教务管理系统

（附请假功能）

姓名：杨圣轩

学号：171164239

班级：17级嵌入一班

目录

[1. 简介 3](#_Toc27952865)

[1.1 目的 3](#_Toc27952866)

[1.2 范围 3](#_Toc27952867)

[2. 概述 3](#_Toc27952868)

[3. 系统分析 4](#_Toc27952869)

[3.1 可行性分析 4](#_Toc27952870)

[3.1.1 业务概览 4](#_Toc27952871)

[3.1.2 技术可行性分析 4](#_Toc27952872)

[3.2 需求分析 4](#_Toc27952873)

[3.2.1 编写目的 5](#_Toc27952874)

[3.2.2 功能需求 5](#_Toc27952875)

[4. 系统整体设计 6](#_Toc27952876)

[4.1 需求分析阶段模型 6](#_Toc27952877)

[4.2 基本动态模型 7](#_Toc27952878)

[5. 系统实现 8](#_Toc27952879)

[5.1 系统功能 8](#_Toc27952880)

[5.2 代码模块 15](#_Toc27952881)

[5.3 数据库模块 18](#_Toc27952882)

[6. 系统的环境与部署 18](#_Toc27952883)

[6.1 系统运行环境 18](#_Toc27952884)

[6.2 系统部署方法 18](#_Toc27952885)

[7. 系统的大小与性能 19](#_Toc27952886)

[8. 系统的质量 19](#_Toc27952887)

项目系统设计书

# 简介

## 目的

本系统通过web方式对学生信息等服务进行系统的管理，并在其中添加学生请假及管理请假记录的功能，采用前后端完全分离的方式创建出来的一个企业级项目。

## 范围

本系统的使用范围为各大高校，为的是给各大高校提供一个对学生信息等服务进行系统管理的平台，其中包含学生请假的请假功能，多种功能结合为师生提供更好的管理方式。

# 概述

本系统为简易的高教教务管理系统，包含的功能有：学生个人信息查询、学生个人信息修改、教师根据班级查询学生信息、学生短期请假、学生请假记录查询、教师根据班级查询请假信息、对假条进行管理等功能。系统采用前后端完全分离的编写方式，前端采用Java Script的顶层框架Ant Design Pro，后台采用Java的Spring Boot框架，如此便可以前后端分离操作，对前后端程序人员提供各方面的简化。

# 系统分析

## 可行性分析

此阶段对系统进行可行性分析，通过技术和业务可行性分析系统的设计可行性。

* + 1. 业务概览

假设系统使用者为大学高校，其具有清晰的管理层分化，那么该系统需要提供的便是为了使学生对自身信息的查询及修改、在线进行短期请假并获取请假结果的功能。

* + 1. 技术可行性分析

本系统所采用为web系统设计，通过前端的js框架搭配后台的java框架及底层的sql数据库进行数据存储及分配，最终部署在服务器端进行测试，无误后可推广使用。

## 需求分析

此阶段需要对系统的需求进行分析，确立编写目的及各种功能的需求和设计。

* + 1. 编写目的

本系统为的是将学生信息管理及请假方式变得简单且系统化，避免人工分配产生必要的麻烦和问题，用一套系统的规划让学生与教师得到更舒适的服务。

* + 1. 功能需求

系统客户端需实现的功能：

（1）学生及教师登录功能：设计登陆界面，提供用户的登录功能，并根据身份切换业务权限。

（2）学生信息查看功能：查看个人信息，根据登录人来查看学生的各种信息。

（3）学生信息管理功能：更改个人信息，根据登录人来修改学生的各种信息。

（4）短期请假功能：根据登陆人信息获取默认信息后进行请假，请假成功后传送到管理员处。

（5）查看请假记录功能：学生根据个人信息查看并修改个人请假记录。

（6）查看学生信息功能：教师登录后可以根据提供的班级信息查询该班级的学生，后可以查看班级某学生信息。

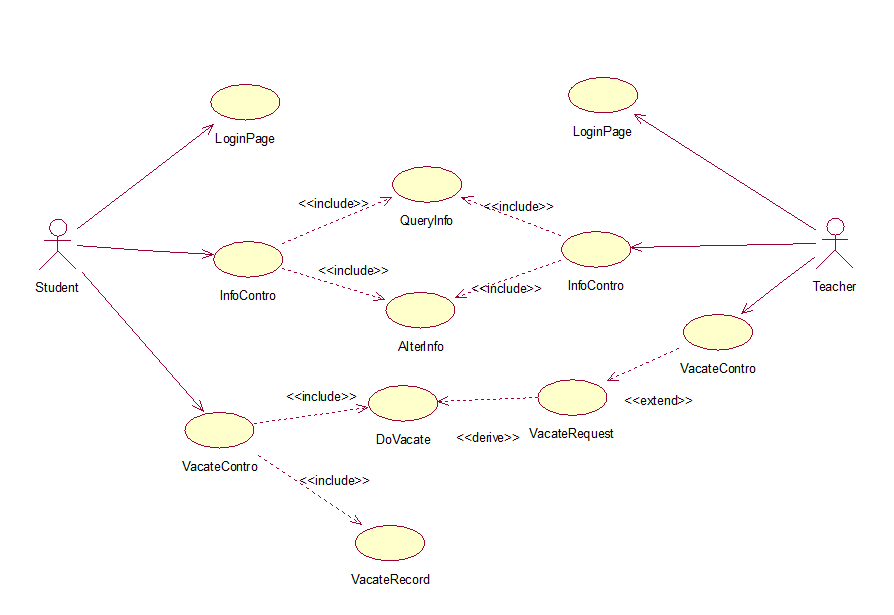
（7）处理请假功能：教师登录后可以根据提供的班级信息查询该班级的请假信息，后可以处理某条请假信息。

# 系统整体设计

对系统进行整体设计，确立基本用例模型及动态模型。

## 需求分析阶段模型

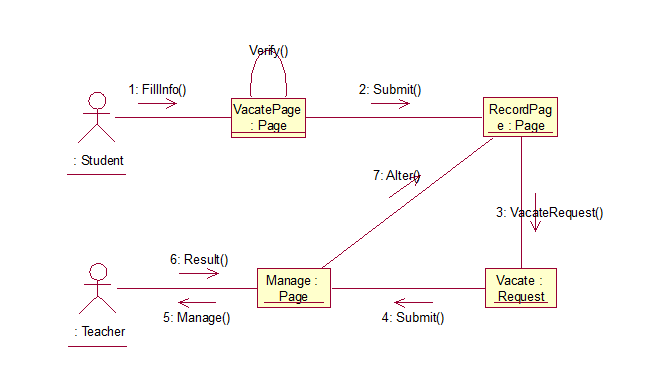
经过需求分析阶段，得出“教务管理管理系统”的整体用例图。图中存在两种主要业务参与者，分别为教师与学生，这些业务参与者对应用例的详细描述如下。



1. Student：参与者，身份类型为user，通过登陆界面进行账号密码确认进行登录，拥有用户的权限，可以查看并修改个人信息，提交短期请假请求及查看请假记录以及修改未审核或者已驳回的请假请求。
2. Teacher：参与者，身份类型为admin，通过登陆界面进行账号密码确认进行登录，拥有管理员的权限，可以查看并修改个人信息与学生信息，处理短期请假请求。

## 基本动态模型

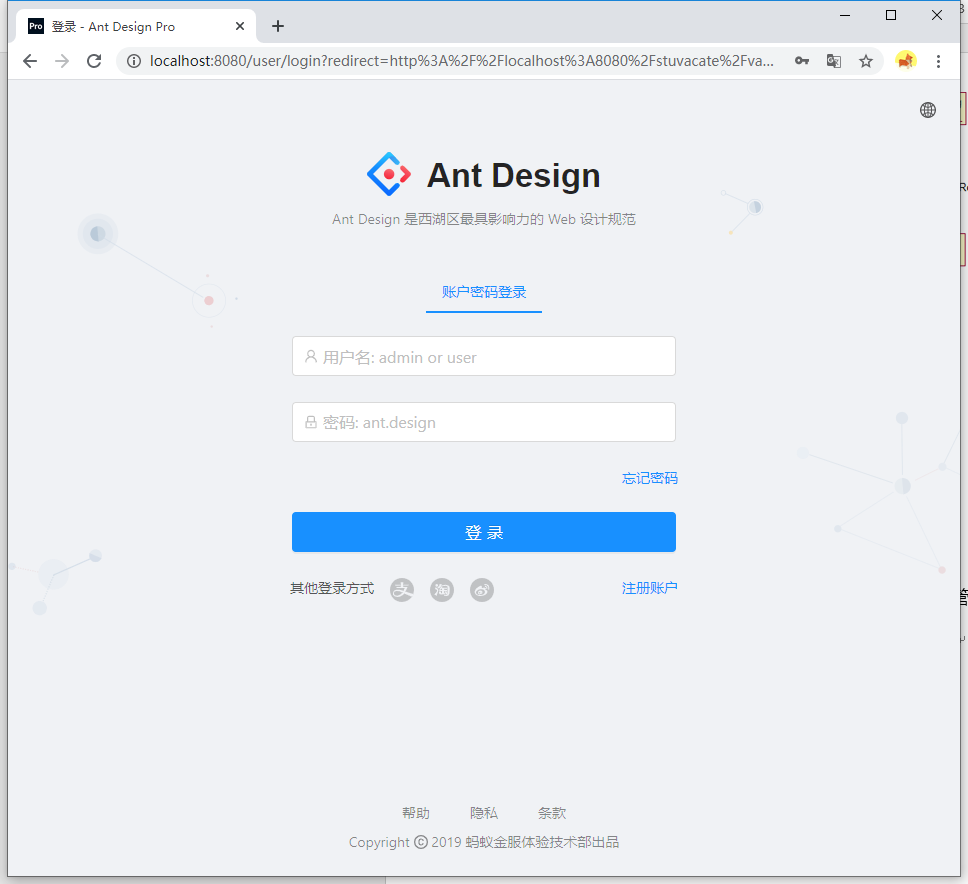
经过详细设计阶段（在此通过代码实现代替），得出学生查询及修改信息、学生请假及修改请假记录、教师查询并修改学生信息、教师查询并处理请假请求等四大类模块以及上述功能需求中所描述的各种功能，在此只展示学生请假模块的通信图。



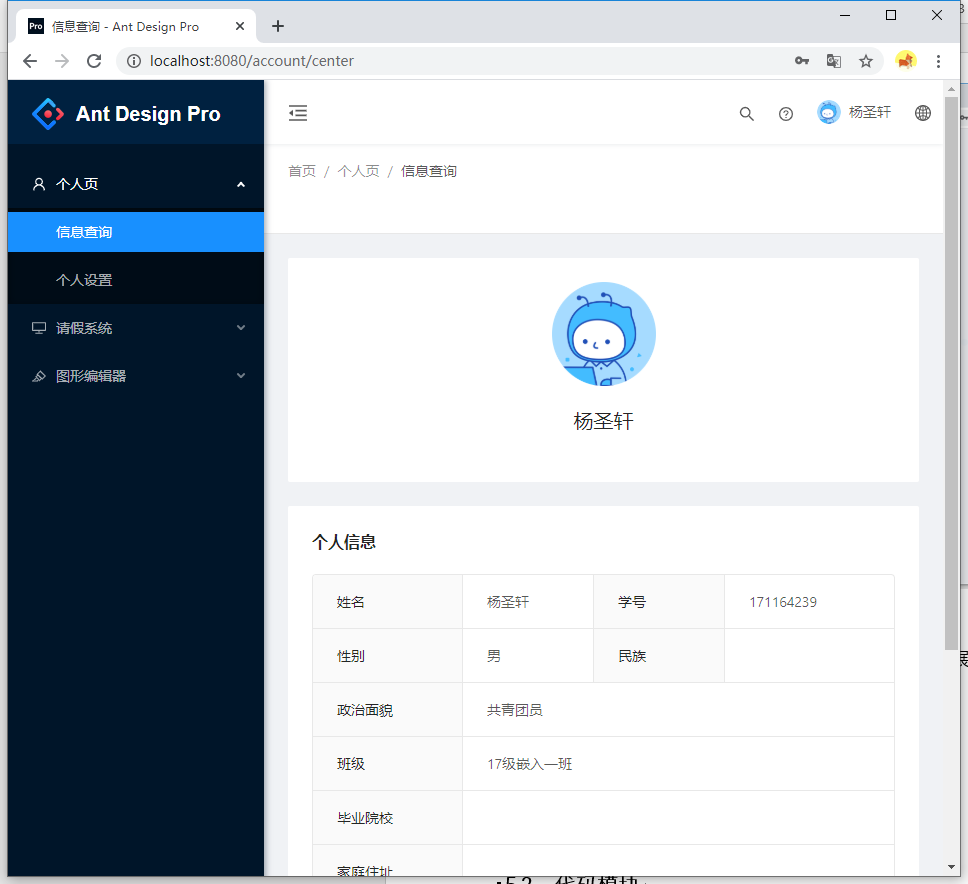
# 系统实现

## 系统功能

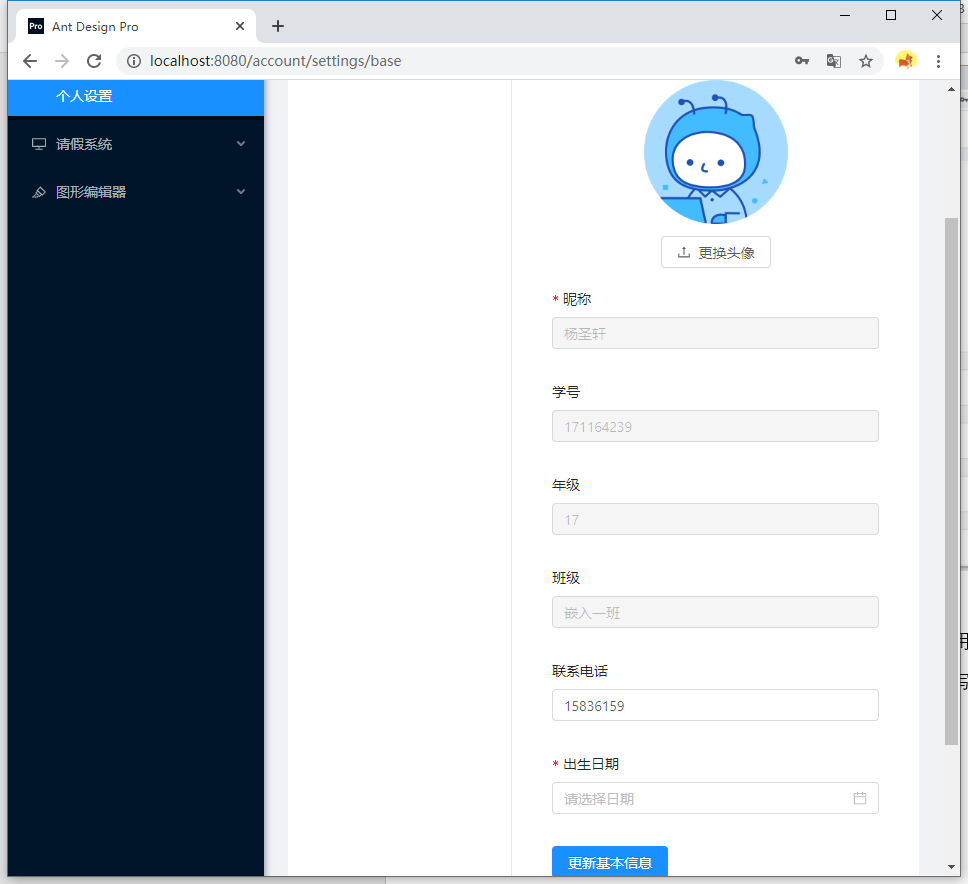
1. 登录功能：用户登录界面，提供学生和教师的登录，管理员信息由后台数据库处理，根据权限显示登陆后不同界面。



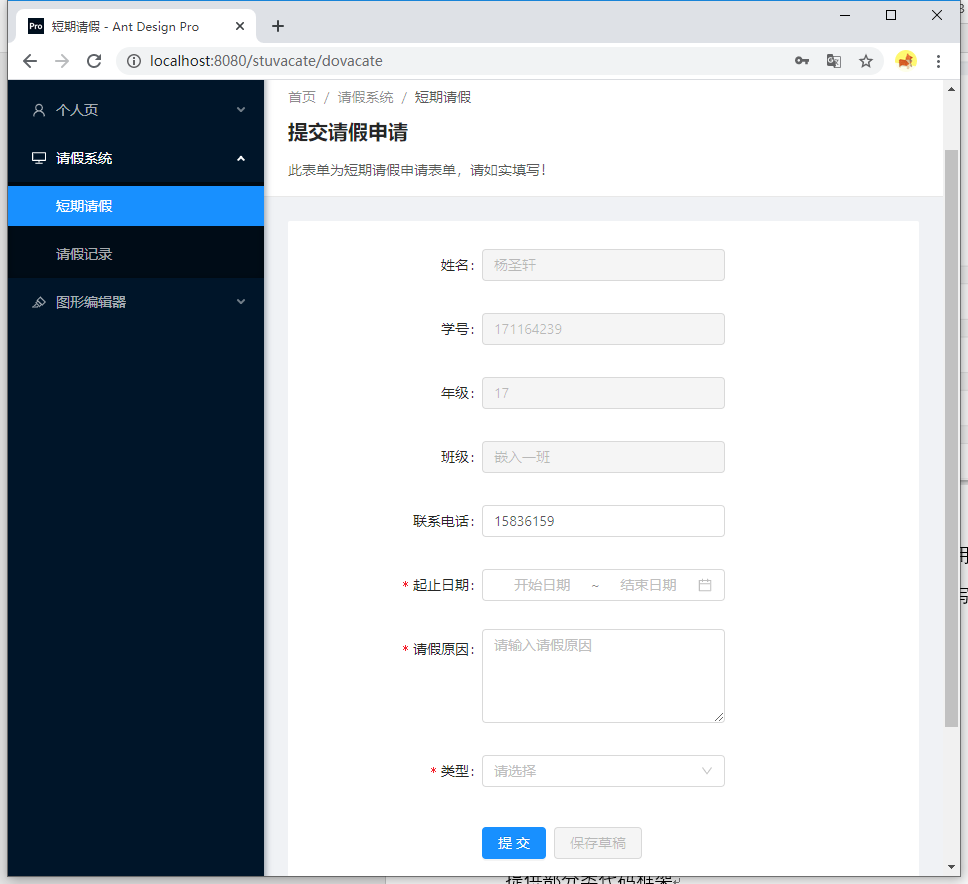
1. 信息查询功能：在用户登陆后，首页会根据登录用户展示其个人信息（学生与教师通用）。



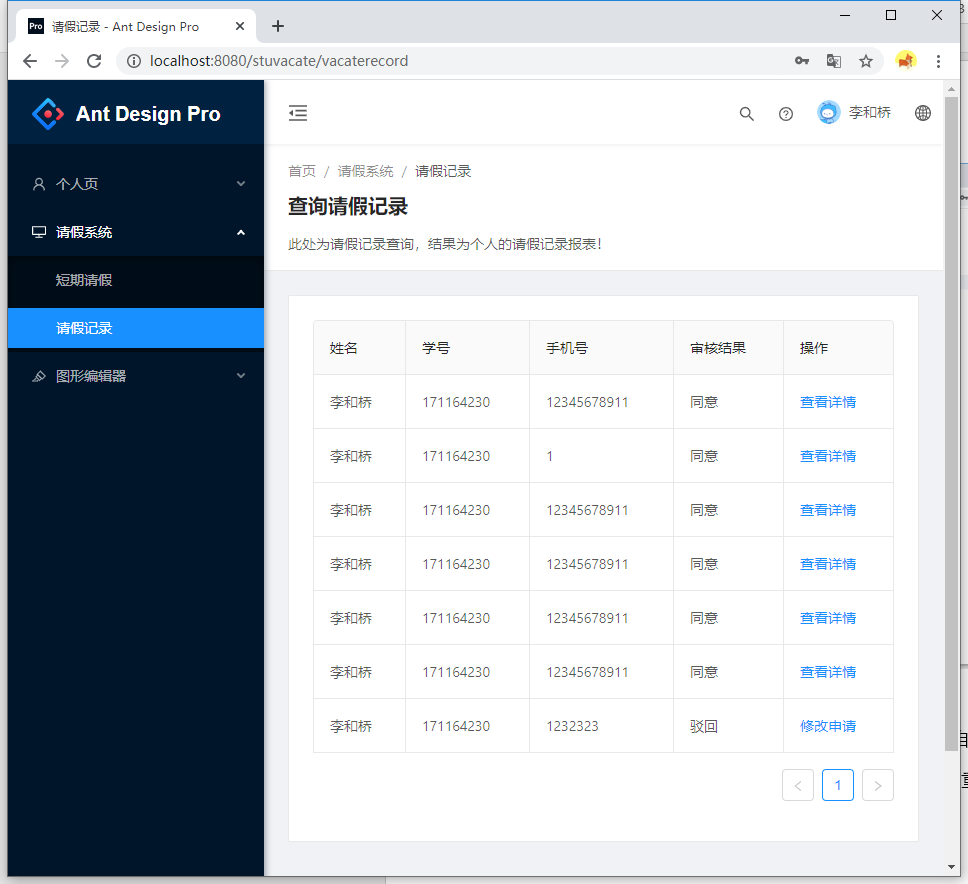
1. 信息修改功能：用户登录后可以根据自己所需，修改除了默认信息之外的信息，默认信息需要管理员于后台修改（学生与教师通用）。

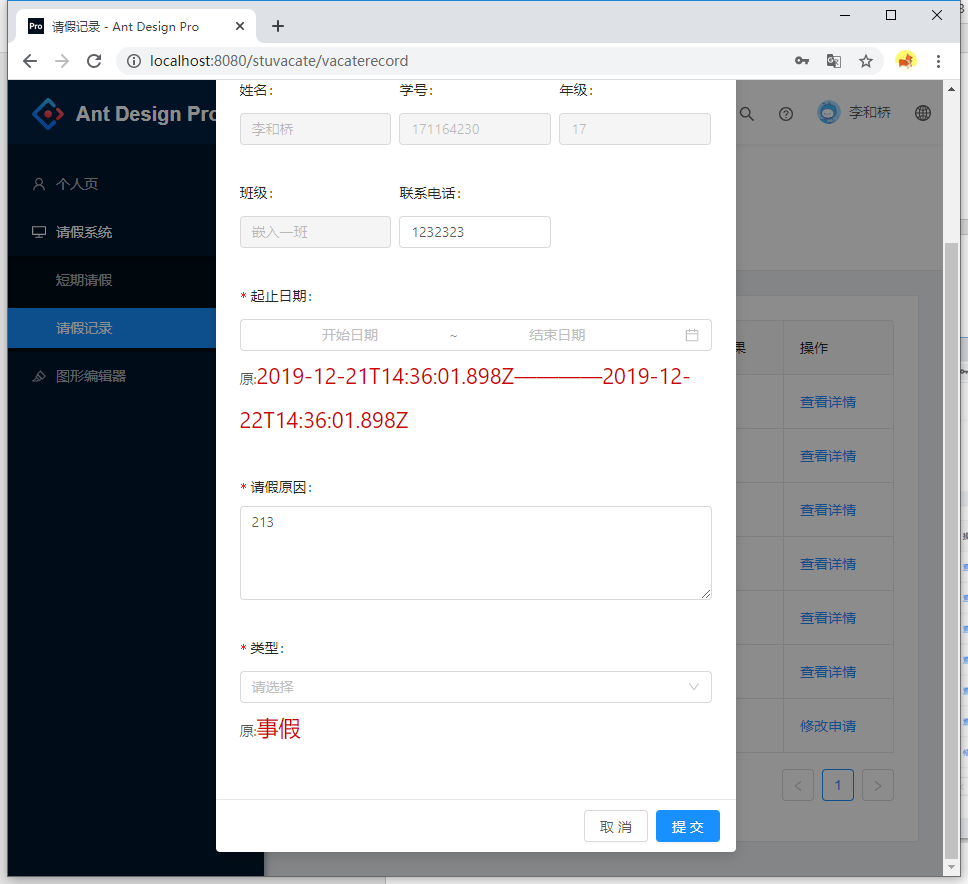


1. 学生请假功能：学生登录后可以在请假系统分类中使用短期请假功能，系统会默认提供学生个人信息，学生只需要填写相关请假信息即可请假，请假成功会获得反馈。

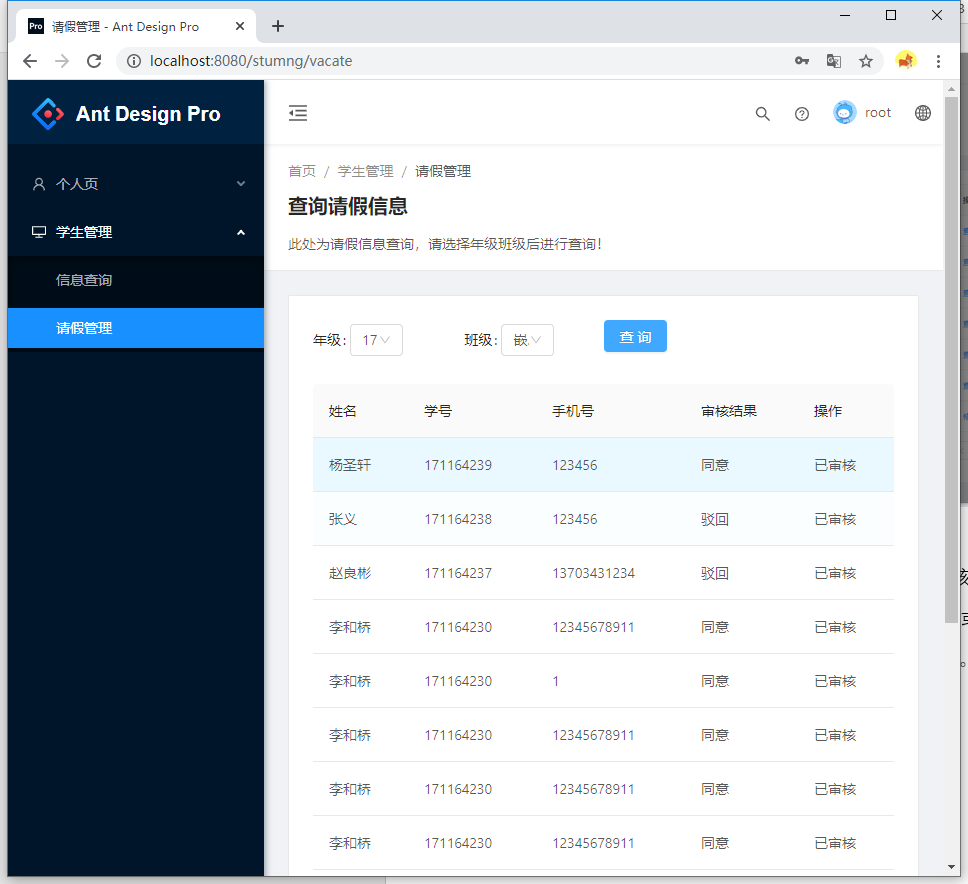


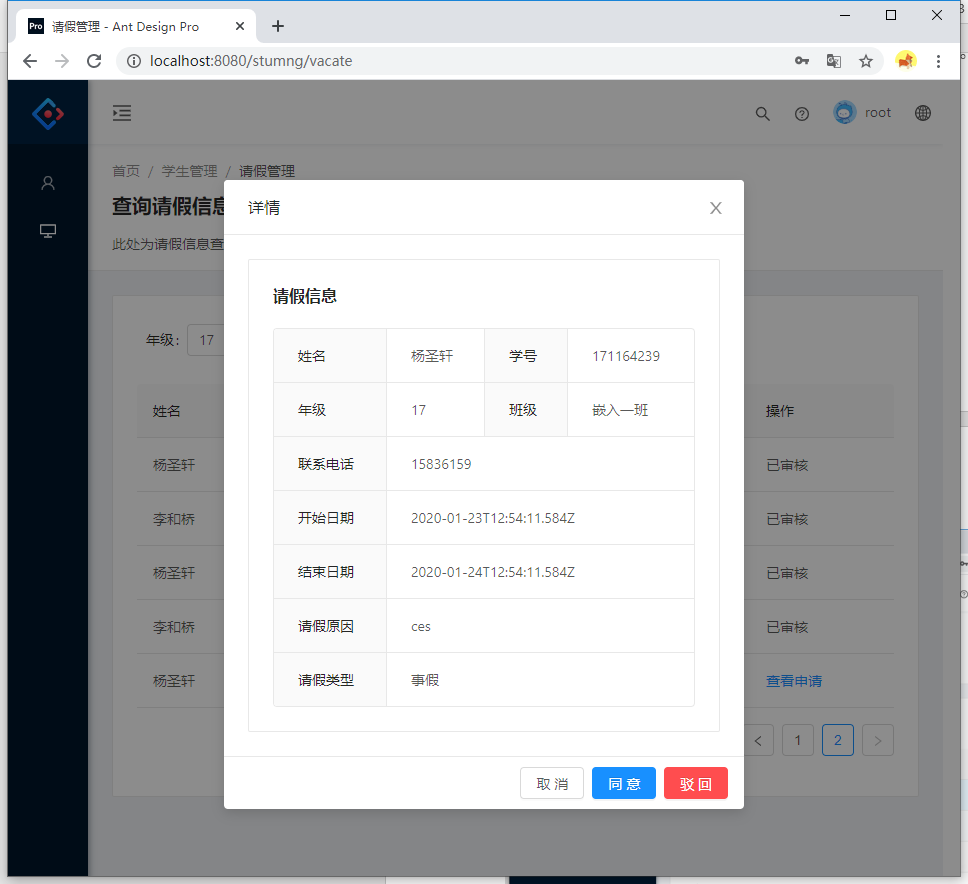
1. 查看记录功能：学生登陆后，可以根据自身信息查看自己曾经的请假记录，已通过不可修改，未通过和未审核可修改重新提交。





1. 处理请假功能：教师登录后，可以根据所填班级查询该班级的请假信息，之后教师可以对每条请假信息进行处理（同意或驳回），之后学生会在自己的界面接收到自己的请假处理结果。





## 代码模块

此处不提供前端代码（有需求可在项目代码中查询），提供部分核心后台Controller代码。

@RestController

@RequestMapping("api")

public class VacateController {

@Autowired

DataSource dataSource;

@Autowired

VacateDao vacateDao;

@RequestMapping("/vacate/insertrecord")

public Map<String,Object> insertrecord(@RequestBody VacateBean vacateBean) throws SQLException {

vacateDao.vacateAdd(vacateBean);

Map<String, Object> tag = new HashMap<>();

tag.put("result","success");

return tag;

}

@RequestMapping("/vacate/getvacatelist")

public Map<String,Object> getvacatelist(@RequestBody JsonBean jsonBean) throws SQLException{

Map<String, Object> tag = vacateDao.getVacateList(jsonBean.getUserid(),jsonBean.getName());

return tag;

}

@RequestMapping("/vacate/updatevacate")

public void updatevacate(@RequestBody VacateBean vacateBean) throws SQLException{

vacateDao.updateVacate(vacateBean);

}

@RequestMapping("/vacate/changeagree")

public void changeagree(@RequestBody VacateBean vacateBean) throws SQLException{

vacateDao.change\_agree(vacateBean);

}

@RequestMapping("/vacate/changerefuse")

public void changerefuse(@RequestBody VacateBean vacateBean) throws SQLException{

vacateDao.change\_refuse(vacateBean);

}

}

……

## 数据库模块

本系统采用Sql数据库为底层，版本为MySql 8.0。数据库代码不在此展示，有需求可直接查看sql文件。

# 系统的环境与部署

## 系统运行环境

IDE：IntelliJ IDEA

数据库：MySql 8.0

另：需要电脑安装git、node.js。

## 系统部署方法

本项目采用了后台Spring Boot的高集合框架，不需要自主配置软件服务器。并且前端框架打包后放在静态文件夹中，使用者使用时只需要用编译器打开项目，让其自动配置maven依赖，然后运行，访问端口号+/user/login（作者采用<http://localhost:8080>为端口地址）即可登录并实现功能。

# 系统的大小与性能

在此描述的是在交付阶段系统具有限定作用的系统容量特征和相响应特征，确定的信息为：

1. 教师和学生信息的存储容量占用极少，服务器足够大的情况下不需要定时清理服务器数据，请假记录可保存一年以上。
2. 可执行文件所占用空间容量只有13MB，其中包括了前端静态文件，前端静态文件打包后占用内存极小，不会影响整体内存占比，交付之前冗余代码已清理整合完毕。
3. 系统关键功能都已实现，唯独在权限方面还留存一小部分bug，作者需要继续学习前端框架，慢慢整改。

# 系统的质量

在此描述的是整个系统的质量维度，确定的信息为：

1. 系统各功能的响应时间：响应时间符合普通用户的反应时间，可以满足用户的使用需求。
2. 系统各部分代码的可复用性：前端框架复用性极强，十分适合各种类型的管理系统，后台框架已清理过冗余代码，在复用性上也表现可观。
3. 系统的可扩展性：扩展性能极强，只需要在后台添加接口并在前端框架中调用接口即可。