9:第一次请求(无 cache)

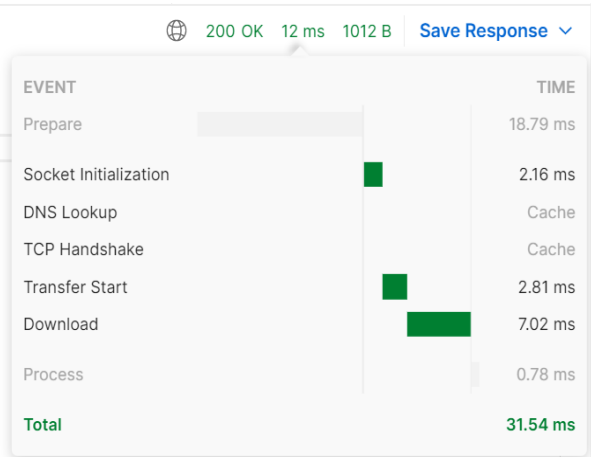


图 10:再次请求(有 cache)

4.5 数据校验(validator)

数据校验由自定义校验类+钩子函数实现



图 11:注册信息校验(密码长度不小于 5)

4.6 其他界面



图 12:当无数据响应时界面

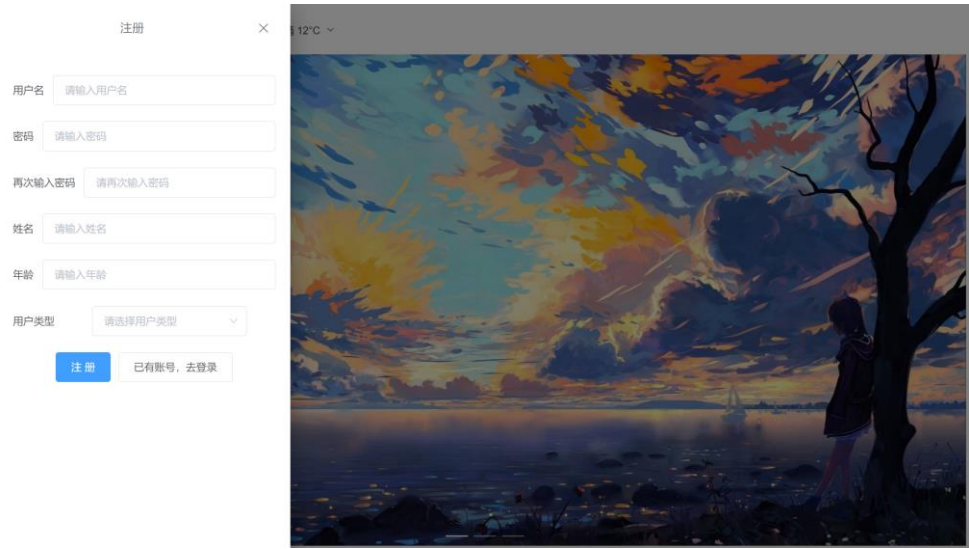


图 13: 注册界面

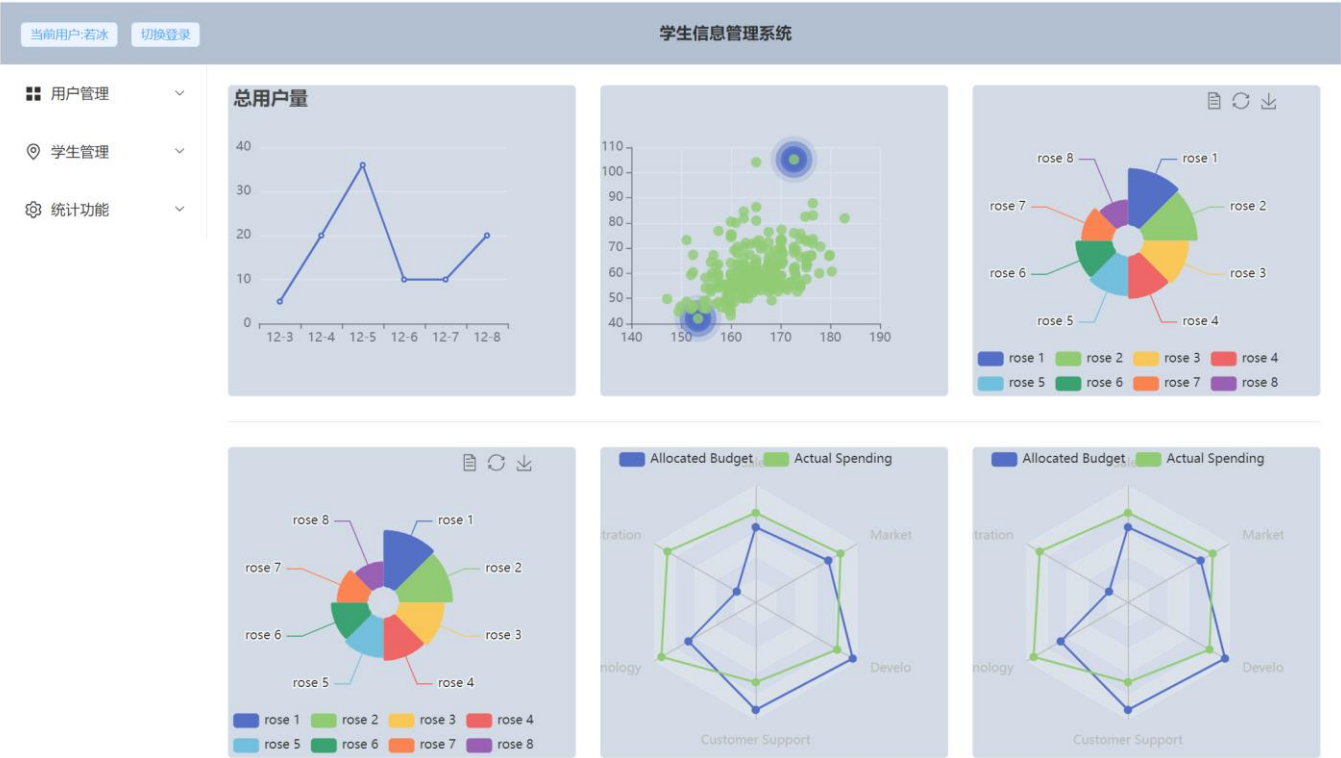


图 14: 数据统计界面(非真实数据)

user_id	name	age	username	password	last_login	role
1	尚若冰	20	若水	827ccb0eea8a706c4c34a16891f84e7b	2021-12-31	1
2	李亮	20	亮	827ccb0eea8a706c4c34a16891f84e7b	2021-12-31	1
3	汤睿	28	Terry	827ccb0eea8a706c4c34a16891f84e7b	2021-12-03	0
4	杨文迪	20	文迪	9559abc957211086589d26975ff1e21c	2021-12-27	1
5	岳姗	21	姗姗	827ccb0eea8a706c4c34a16891f84e7b	2021-12-20	1
6	董毅明	55	小明	66d038ea6d11735a7deb9cc6118b6cd	(Null)	0
8	杨世瀚	49	班主任	827ccb0eea8a706c4c34a16891f84e7b	(Null)	0

图 15: 对数据库中的密码进行 MD5 加密

5 系统目前存在的问题

5.1 数据库设计问题

数据库采用 3NF 设计，消除了数据冗余，但是对于数据库管理员(DBA)而言，数据表的可读性较差，今后设计数据库时注意，增加一定的冗余列可以提高可读性。

5.2 受时间限制未能实现的功能问题与性能优化问题

前端界面语法不够统一，既有使用 Vue3 语法的页面，也有使用 Vue2 语法的页面，今后统一。目前 JavaScript 使用 promise，今后使用 async/await 优化。

6 心得体会

本次实践历时两个月的学习、设计和编码，这段时间内加深了对于 JavaScript 的练习、同时学习了 Node.js、Vue、Vuex、Vue-router、Redis 等技术框架。

这段时间坚持阅读练习了 Django-Rest-Framework、Vue、Vuex 的英文文档，并且阅读了 Django-Rest-Framework 一些源码，这是进步所在。此次实验的不足之处在于，太多的考虑技术实现，较少关注业务，今后改正

我也更加体会到了自主学习的重要性，接下来的一年半时间，我会更加认真的学习专业知识，自学所需的技术知识，不断实践努力向着更高的标准前进。