**软件说明文档**

**软件名称:寝室报修管理系统**

**姓 名**  **蒋绪**

**单 位**  **湖北文理学院**

**指导老师**  **宁彬**

目录

**[1. 系统功能概述](#_Toc21668_WPSOffice_Level1)** **[1](#_Toc21668_WPSOffice_Level1)**

[1.1学生详细功能：](#_Toc3500_WPSOffice_Level2) [2](#_Toc3500_WPSOffice_Level2)

[1.2宿管详细功能](#_Toc13521_WPSOffice_Level2) [3](#_Toc13521_WPSOffice_Level2)

[1.3维修人员详细功能](#_Toc16694_WPSOffice_Level2) [3](#_Toc16694_WPSOffice_Level2)

[1.4管理员详细功能](#_Toc6201_WPSOffice_Level2) [4](#_Toc6201_WPSOffice_Level2)

[图1-1系统整体详细功能图](#_Toc22922_WPSOffice_Level2) [5](#_Toc22922_WPSOffice_Level2)

**[2. 用户功能操作说明](#_Toc3500_WPSOffice_Level1)** **[6](#_Toc3500_WPSOffice_Level1)**

[2.1学生操作说明](#_Toc20920_WPSOffice_Level2) [6](#_Toc20920_WPSOffice_Level2)

[2.1.1学生登录2-1-1图](#_Toc18234_WPSOffice_Level2) [6](#_Toc18234_WPSOffice_Level2)

[2.1.1学生注册](#_Toc21827_WPSOffice_Level2) [6](#_Toc21827_WPSOffice_Level2)

[2.1.2学生注册2-1-2图](#_Toc1382_WPSOffice_Level2) [7](#_Toc1382_WPSOffice_Level2)

2.1.2[学生主页2-1-3图](#_Toc13428_WPSOffice_Level2) [7](#_Toc13428_WPSOffice_Level2)

[2.1.3学生修改界面2-1-4图](#_Toc15958_WPSOffice_Level2) [8](#_Toc15958_WPSOffice_Level2)

2.1.3[学生修改成功2-1-5图](#_Toc6972_WPSOffice_Level2) [8](#_Toc6972_WPSOffice_Level2)

2.1.4[资源分享2-1-6图](#_Toc5592_WPSOffice_Level2) [9](#_Toc5592_WPSOffice_Level2)

[2.1.4寝室报修2-1-7图](#_Toc12108_WPSOffice_Level2) [9](#_Toc12108_WPSOffice_Level2)

2.1.5[寝室报修成功2-1-8图](#_Toc20360_WPSOffice_Level2) [10](#_Toc20360_WPSOffice_Level2)

2.1.5[公告信息2-1-9图](#_Toc20407_WPSOffice_Level2) [10](#_Toc20407_WPSOffice_Level2)

2.1.6[维修评价2-1-10图](#_Toc30837_WPSOffice_Level2) [11](#_Toc30837_WPSOffice_Level2)

2.1.6[维修评价成功2-1-11图](#_Toc24206_WPSOffice_Level2) [11](#_Toc24206_WPSOffice_Level2)

2.1.7[学生投诉2-1-12图](#_Toc14007_WPSOffice_Level2) [12](#_Toc14007_WPSOffice_Level2)

2.1.7[学生投诉可选择信息2-1-13图](#_Toc19600_WPSOffice_Level2) [12](#_Toc19600_WPSOffice_Level2)

[2.1.8回馈消息2-1-14图](#_Toc14069_WPSOffice_Level2) [13](#_Toc14069_WPSOffice_Level2)

[2.2宿管操作说明](#_Toc7971_WPSOffice_Level2) [13](#_Toc7971_WPSOffice_Level2)

[2.2.1宿管登录2-2-1图](#_Toc12020_WPSOffice_Level2) [13](#_Toc12020_WPSOffice_Level2)

[2.2.2宿管注册2-2-2图](#_Toc11045_WPSOffice_Level2) [14](#_Toc11045_WPSOffice_Level2)

2.2.3[宿管主页2-2-3图](#_Toc26006_WPSOffice_Level2) [14](#_Toc26006_WPSOffice_Level2)

2.2.4[个人操作2-2-4图](#_Toc21350_WPSOffice_Level2) [15](#_Toc21350_WPSOffice_Level2)

2.2.4[发布消息2-2-5图](#_Toc29695_WPSOffice_Level2) [15](#_Toc29695_WPSOffice_Level2)

2.2.5[公告信息2-2-6图](#_Toc20419_WPSOffice_Level2) [16](#_Toc20419_WPSOffice_Level2)

[2.3维修人员操作说明](#_Toc20667_WPSOffice_Level2) [16](#_Toc20667_WPSOffice_Level2)

2.3.1[维修登录2-3-1图](#_Toc27191_WPSOffice_Level2) [16](#_Toc27191_WPSOffice_Level2)

2.3.2[维修注册2-3-2图](#_Toc32013_WPSOffice_Level2) [17](#_Toc32013_WPSOffice_Level2)

2.3.3[维修主页2-3-3图](#_Toc18783_WPSOffice_Level2) [17](#_Toc18783_WPSOffice_Level2)

2.3.4[个人操作修改2-3-4图](#_Toc11737_WPSOffice_Level2) [18](#_Toc11737_WPSOffice_Level2)

2.3.[5查看维修任务2-3-5图](#_Toc23626_WPSOffice_Level2) [18](#_Toc23626_WPSOffice_Level2)

[2.3.6公告信息2-3-6图](#_Toc21704_WPSOffice_Level2) [19](#_Toc21704_WPSOffice_Level2)

**[3. 管理员功能操作说明](#_Toc13521_WPSOffice_Level1)** **[19](#_Toc13521_WPSOffice_Level1)**

[3.1 管理员登录](#_Toc22919_WPSOffice_Level2) [19](#_Toc22919_WPSOffice_Level2)

3.1.1[管理员登陆3-1图](#_Toc21395_WPSOffice_Level2) [19](#_Toc21395_WPSOffice_Level2)

[3.2管理员主页](#_Toc5061_WPSOffice_Level2) [19](#_Toc5061_WPSOffice_Level2)

3.2.1[管理员主页3-2图](#_Toc12952_WPSOffice_Level2) [20](#_Toc12952_WPSOffice_Level2)

[3.3数据管理](#_Toc14488_WPSOffice_Level2) [20](#_Toc14488_WPSOffice_Level2)

3.3.1[学生管理3-3-1图](#_Toc15652_WPSOffice_Level2) [20](#_Toc15652_WPSOffice_Level2)

[3.3.2学生管理编辑3-3-2图](#_Toc29195_WPSOffice_Level2) [21](#_Toc29195_WPSOffice_Level2)

[3.3.3宿管管理3-3-3图](#_Toc14625_WPSOffice_Level2) [21](#_Toc14625_WPSOffice_Level2)

3.3.4[宿管管理编辑3-3-4图](#_Toc27184_WPSOffice_Level2) [22](#_Toc27184_WPSOffice_Level2)

3.3.5[维修管理3-3-5图](#_Toc23806_WPSOffice_Level2) [22](#_Toc23806_WPSOffice_Level2)

3.3.6[维修编辑3-3-6图](#_Toc26578_WPSOffice_Level2) [23](#_Toc26578_WPSOffice_Level2)

3.3.7[资源管理3-3-7图](#_Toc32269_WPSOffice_Level2) [23](#_Toc32269_WPSOffice_Level2)

3.3.8[维修任务分配3-3-8图](#_Toc13798_WPSOffice_Level2) [24](#_Toc13798_WPSOffice_Level2)

3.3.9[维修任务详情3-3-9图](#_Toc688_WPSOffice_Level2) [24](#_Toc688_WPSOffice_Level2)

3.3.10[维修分配3-3-10图](#_Toc21372_WPSOffice_Level2) [25](#_Toc21372_WPSOffice_Level2)

[3.3.11身份识别和权限管理3-3-11图](#_Toc16139_WPSOffice_Level2) [25](#_Toc16139_WPSOffice_Level2)

3.3.12[投诉反馈3-3-12图](#_Toc22541_WPSOffice_Level2) [26](#_Toc22541_WPSOffice_Level2)

[3.3.13服务评价统计3-3-13图](#_Toc6564_WPSOffice_Level2) [26](#_Toc6564_WPSOffice_Level2)

**[4. 后台源代码](#_Toc16694_WPSOffice_Level1)** **[26](#_Toc16694_WPSOffice_Level1)**

[4.1 Controller层](#_Toc26647_WPSOffice_Level2) [26](#_Toc26647_WPSOffice_Level2)

[4.2 Dao层](#_Toc16327_WPSOffice_Level2) [73](#_Toc16327_WPSOffice_Level2)

[4.3 Mapping层](#_Toc14731_WPSOffice_Level2) [82](#_Toc14731_WPSOffice_Level2)

[4.4 pojo层](#_Toc11255_WPSOffice_Level2) [96](#_Toc11255_WPSOffice_Level2)

[4.5 Service接口层](#_Toc20221_WPSOffice_Level2) [127](#_Toc20221_WPSOffice_Level2)

[4.6 Serviceimp实现层](#_Toc27328_WPSOffice_Level2) [135](#_Toc27328_WPSOffice_Level2)

[4.7 Util层](#_Toc28137_WPSOffice_Level2) [168](#_Toc28137_WPSOffice_Level2)

**[5. 前台源代码](#_Toc6201_WPSOffice_Level1)** **[178](#_Toc6201_WPSOffice_Level1)**

[5.1 主页面代码](#_Toc29253_WPSOffice_Level2) [178](#_Toc29253_WPSOffice_Level2)

[5.2 页面路由](#_Toc18456_WPSOffice_Level2) [263](#_Toc18456_WPSOffice_Level2)

[5.3 页面接口信息](#_Toc14489_WPSOffice_Level2) [269](#_Toc14489_WPSOffice_Level2)

**1. 系统功能概述**

随着时代的改变和网络的快速发展，当下的生活人们更喜欢与网络打交道，可以免去生活中很多复杂和繁琐的步骤。特别是当前网络主流阶段，大学生更是离不开网络。

每一年高考过后都会有大量新生涌入大学校园，这时候寝室就是我们大学四年的家，既然是我们的家，难免就会有损坏的时候，当桌子、椅子、厕所、窗户等其他地方出现损坏的时候，我们就得去报修这些相应的东西。但是自己去报修是挺麻烦的，要按时申请、排队、填表、预约时间、等待维修等一系列的繁琐操作之后才能把寝室那损坏的地方给报修成功。因此，为了解决这一系列的繁琐工序，许多的大学校园也相应产生给学生报修寝室的平台，这一系列的报修平台同样允许学生在上面根据自己需求进行报修。但是，我们会发现这个报修的功能在整个平台只是冰山一角，随着近几年的大学相应的扩招，学生数量增加，寝室数量并不能快速增加，所以更多的学生共同使用寝室资源，相应的损坏几率、损坏程度、位置等都相应增加。因此，普通的寝室报修并不能更好的适应当前学生的需求。再者是校园网都是一个比较综合性的网站，并没有针对寝室报修这一块有相应的平台。所以，着手搭建一个针对校园寝室报修管理系统是一个很实用，他可以针对性解决寝室报修这一块，免去一切繁琐的过程。这是很有意义的。

因此，针对这个问题。遵循软件工程开发规范，着手设计并实现一个基于B/S结构的寝室报修管理系统，目的是为了解决寝室报修的繁琐过程、减少时间的浪费、增加报修效率、增加寝室维修进度等，方便学生报修、方便维修人员进行维修工作。这样子使得寝室报修更加的方便、快捷，针对性的维修。快捷的使用方式，学生只需要上上网便可轻松将自己寝室损坏之处进行报修，避免繁琐过程。而且里面给学生增加了投诉功能，学生可以根据自己的寝室的维修效果反馈和分享自己寝室，便于学生的及时反馈。而且学生实时关注本楼栋的公告信息，关注寝室动态。

寝室报修管理系统根本性的解决了大部分校园的寝室报修问题，针对性的报修平台，细致的描述报修，学生更加方便使用该平台对寝室进行报修、进行寝室分享等基本操作。维修人员更快捷的处理报修任务，避免任务的杂乱。宿管可以通过发布通告形式通知学生最近楼栋情况，学生可以及时预防和处理。同时为了更好的管理，管理员可以对数据进行修改、维护等。

整个寝室报修管理系统主要分为俩个大模块，前台和后台俩个部分，前台主要负责界面设计、实现用户操作等。前台用户分为四种用户，分别是学生、维修人员、宿管、管理员（1个）。

## 1.1学生详细功能：

1. 学生登录

学生通过输入自己的学号、密码进行登录进入该系统，选择自己的标识登录，标识不正确登录不成功，得重新登录。

1. 学生注册

学生在没有账号和密码之前可以先通过注册页面输入相关信息注册自己的账号。

1. 学生主页

该页面显示当前学生可操作的功能模块和学生分享的寝室图，当然也可以投入广告页面或者公告等。

1. 学生个人操作

这里面又分为俩个小模块：

a.修改信息：学生可以通过此处修改自己变更或者出错的信息。

b.寝室共享：学生可以通过该页面分享自己漂亮的寝室环境与他人欣赏。

1. 学生寝室报修

这是一个主要的界面，主要实现学生此出对寝室的损坏情况进行报修，通过描述寝室损坏情况，损坏位置，上传图片等进行报修。

1. 公告信息

学生通过此出查看楼栋的相关通知，避免学生跑到楼下查看通知，学生可以时刻关注寝室所在楼栋的情况，便于及时的预防和获取信息，并且还可以获取最近的评价信息，通过这些评价学生可以自主选择维修人员来维修自己寝室。

1. 维修评价

学生通过在完成报修之后，对维修的效果进行评价，并且以最高分五分的打分制形式对此次维修进行评价。

1. 投诉

学生对本次报修结果不满意或者有损坏，又或是出现东西丢失等相关情况进行投诉。

1. 回馈信息

当学生选择回馈信息时，便可以看到管理员反馈学生投诉的信息。

1. 退出登录

学生可以通过点击此处退出本次系统登录

## 1.2宿管详细功能

1. 宿管登录

宿管可以登录该系统，并且可以进行一系列操作，并且选择自己的标识。

1. 宿管注册

如果宿管没有该账号，就可以通过注册页面进行输入录入账号进行注册。然后返回登录，并且本注册界面与学生的有差别，相应录入的详细信息也不相同。

1. 宿管主页

该页面与学生主页基本一致，主要是操作功能改变了。

1. 宿管个人操作

宿管可以通过该功能来对本身的信息进行修改、保存。当然宿管也可以分享自己的寝室。

1. 宿管寝室报修

宿管同样可以使用报修功能，对自己的寝室进行报修。

1. 宿管消息发布

该功能主要是发布本楼栋的消息，宿管可以通过该功能发布一系列的通告，学生可以通过其他查看宿管发布的公告。便于即使通知学生，避免再使用A4纸粘贴墙壁，提高通知效率。

1. 公告信息

同样，宿管可以通过公告信息功能查看自己发布的公告，如果发现不合理的地方可以重新发布该公告。

1. 退出登录

宿管点击退出本次登录。

## 1.3维修人员详细功能

1. 维修人员登录

维修人员可以通过该界面进行登录，登陆时得选择自己的标识进行登录。

1. 维修人员注册

该页面给维修人员提供注册，主要注册输入：职工号、年龄、密码等数据，与学生注册和宿管注册不相同。

1. 维修人员个人操作

维修人员通过该页面对自己录入的详细信息进行修改保存。

1. 查看维修任务

当维修人员选择该页面，将会展现当前维修人员的维修任务和当前所有的维修任务，这样子维修任务可以按照自己的时间安排私下与其他交换任务也是没有问题的。并且维修人员可以接受该任务，当任务完成之后，维修人员将点击任务完成。

1. 公告信息

维修人员可以通过该页面获取楼栋的通知，即使对自己等任务进行处理或调整。

1. 退出登录

点击退出登录。

## 1.4管理员详细功能

1. 管理员登录

管理员通过选择标识和输入管理员账号、密码进行登录，由于便于数据的安全性考虑，管理员只有一个，所以管理员不需要注册。

1. 管理员主页

该页面功能比较多，管理员可以操作所有的功能。

1. 管理员数据管理

a.学生管理：当管理员进入该页面，就可以对后台数据库的学生的数据进行编辑、保存、删除等操作。用来对学生数据的维护和管理。

b.宿管管理：管理员通过该界面对宿管数据进行操作，主要有编辑、删除等，对宿管数据进行维护和管理。

c.维修管理：该页面对维修人员的数据进行可编辑和删除信息管理。

d.资源管理：管理员可以通过该功能对学生分享的寝室进行编辑、删除等管理，便于数据维护。

e.维修任务分配：管理员通过该页面对学生的提出的报修任务进行分配维修人员对该任务进行维修，该页面主要功能就是给报修任务按合理的安排分配维修人员进行维修。

f.身份识别和权限管理：当宿管和维修人员注册时，会将注册人员数据展现给管理员，由管理员确认该注册人员身份是否属于宿管和维修人员，避免一些“不良人员”冒充注册，当管理员获取注册信息之后给与相应的权限，宿管或维修人员才能够登录成功。

j.投诉反馈：当学生对维修效果或其他的问题提出投诉，管理员给出相应的反馈信息。

h.服务评价统计：主要统计学生对维修服务的评价。

以下是整个系统详细功能图如图1-1所示：



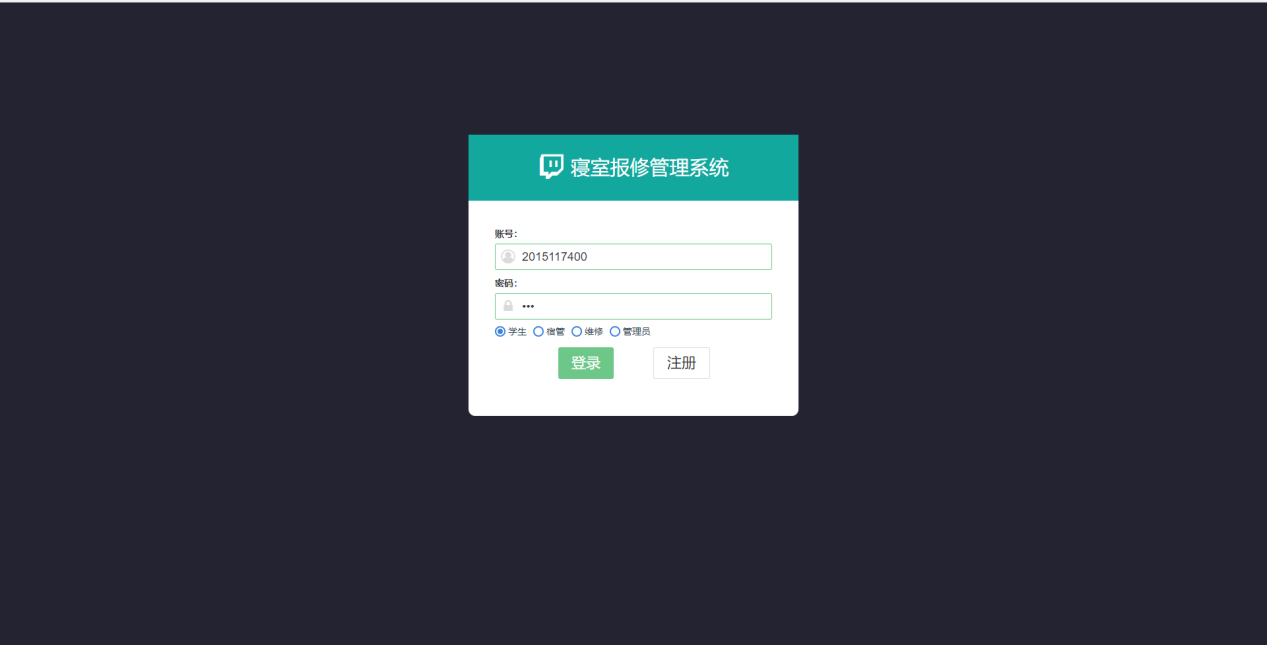
图1-1系统整体详细功能图

# 用户功能操作说明

## 2.1学生操作说明

2.1.1学生登录

学生进入寝室报修管理系统之后，显示学生登录界面，学生输入正确学号、密码之后便可成功登录。如图2-1-1所示。



学生登录2-1-1图

2.1.2学生注册

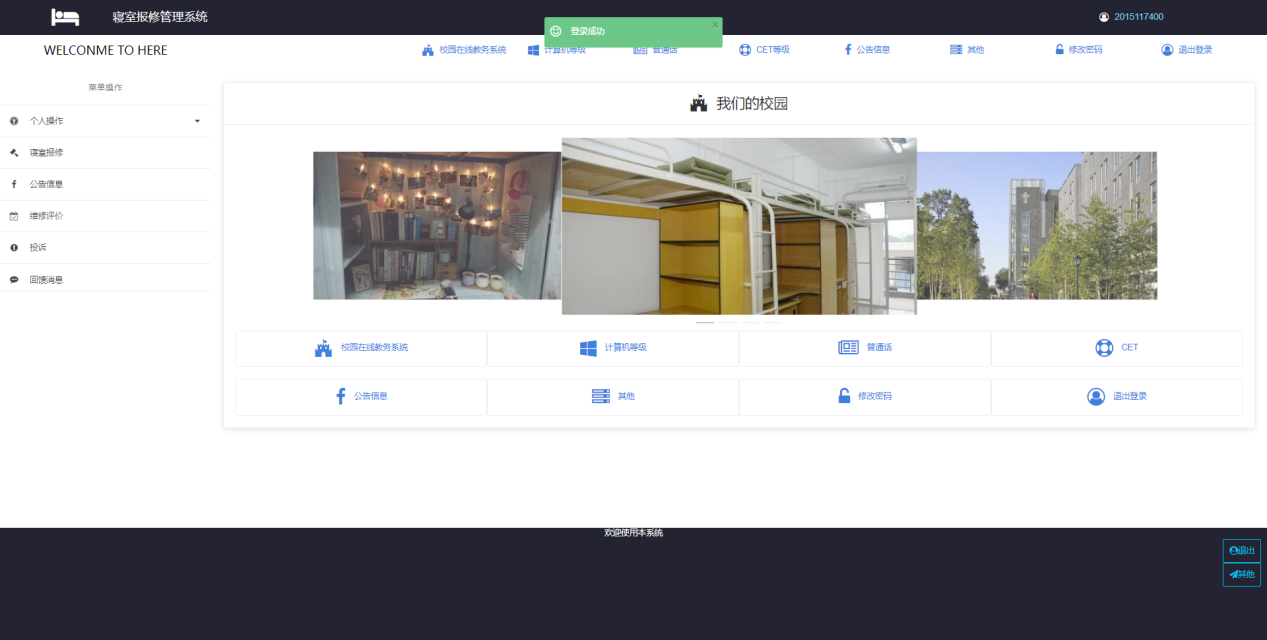
学生如果登录不成功，点击注册跳转到注册页面进行账号信息注册。如图2-1-2所示。



学生注册2-1-2图

2.1.3学生主页

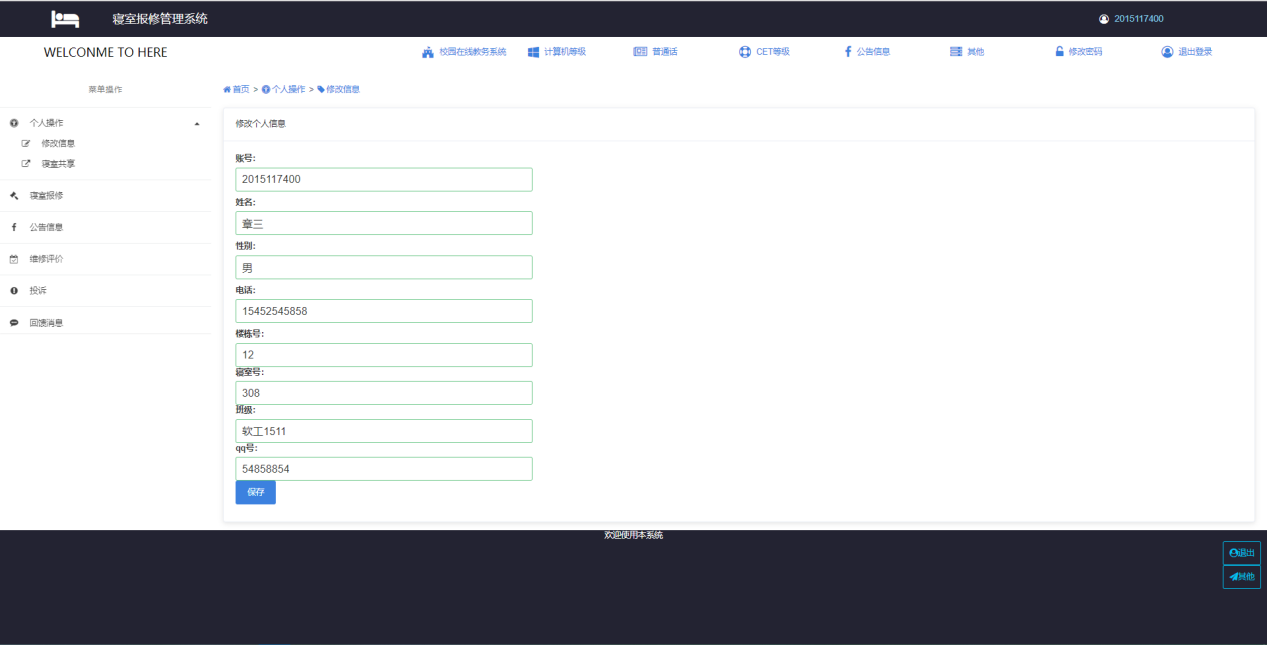
学生登录成功之后的跳转页面，如果学生没有账号或在登录不成功是不会跳转该页面，当然直接访问http://localhost:8081/#/home也同样访问不成功的。该页面主要显示当前用户可以操作的功能模块和展示校园优秀寝室或学生分享的寝室，当然还用提共给学生的一些推荐学习网站等。如图2-1-3所示。



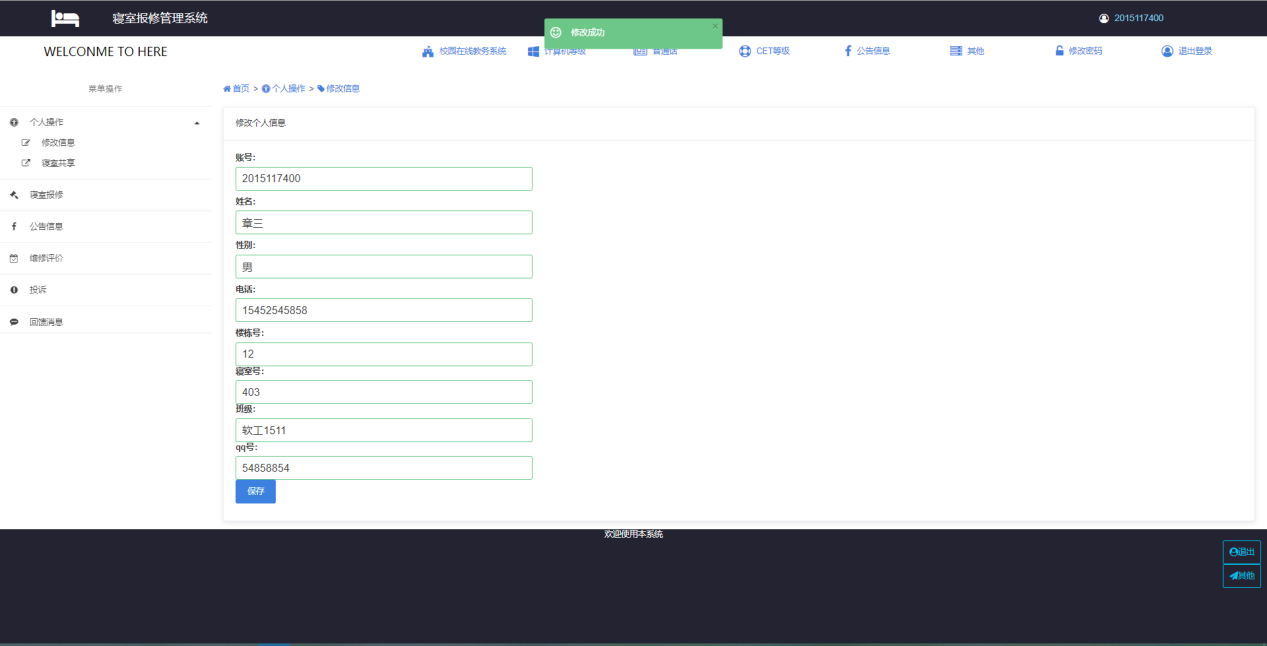
学生主页2-1-3图

2.1.4学生个人操作

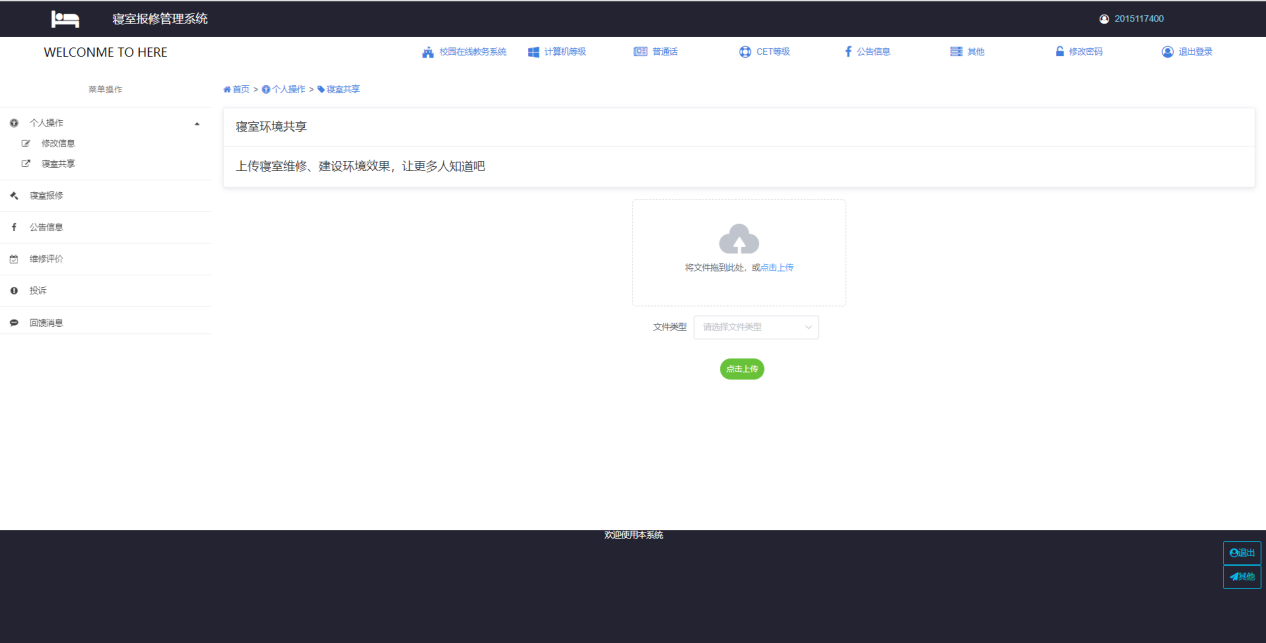
学生登录成功之后可以通过个人操作来对自己的信息进行变更，当然还可以通过该操作进行寝室分享，把自己的寝室环境或维修的成功分享给他人观看。如图2-1-4和2-1-5、2-1-6所示。



学生修改界面2-1-4图



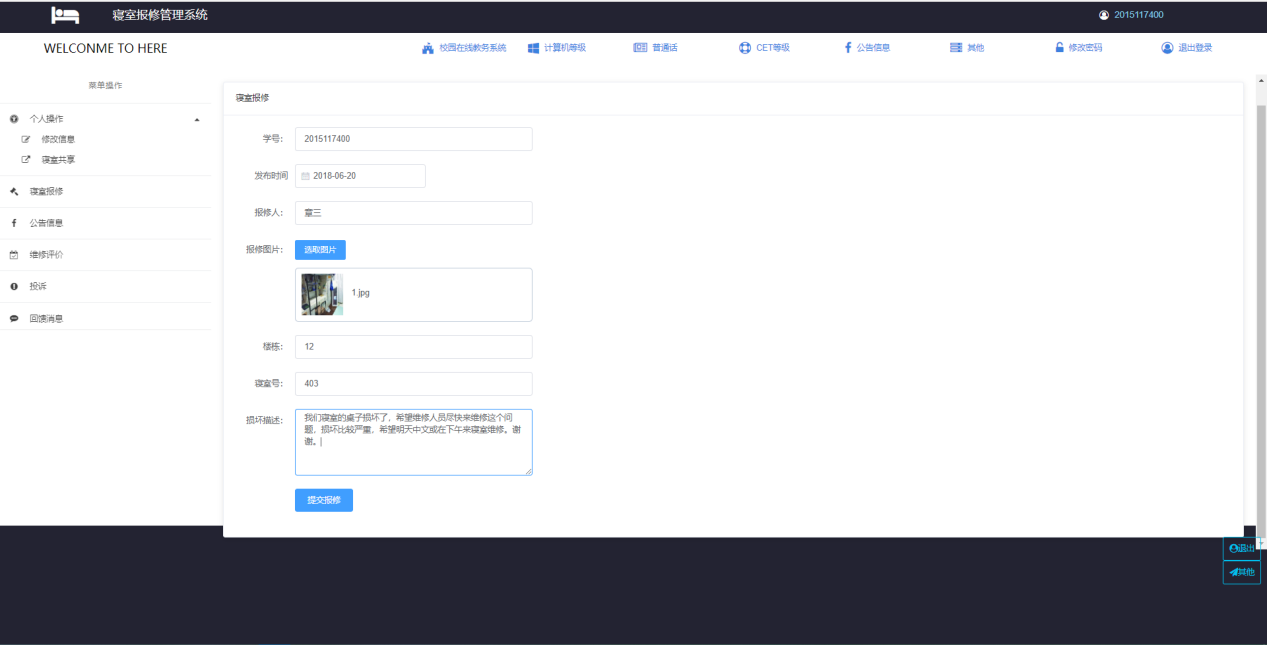
学生修改成功2-1-5图



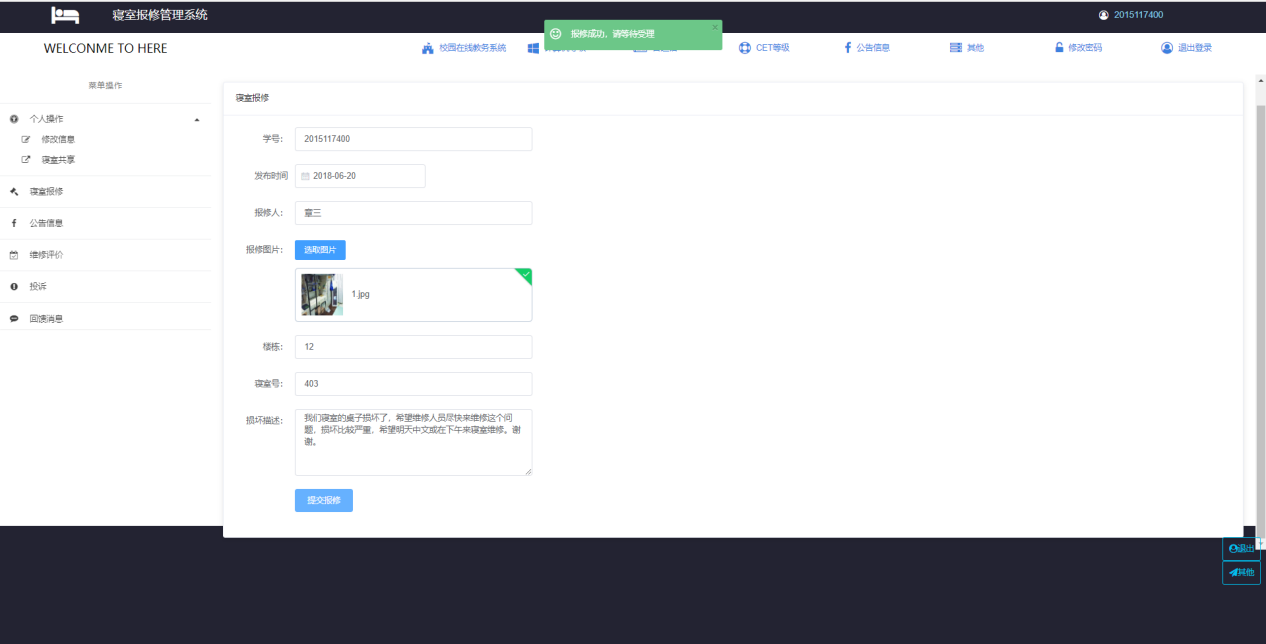
资源分享2-1-6图

2.1.5寝室报修

学生在登录进入主页面之后，选择寝室报修功能，学生可以通过该功能实现对寝室损坏之处进行报修处理，输入相关损坏的描述，提交报修请求，之后等待维修服务即可，无需其他繁琐操作。如图2-1-7和2-1-8所示。



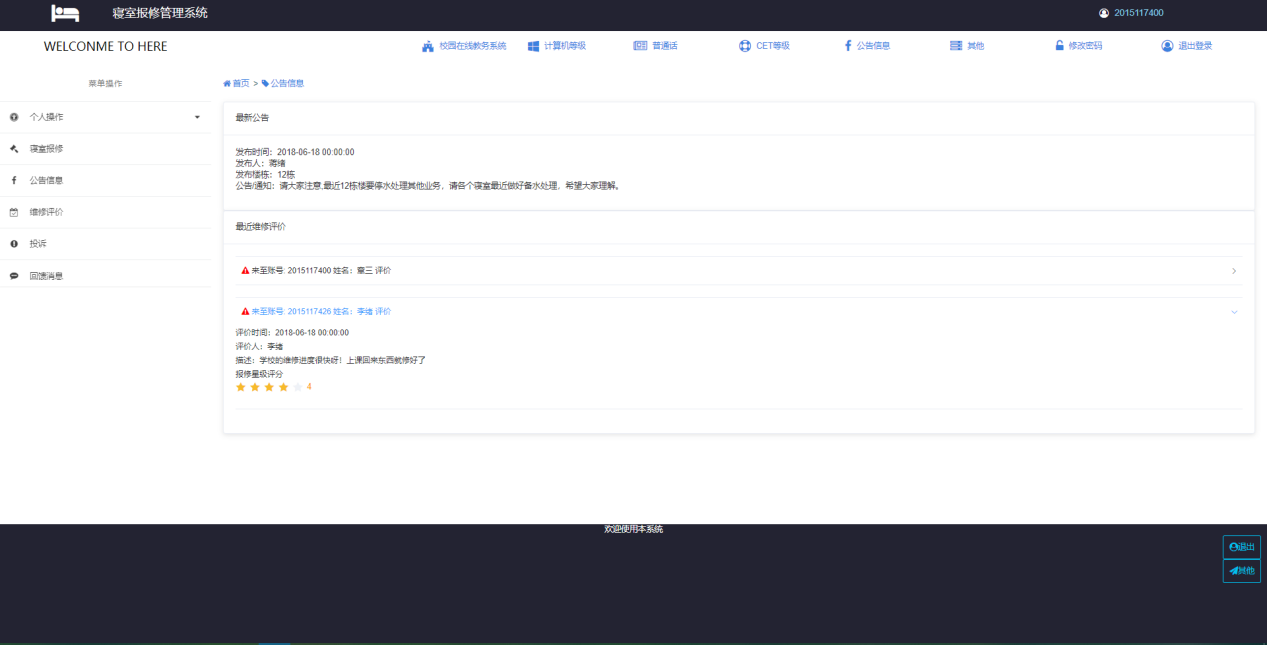
寝室报修2-1-7图



寝室报修成功2-1-8图

2.1.6公告信息

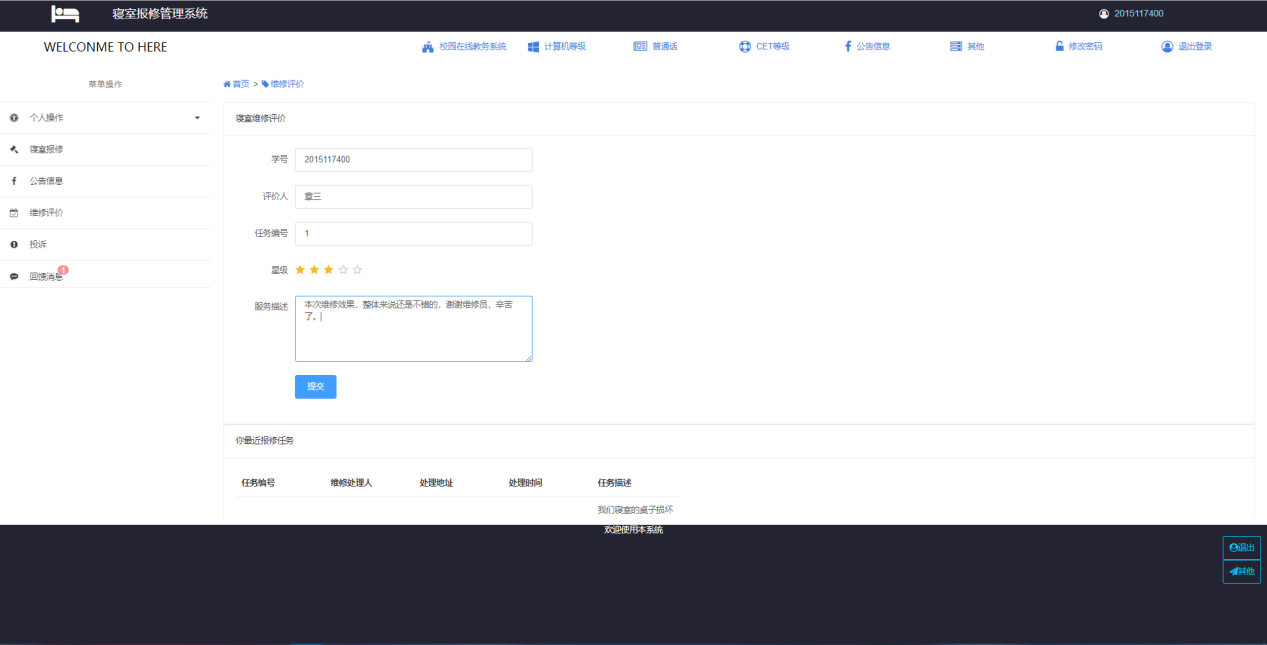
学生选择该功能便可查看有宿管发布寝室楼栋最新的通知或者公告，还有可以查看最近的学生对寝室报修情况的评价信息以及评分信息。如图2-1-9所示。



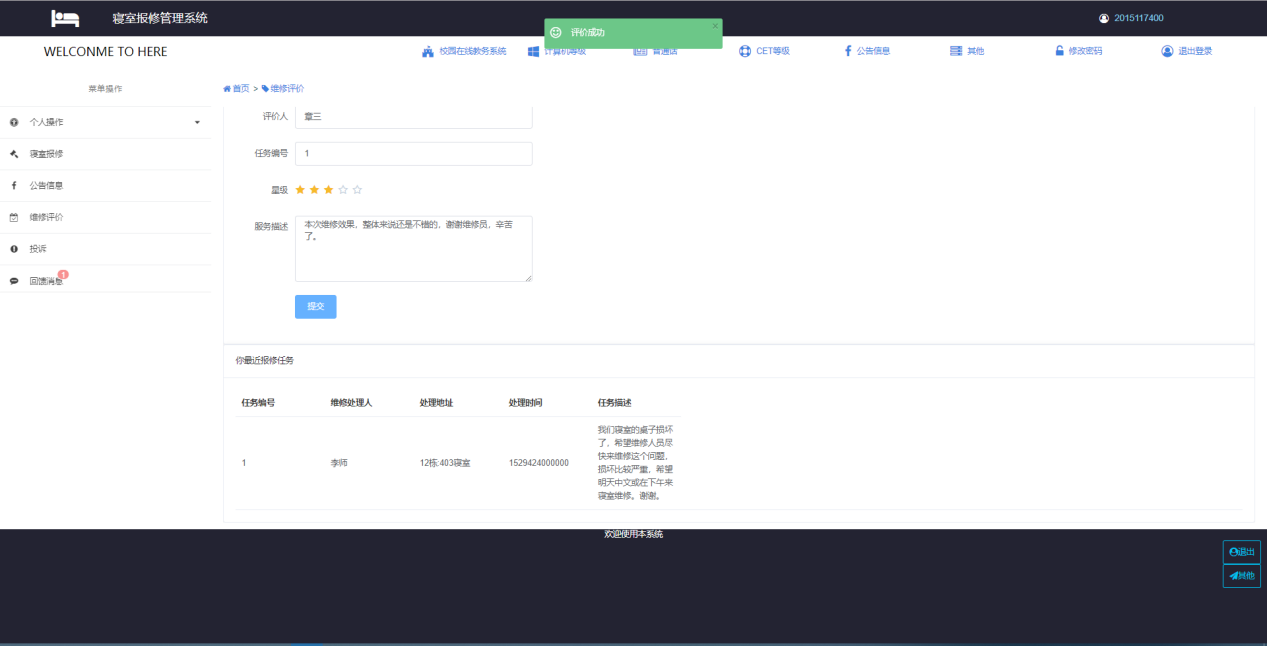
公告信息2-1-9图

2.1.7维修评价

当选择该模块学生可以通过该模块对寝室维修的效果进行评价。当前页面显示你最近报修的任务信息，你可以根据处理你报修的人物进行选择性评价。如图2-1-10和2-1-11所示。



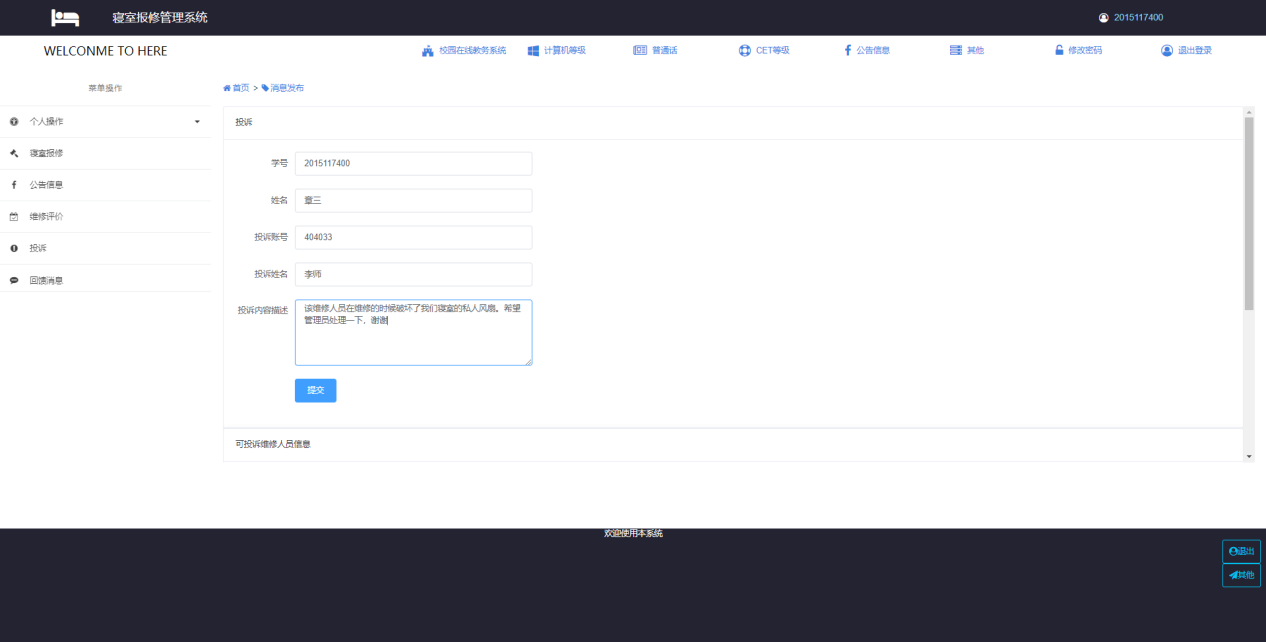
维修评价2-1-10图



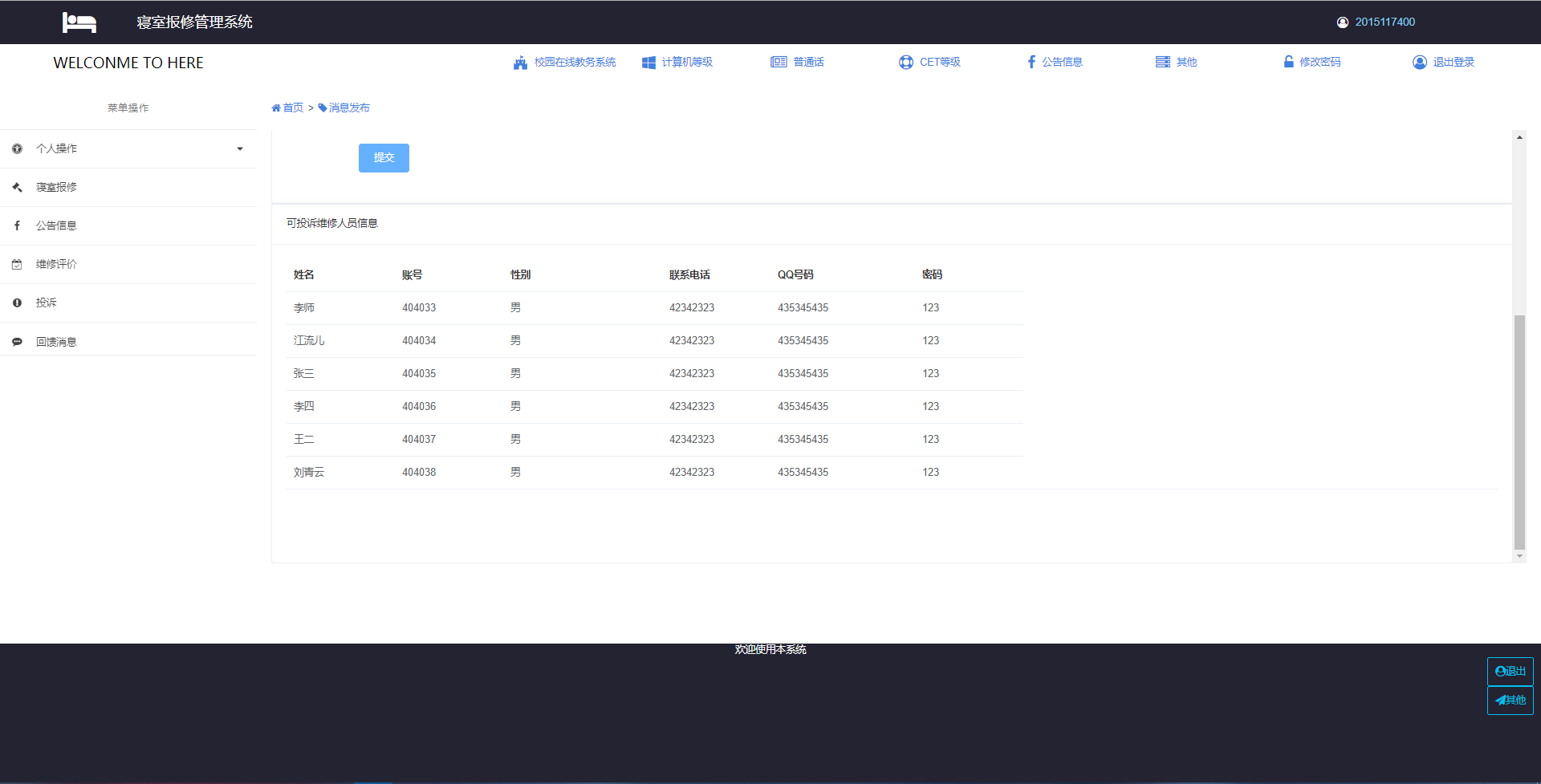
维修评价成功2-1-11图

2.1.8投诉

当学生报修寝室的维修效果不良好或者其他因素想要投诉维修人员可以由该功能发出投诉，当管理员收到之后给你回复。当然学生也可以从此次提出自己对维修的宝贵意见或者建议等，页面下方是维修人员的详细信息显示，方便学生选择要投诉的维修人员。如图2-1-12和2-1-13所示。



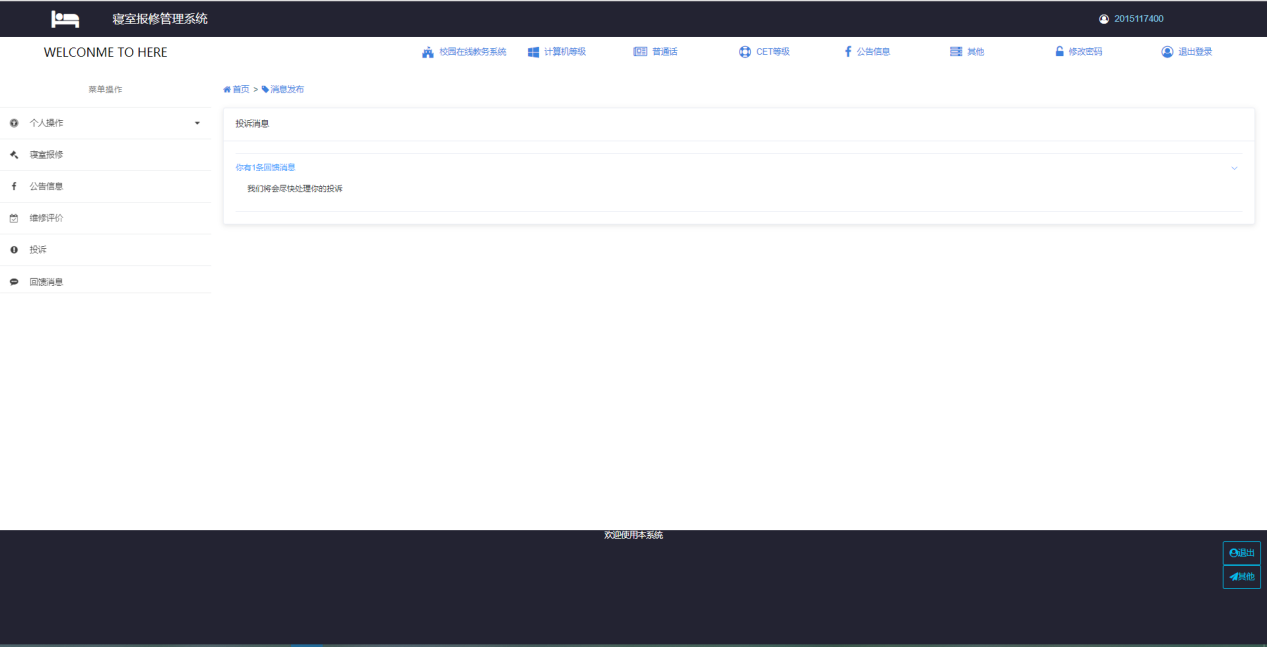
学生投诉2-1-12图



学生投诉可选择信息2-1-13图

2.1.9回馈消息

接收管理员对你的投诉反馈消息，学生进入此页面之后将会获取到管理员对的投诉、意见、建议的反馈。如图2-1-14所示。

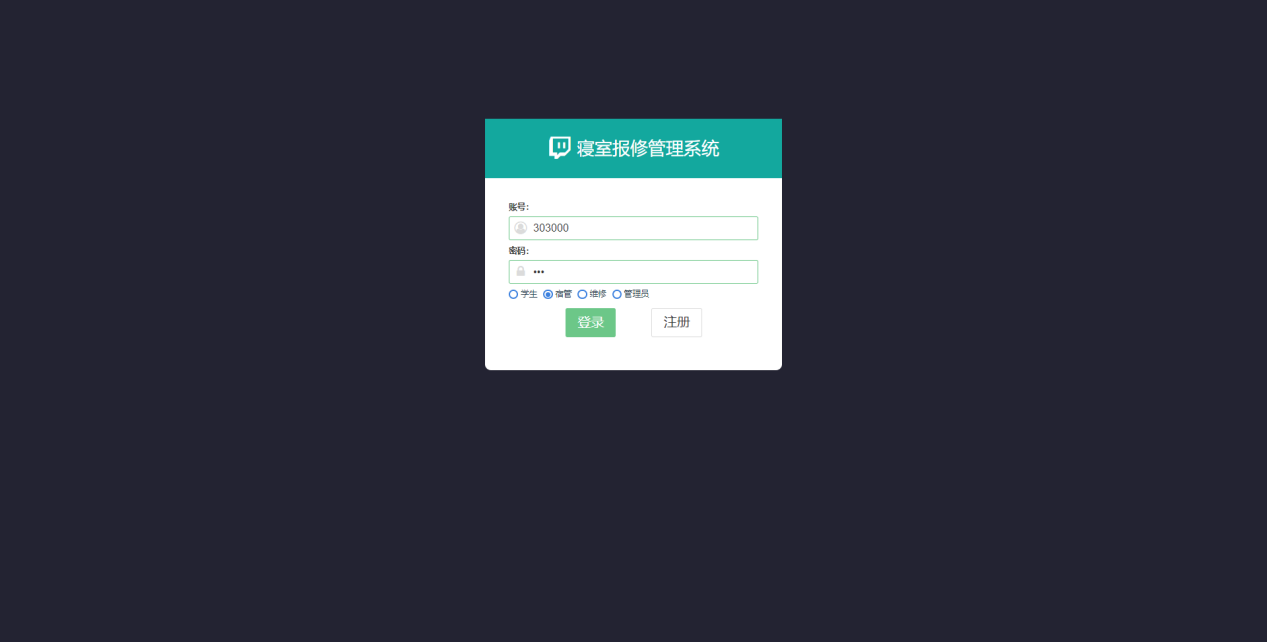


回馈消息2-1-14图

## 2.2宿管操作说明

2.2.1宿管登录

宿管进入寝室报修管理系统之前得登录系统，登录成功之后返回主页界面并且才可进行其他操作。如图2-2-1所示。



宿管登录2-2-1图

2.2.2宿管注册

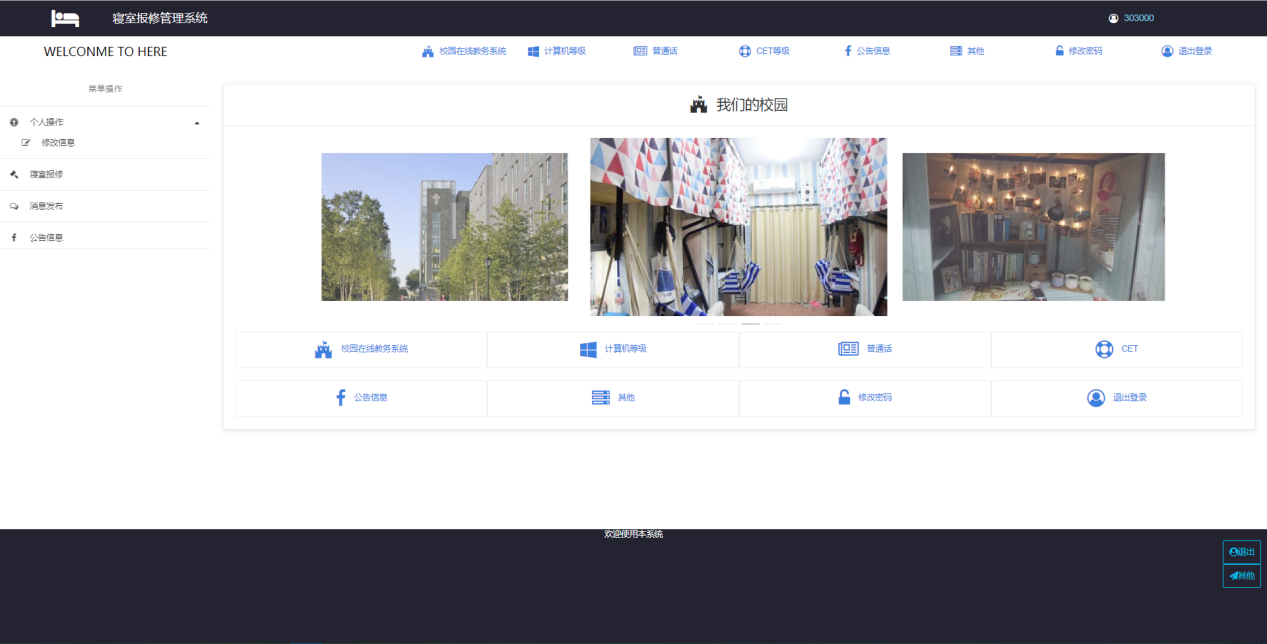
宿管没有注册账号的将要在注册页面利用职工号注册账户信息，等待管理员吧确认之后才能够登录成功，相比与学生注册页面少了寝室号的录入。如图2-2-2所示。



宿管注册2-2-2图

2.2.3主页

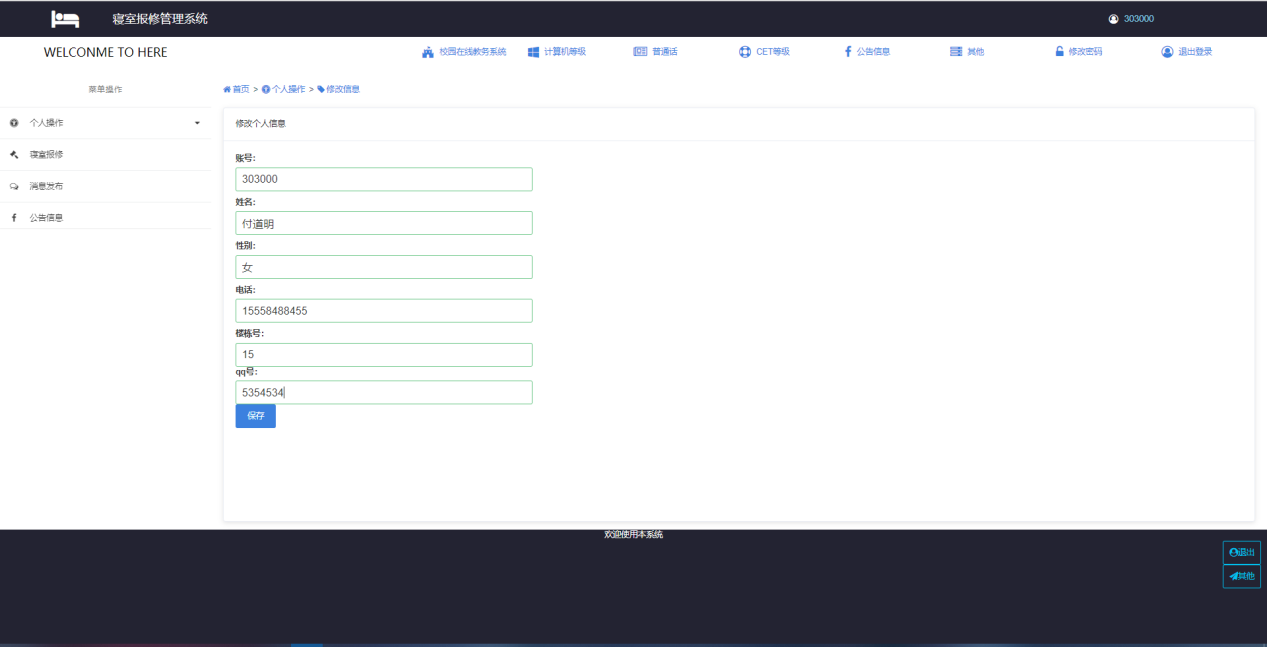
宿管登录成功之后进入主页，主页显示当前宿管可以操作的相关功能。如图2-2-3所示。



宿管主页2-2-3图

2.2.4个人操作

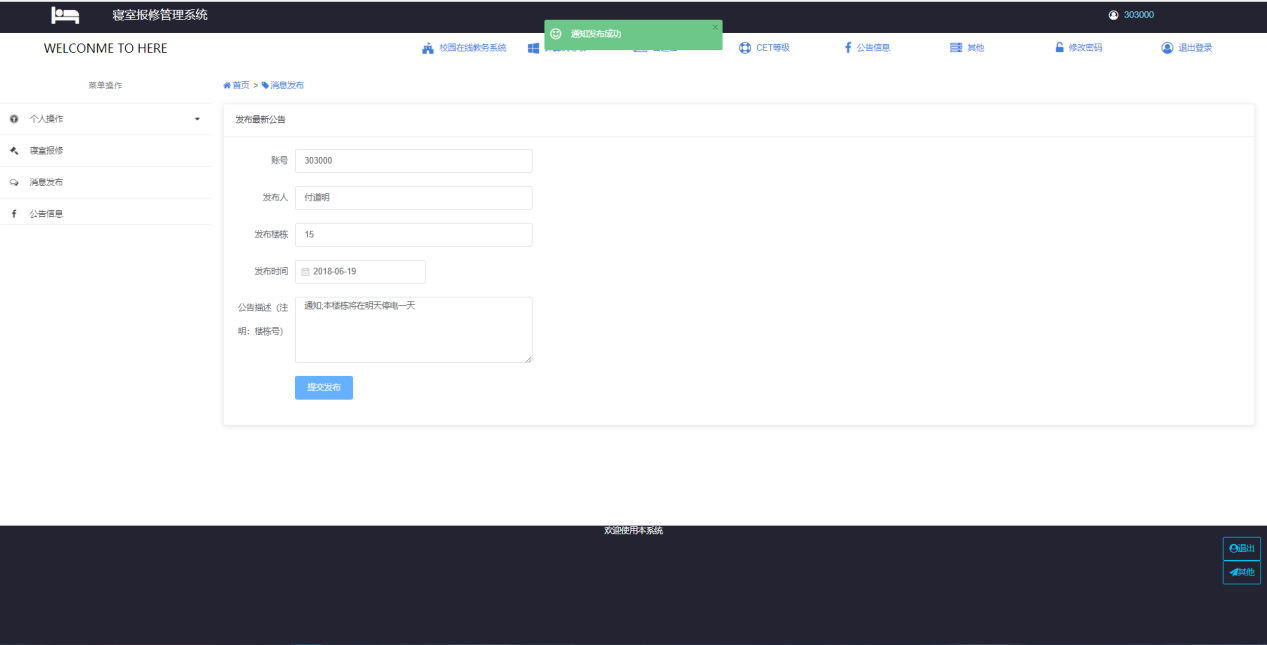
当宿管登录成功之后进入个人操作菜单，菜单里面只有一个功能模块是个人信息修改。宿管可以通过该界面修改本身的信息。如图2-2-4所示。



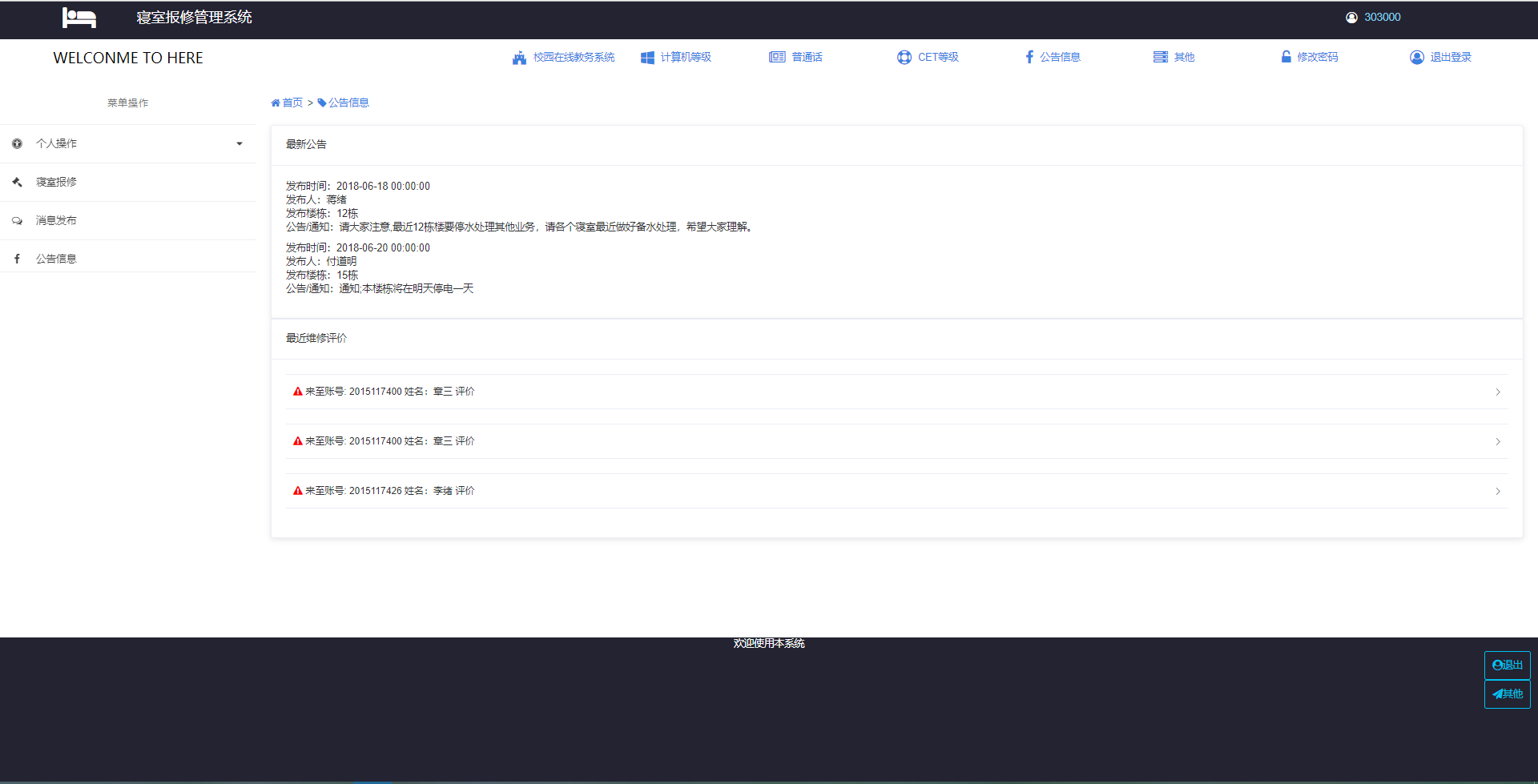
个人操作2-2-4图

2.2.5宿管发布消息

宿管可以通过选择该功能进行楼栋通知发布，便于通知学生最近楼栋的最新消息。当然这上面也可以发布寻物启事、找人等相关消息都可以在此发布。如图2-2-5和2-2-6所示。



发布消息2-2-5图

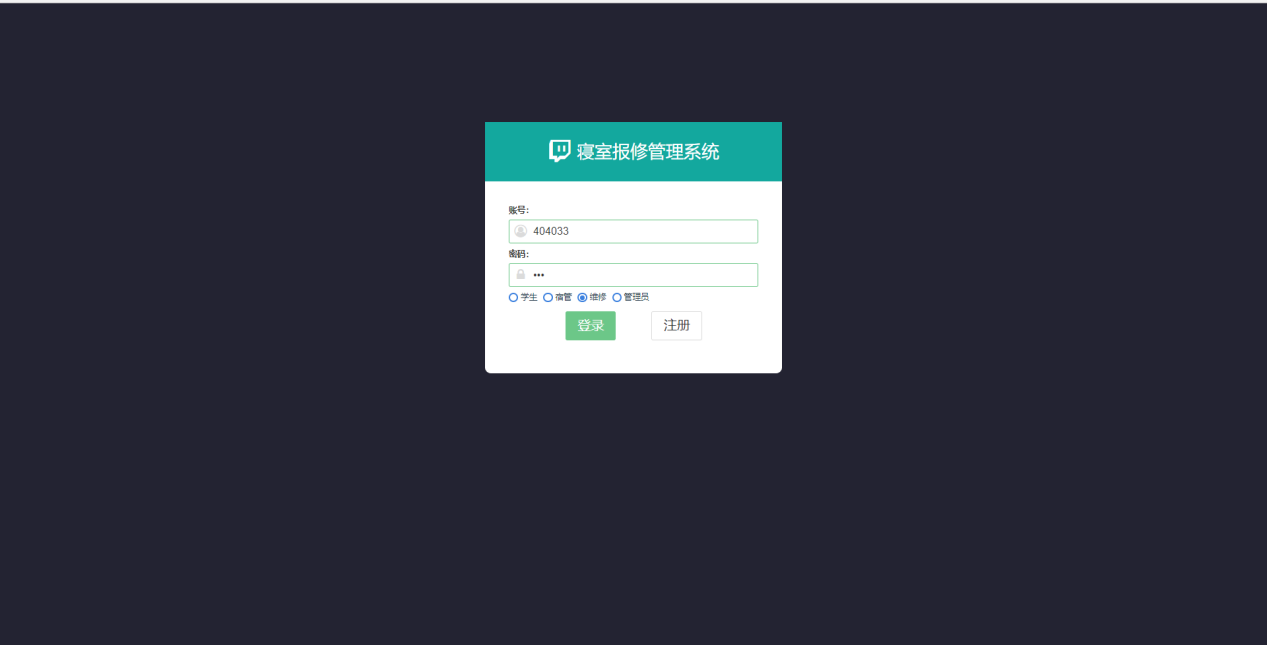


公告信息2-2-6图

## 2.3维修人员操作说明

2.3.1维修登录

维修人员进入登录页面输入本身的账户就可以登录，然后对其功能进行操作。如图2-3-1所示。



维修登录2-3-1图

2.3.2维修注册

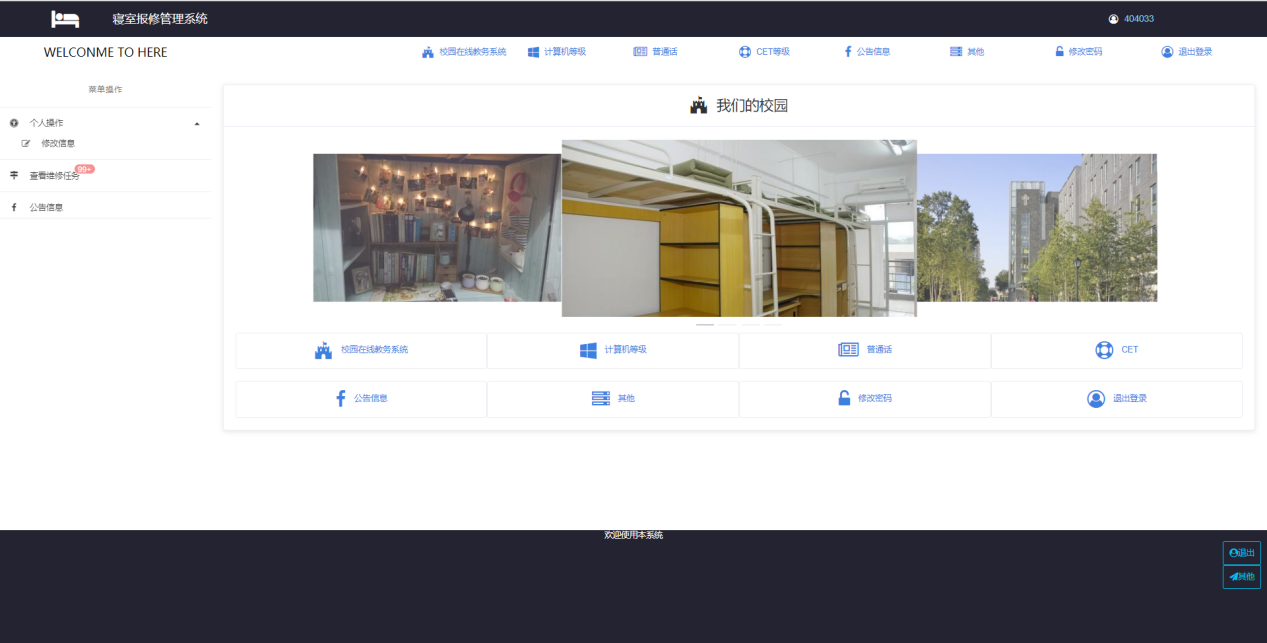
当维修人员没有账号可以通过注册界面进行账号信息注册，等待管理员确认之后便可以登录。如图2-3-2图。



维修注册2-3-2图

2.3.3维修主页

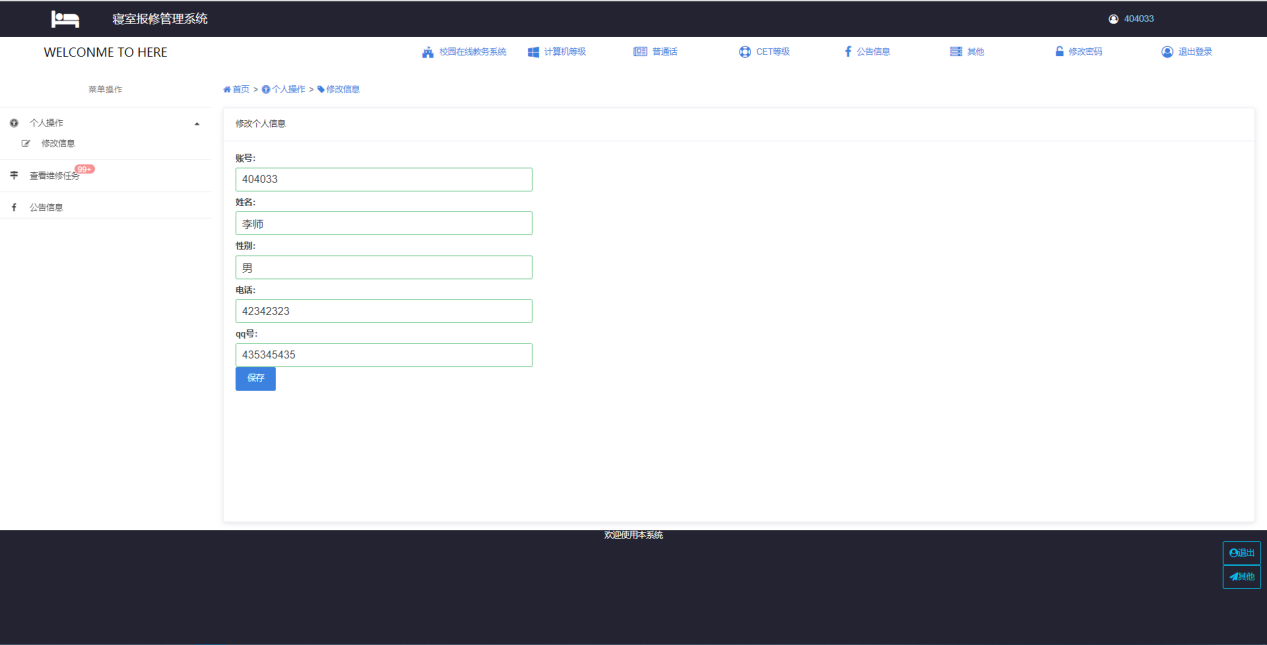
等维修人员登录成功之后，将会跳转路由当主页，展示宿管可以操作的功能模块。如图2-3-3图所示。



维修主页2-3-3图

2.3.4维修个人操作

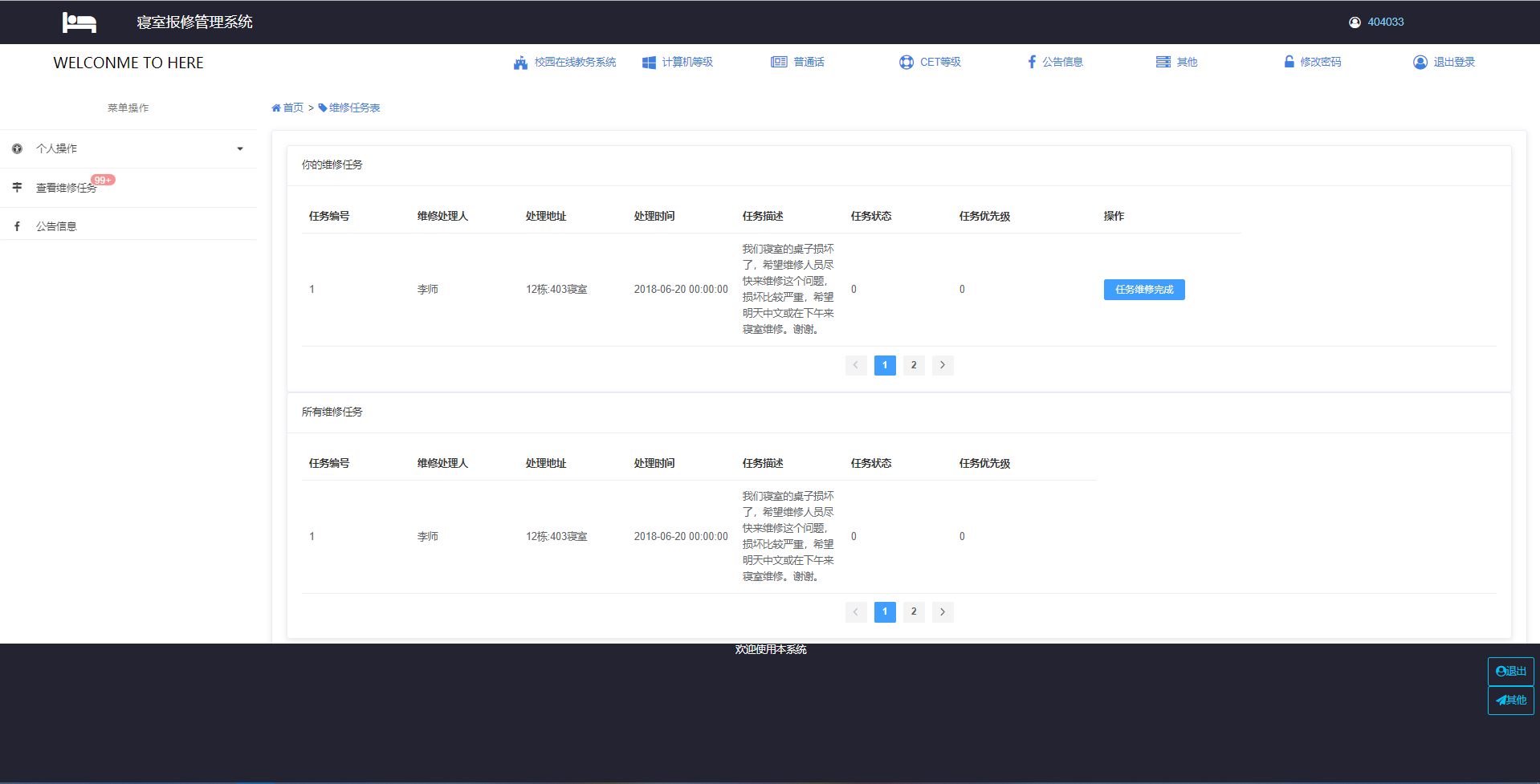
维修人员登录成功之后可以通过该页面修改维修人员的信息。如图2-3-4所示。



个人操作修改2-3-4图

2.3.5查看维修任务

该模块是维修人员的主要模块，该模块提高给维修人员查看当前维修人员的维修任务，维修人员可以由该页面接受任务和完成任务等操作。并且可以查看到所有的任务分布情况，便于合理安排。如图2-3-5所示。



查看维修任务2-3-5图

2.3.6公告信息

维修人员同样可以查看公告信息，由宿管发布的楼栋信息来合理安排自己维修的时间。如图2-3-6所示。

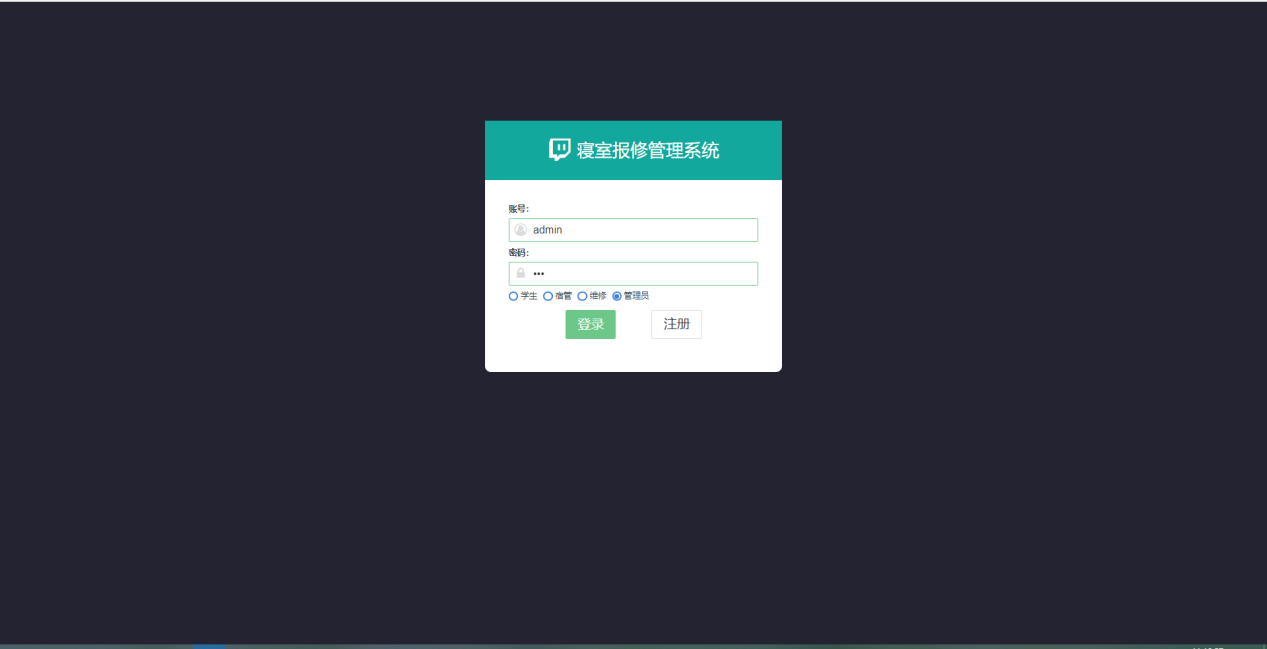


公告信息2-3-6图

# 3. 管理员功能操作说明

## 3.1 管理员登录

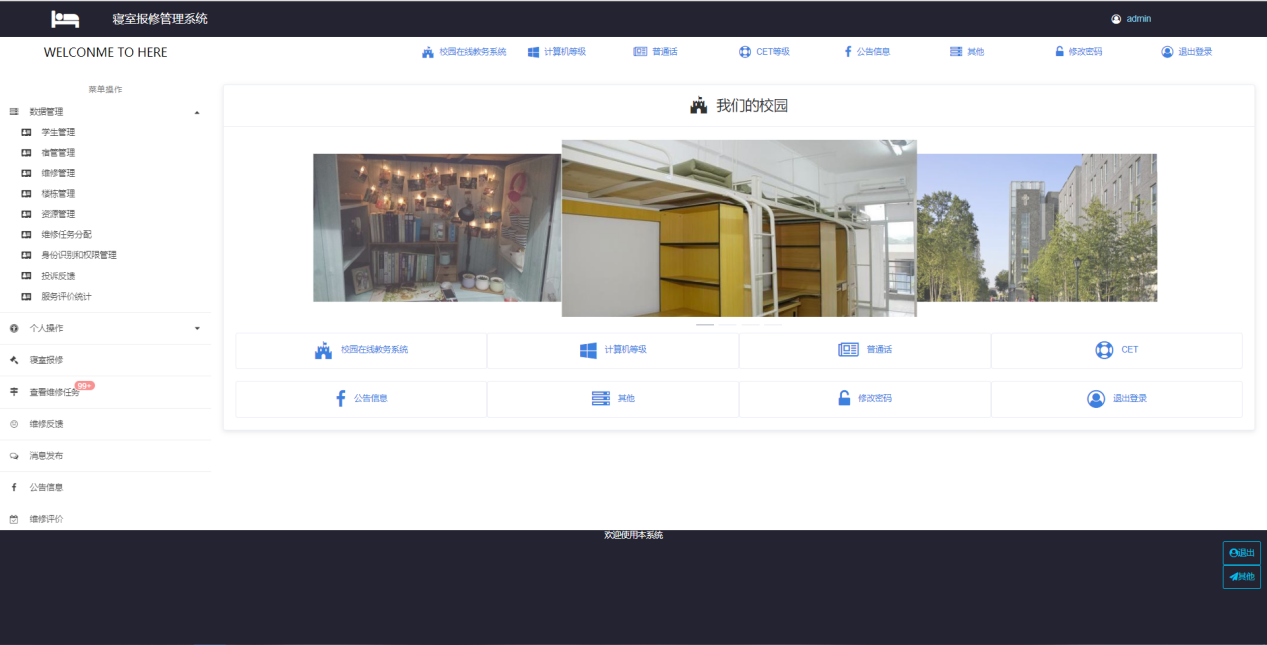
管理员输入账号密码登录，登录成功后跳转主界面。如图3-1所示。



管理员登陆3-1图

## 3.2管理员主页

当管理员登录成功之后的主页。如图3-2所示。

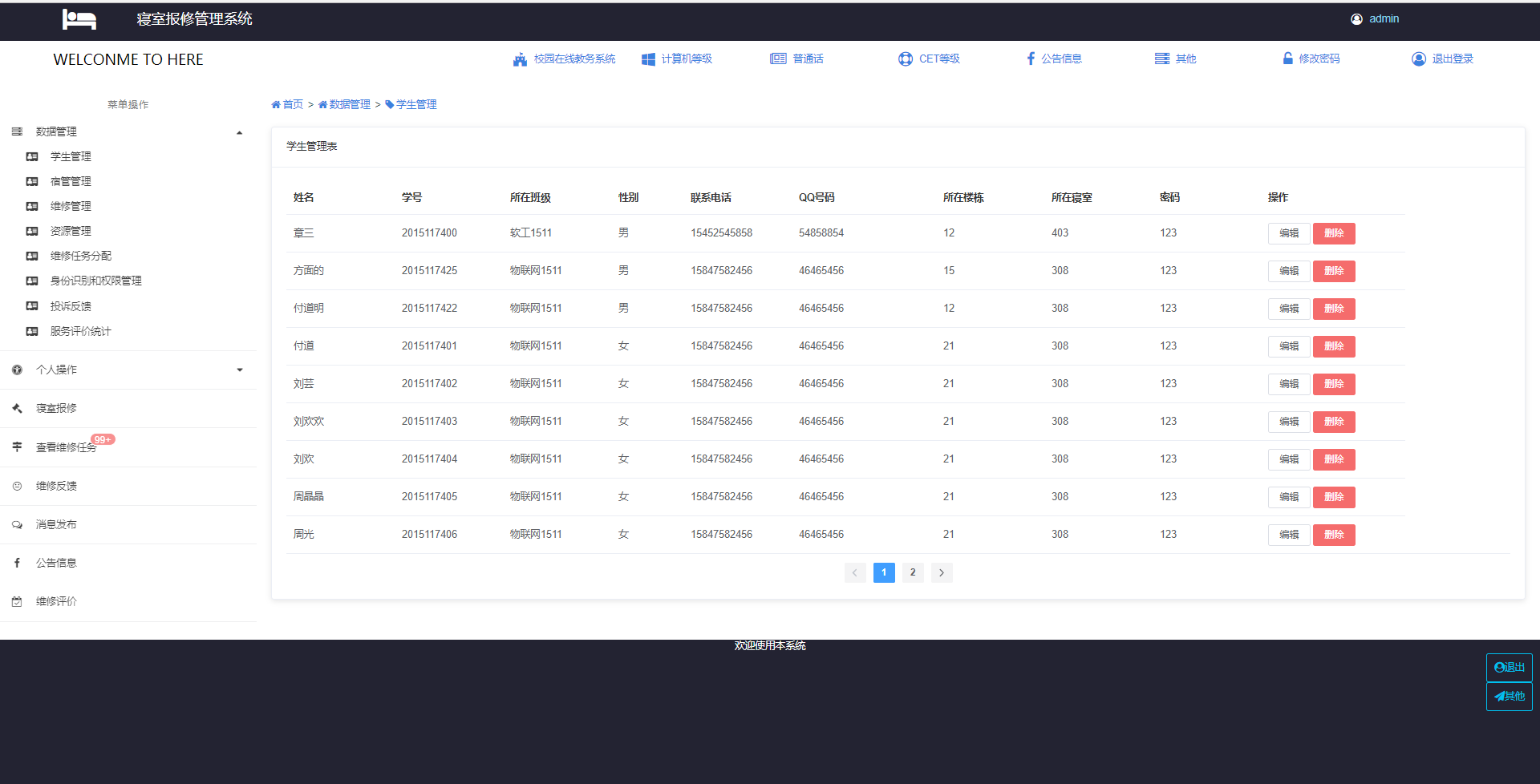


管理员主页3-2图

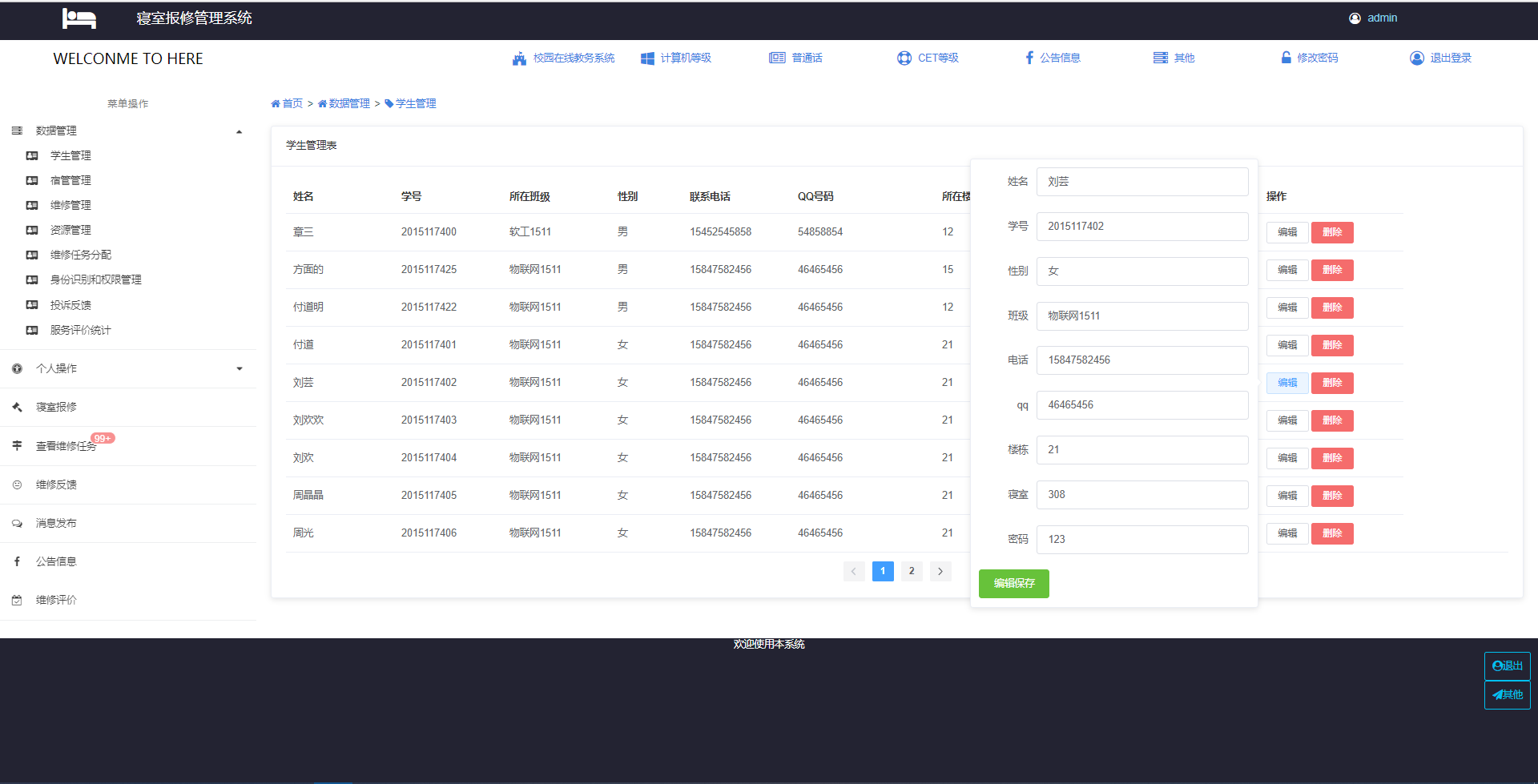
## 3.3数据管理

3.3.1学生管理

当管理员登录成功之后，进入学生管理模块之后，就可以对学生的数据进行管理，比如编辑、删除等操作。如图3-3-1和3-3-2所示。



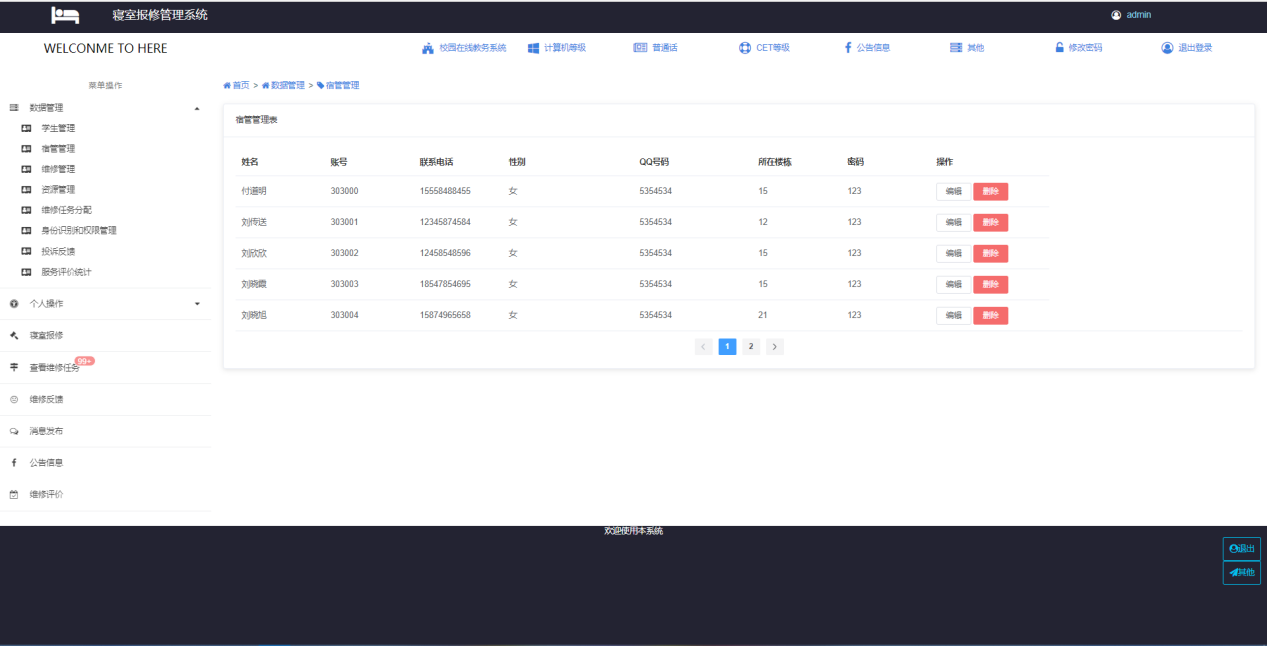
学生管理3-3-1图



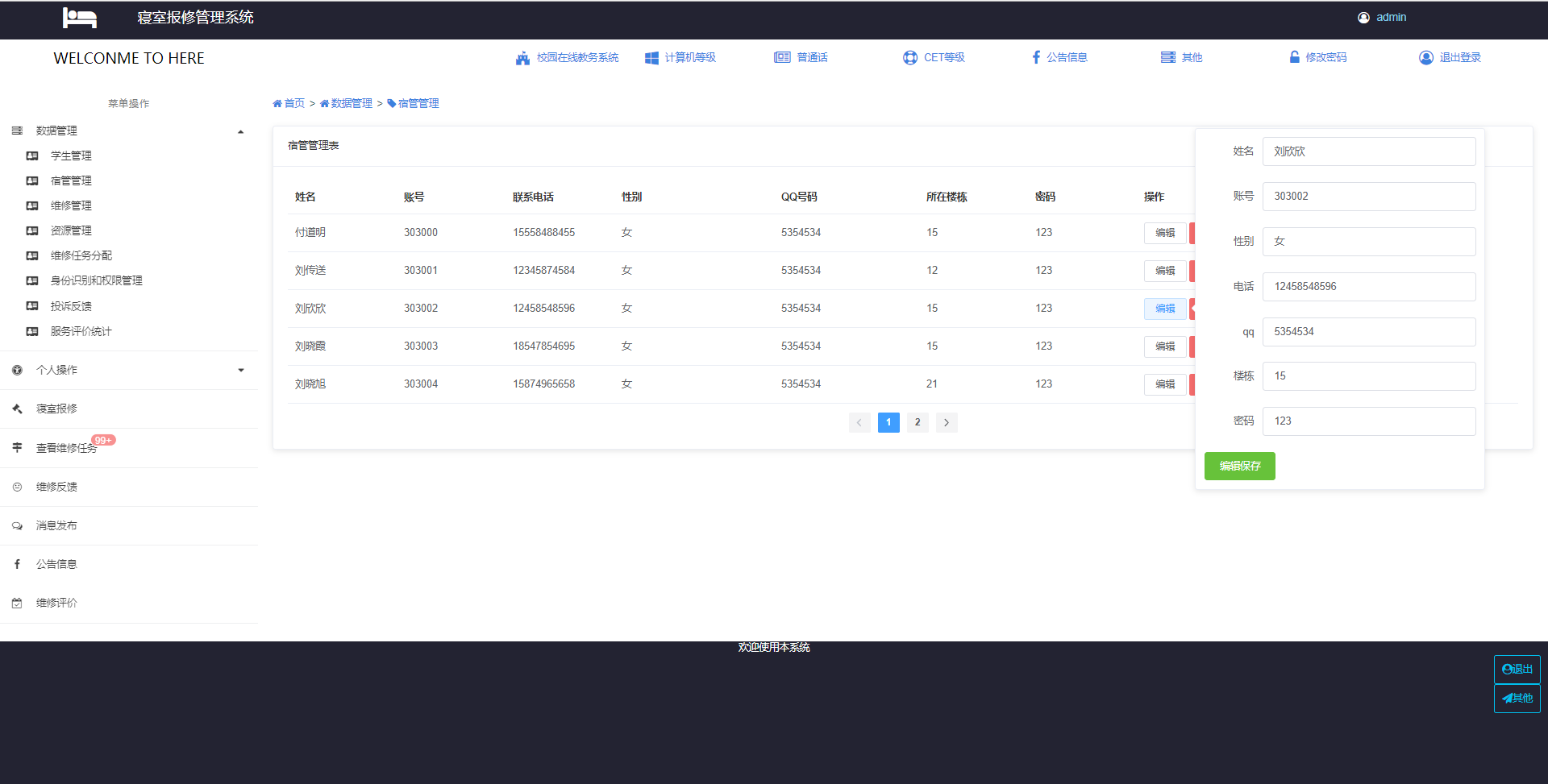
学生管理编辑3-3-2图

3.3.2宿管管理

宿管管理界面，主要是提供给管理员的宿管信息操作，编辑、删除等操作。如图3-3-3和3-3-4所示。



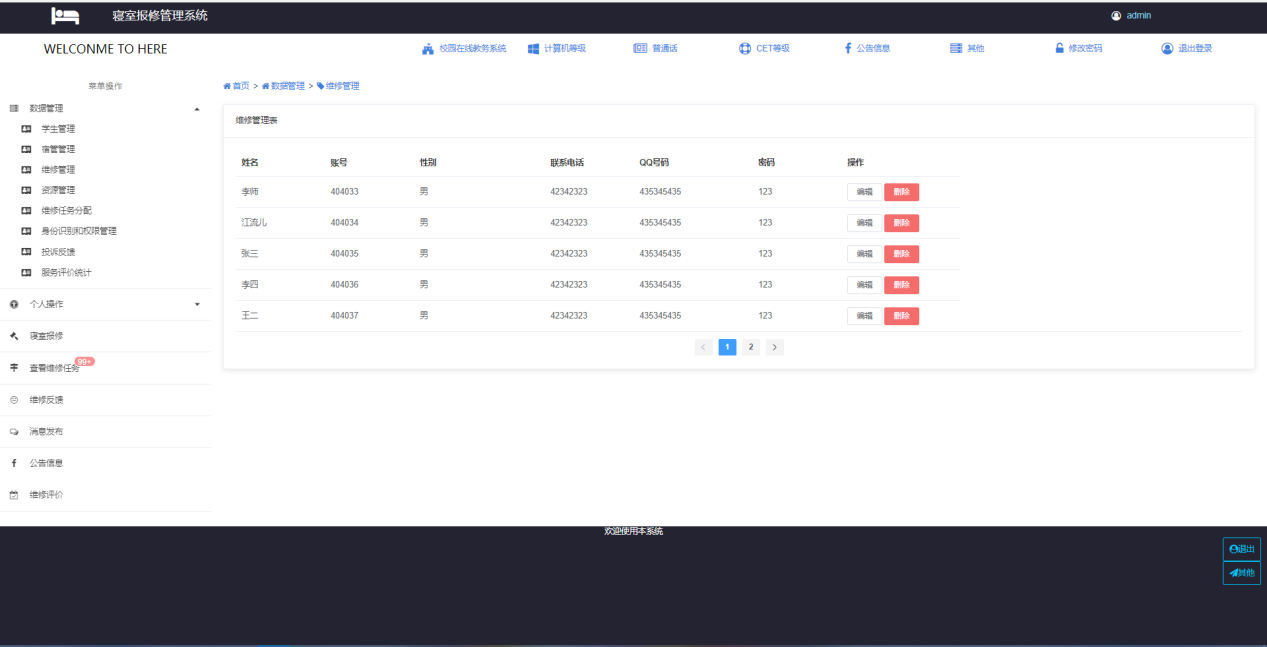
宿管管理3-3-3图



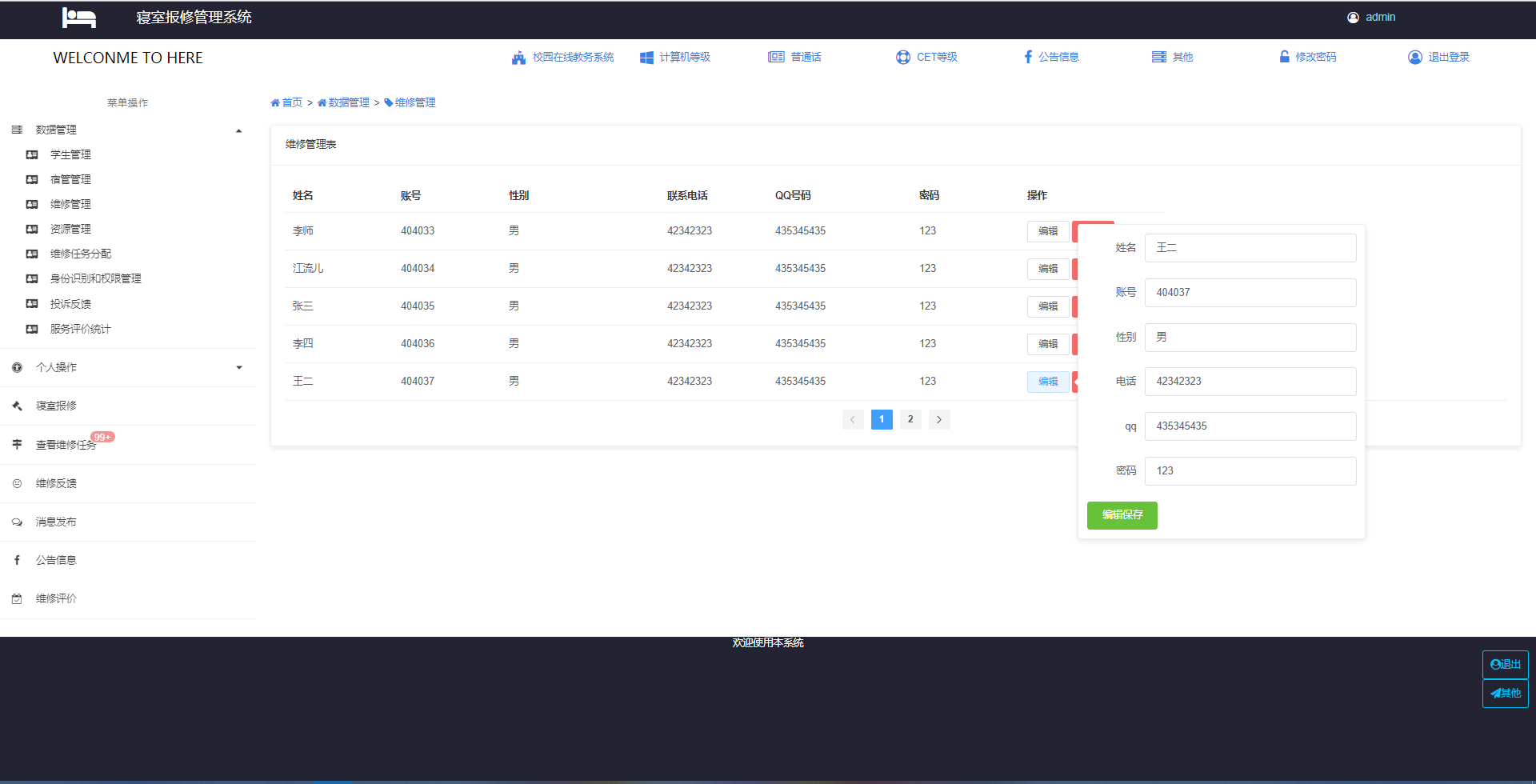
宿管管理编辑3-3-4图

3.3.3维修管理

管理员提高维修管理表对维修人员的消息进行编辑、删除等操作，如图3-3-5和3-3-6所示。



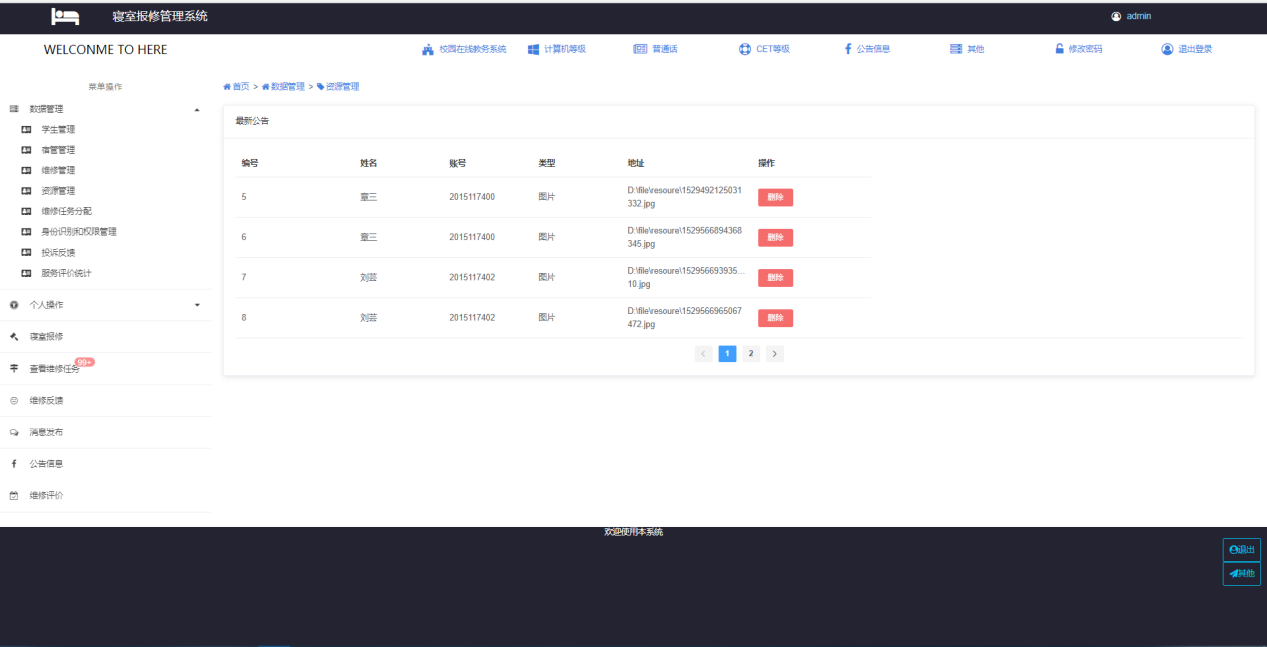
维修管理3-3-5图



维修编辑3-3-6图

3.3.4资源管理

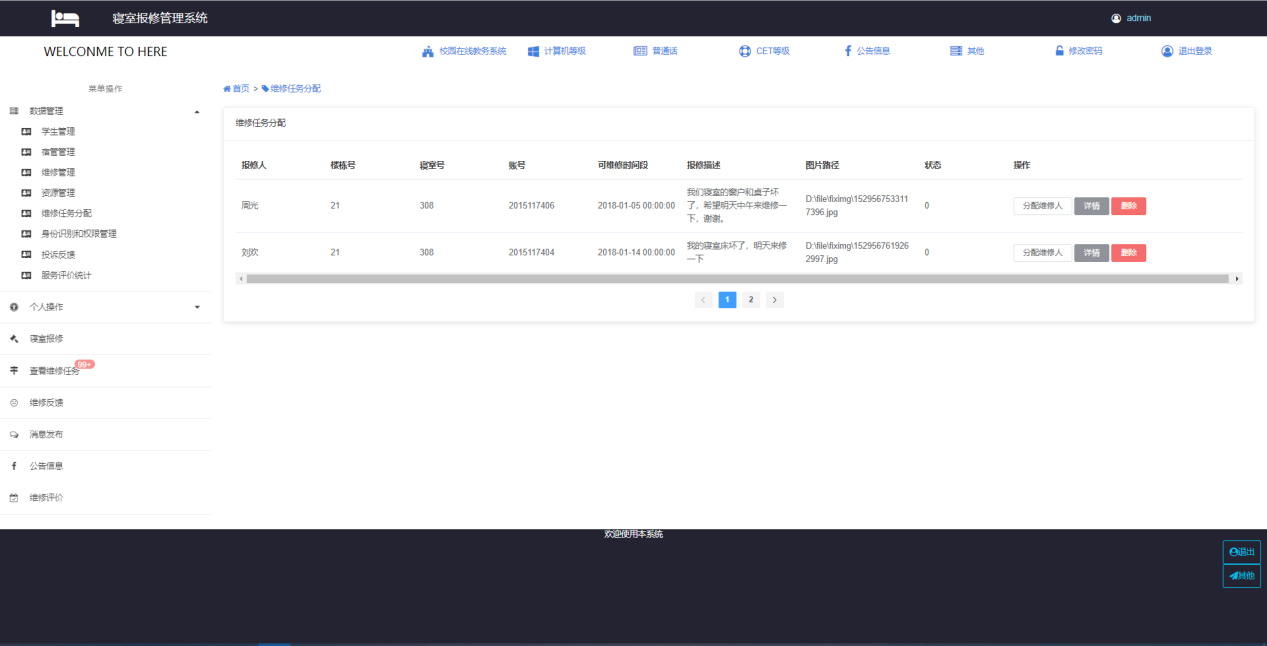
管理员通过该模块对学生分享的寝室资源进行管理，将比较老的数据删除，维护数据。如图3-3-7所示。



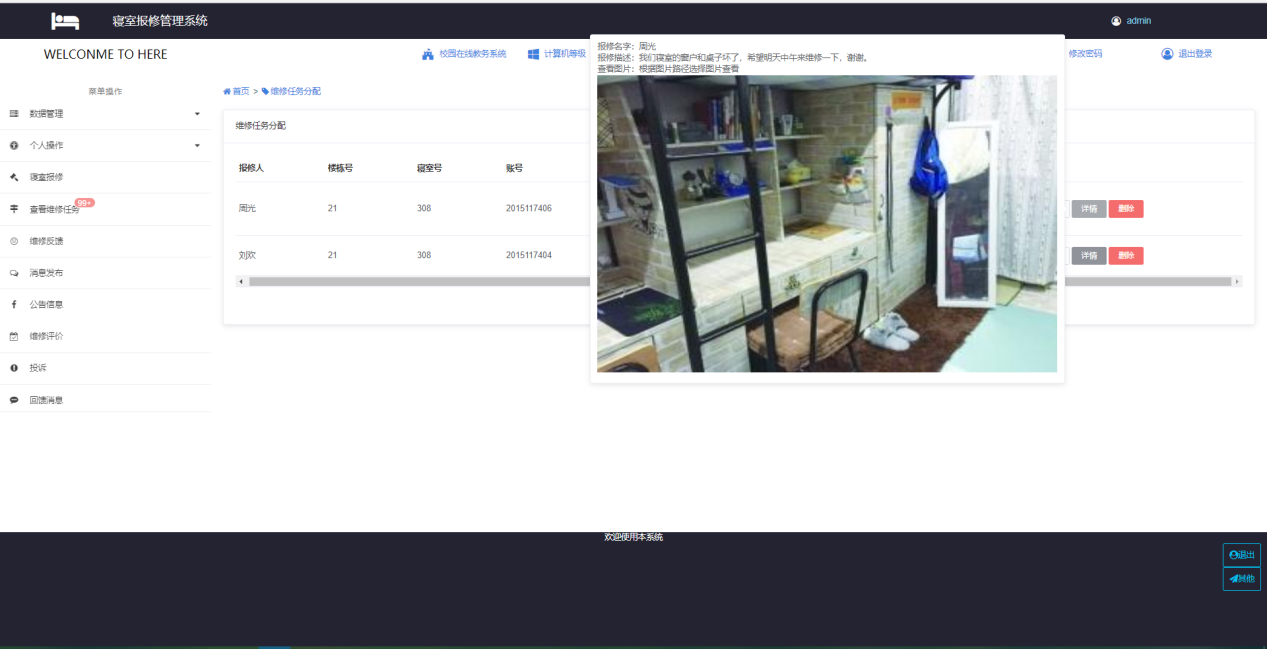
资源管理3-3-7图

3.3.5维修任务分配

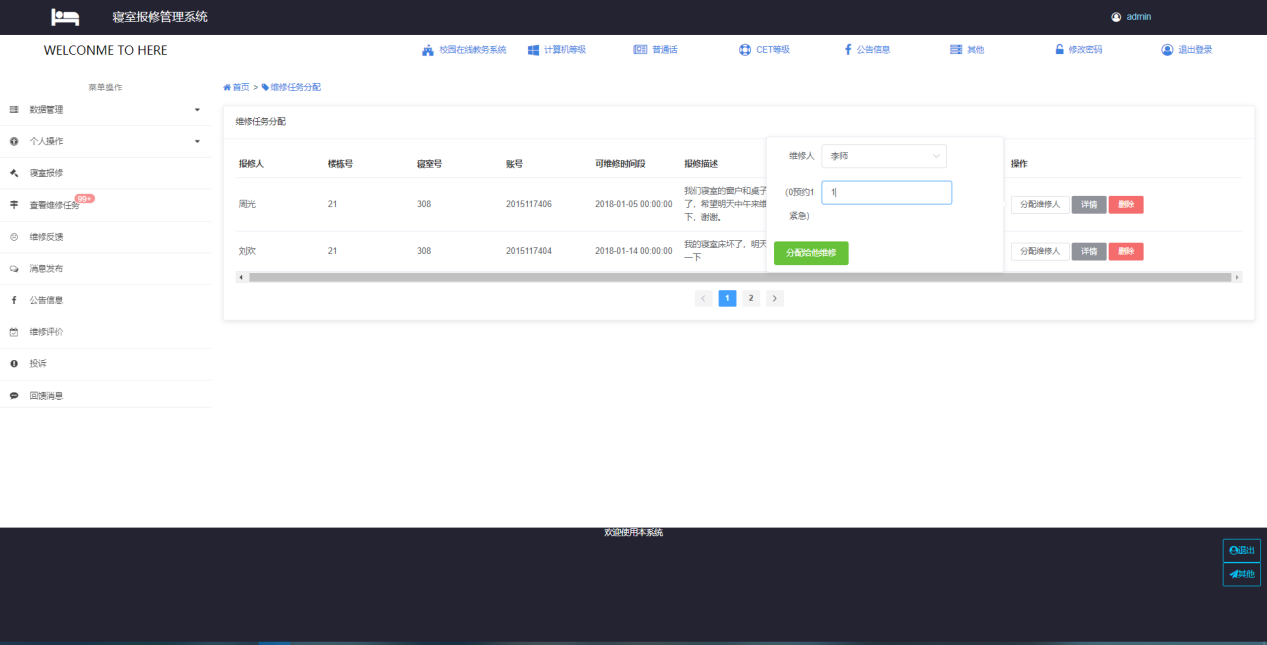
当学生提出寝室报修请求之后，管理员将给学生寝室分配维修人员对其寝室进行维修服务，这个模块就是给管理员提供维修任务分配的，并且由管理员控制维修模式（0预约，1紧急），分配成功之后将自动删除这一条寝室报修请求。如图3-3-8、3-3-9、3-3-10所示。



维修任务分配3-3-8图



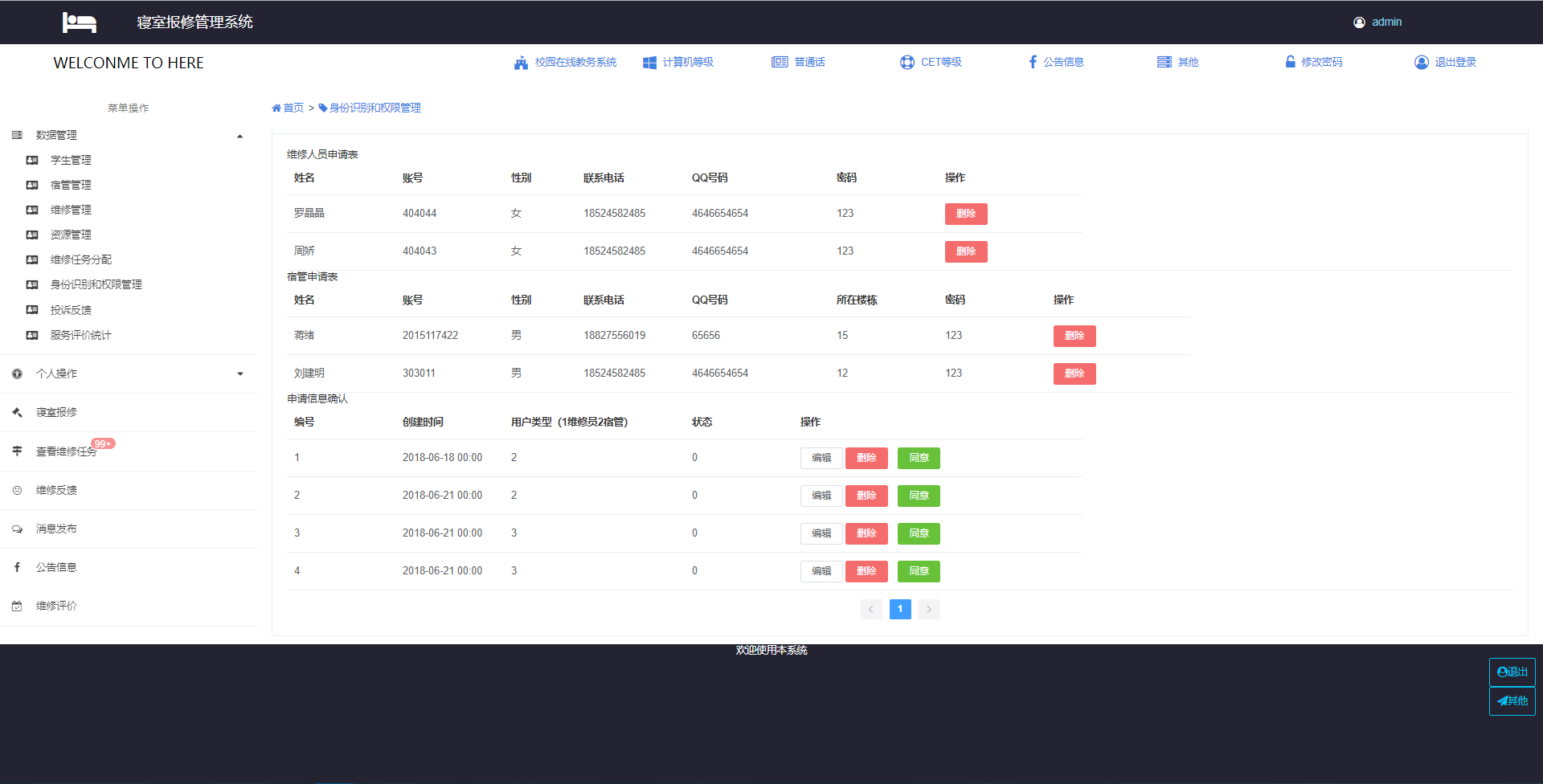
维修任务详情3-3-9图



维修分配3-3-10图

3.3.6身份识别和权限管理

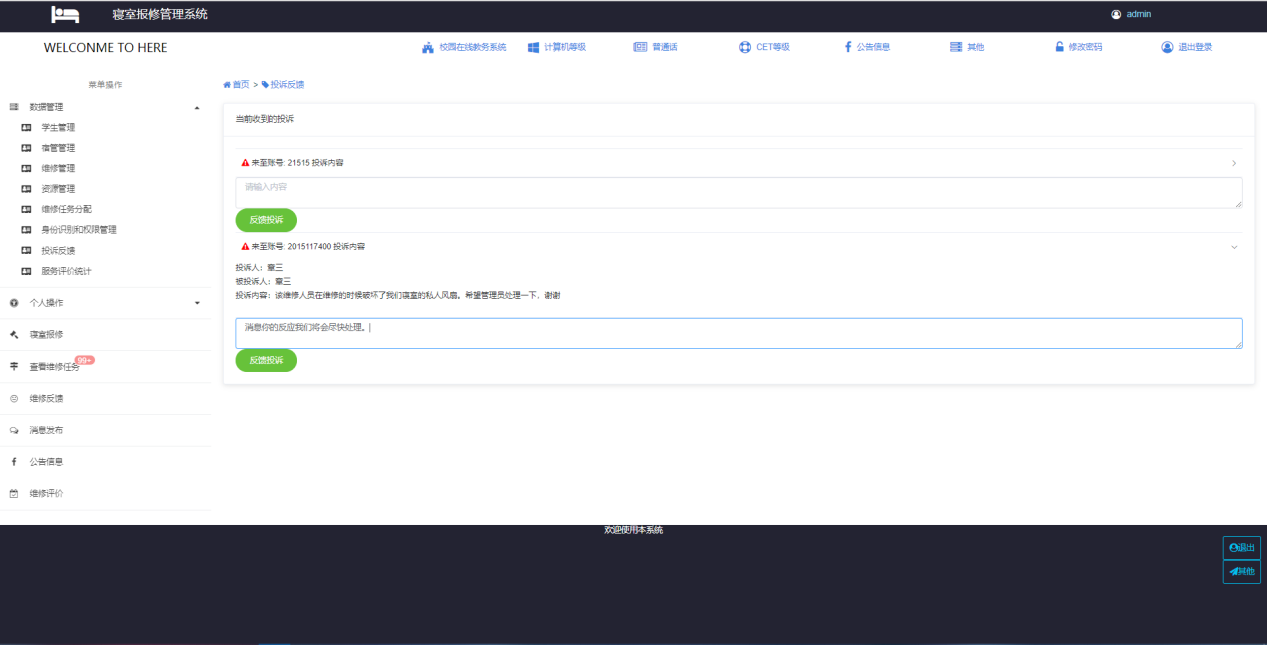
当由宿管或者维修人员注册账户时，将注册的账户数据提交到管理员处，等待管理员确认该人员是否属于宿管或者维修人员，避免“不良分子”产生，所以管理员可以通过该模块对宿管和维修人员有一个申请确认，并且可以对其进行编辑、删除等操作。如图3-3-11所示。



身份识别和权限管理3-3-11图

3.3.7投诉反馈

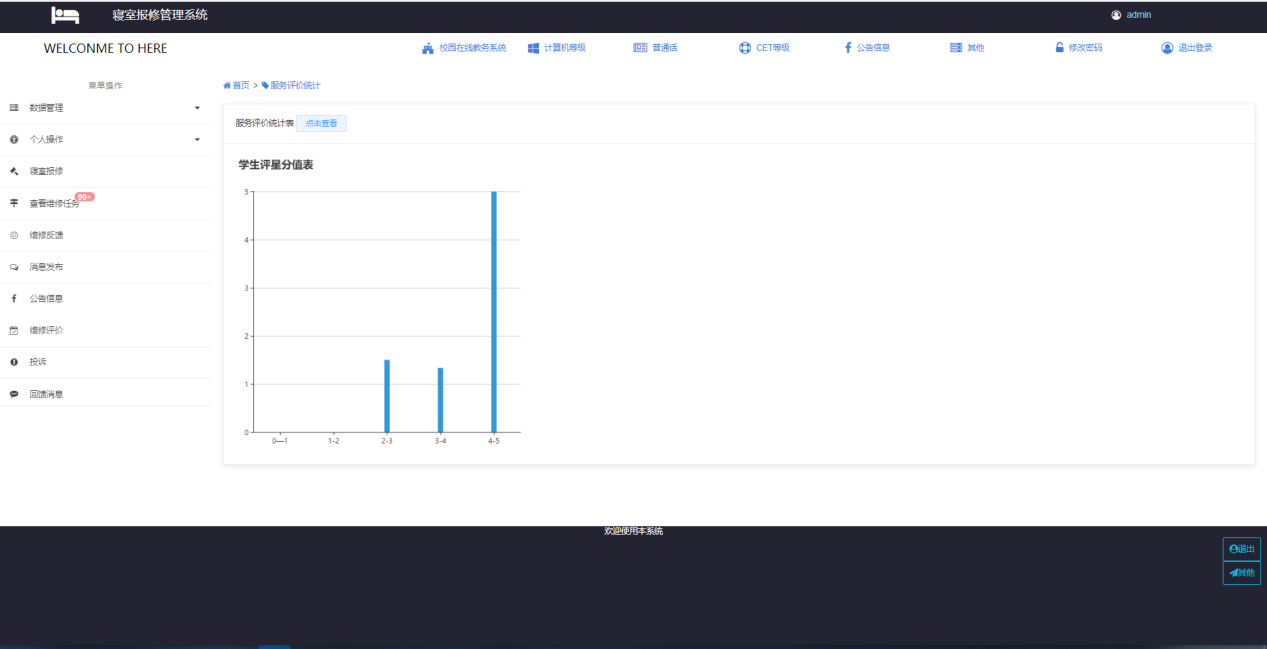
该模块主要提高回复学生的投诉信息。如图3-3-12所示。



投诉反馈3-3-12图

3.3.8服务评价统计

管理员可以通过该模块直观的查看最近的评分情况。如图3-3-13所示。



服务评价统计3-3-13图

# 4. 后台源代码

## 4.1 Controller层

4.1.1 LonginController.java类

package com.jx.controller;

import static org.hamcrest.CoreMatchers.nullValue;

import javax.security.auth.message.callback.PrivateKeyCallback.Request;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.HousemanPojo;

import com.jx.pojo.StudentPojo;

import com.jx.pojo.SupermanPojo;

import com.jx.pojo.UpdatmanPojo;

import com.jx.service.HousemanService;

import com.jx.service.StudentService;

import com.jx.service.SupermanService;

import com.jx.service.UpdatmanService;

@Controller

public class LonginController {

@Autowired

private StudentService studentService;

@Autowired

private HousemanService housemanService;

@Autowired

private UpdatmanService updatmanService;

@Autowired

private SupermanService supermanService;

@RequestMapping("/login")

@ResponseBody

public Object login(HttpServletRequest req) {

String number = req.getParameter("number");

String pwd = req.getParameter("password");

String flag = req.getParameter("flag");

Integer flag1;

if(number == null ||number.equals("")||pwd == null || pwd.equals("")) return null;

try {

flag1 = Integer.valueOf(flag);

} catch (Exception e) {

return null;

}

if (flag1 == 1) {

//学生

StudentPojo studentPojo=studentService.findById(number);

if (studentPojo == null) return null;

if (studentPojo.getStudentpassword().equals(pwd)) {

req.getSession().setAttribute("number", number);

req.getSession().setAttribute("flag", flag);

return studentPojo;

}

return null;

}else if (flag1 == 2) {

//宿管

HousemanPojo housemanPojo = housemanService.findById(number);

req.getSession().setAttribute("number", number);

req.getSession().setAttribute("flag", flag);

if(housemanPojo.getHousemanpassword().equals(pwd)) {

return housemanPojo;

}

return null;

}else if (flag1 == 3) {

//维修员

UpdatmanPojo updatmanPojo = updatmanService.findById(number);

if (updatmanPojo.getUpdatmanpassword().equals(pwd)) {

req.getSession().setAttribute("number", number);

req.getSession().setAttribute("flag", flag);

return updatmanPojo;

}

return null;

}else if (flag1 == 4) {

//管理员

SupermanPojo supermanPojo = supermanService.findById(number);

if (supermanPojo.getSuperpwd().equals(pwd)) {

req.getSession().setAttribute("number", number);

req.getSession().setAttribute("flag", flag);

return supermanPojo;

}else {

return null;

}

}else {

return null;

}

}

@RequestMapping("/loginout")

@ResponseBody

public boolean loginout(HttpServletRequest req) {

if(req.getSession().getAttribute("number")!=null) {

req.removeAttribute("number");

req.removeAttribute("flag");

return true;

}

return false;

}

}

4.1.2 RegisteController.java类

package com.jx.controller;

import java.util.Date;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.ApplyPojo;

import com.jx.pojo.HousemanPojo;

import com.jx.pojo.StudentPojo;

import com.jx.pojo.UpdatmanPojo;

import com.jx.service.ApplyService;

import com.jx.service.StudentService;

import com.jx.util.JsonUtils;

@Controller

public class RegisteController {

@Autowired

private StudentService studentService;

@Autowired

private ApplyService applyService;

@RequestMapping("/registre")

@ResponseBody

public boolean registre(HttpServletRequest req) {

String flag = req.getParameter("flag");

Integer flag1;

try {

flag1 = Integer.valueOf(flag);

} catch (Exception e) {

return false;

}

if(flag1==1) {

//学生

String studentnum = req.getParameter("studentnum");

String studentname = req.getParameter("studentname");

String studenttel = req.getParameter("studenttel");

String studentsex = req.getParameter("studentsex");

String studentcode = req.getParameter("studentcode");

String studenthouse = req.getParameter("studenthouse");

String studentclass = req.getParameter("studentclass");

String studentqq = req.getParameter("studentqq");

String studentpassword = req.getParameter("studentpassword");

Integer studentcode1,studenthouse1;

System.out.println(studentnum+studenttel+"::"+studenthouse+",,"+studentpassword);

try {

studentcode1 = Integer.valueOf(studentcode);

studenthouse1 = Integer.valueOf(studenthouse);

} catch (Exception e) {

return false;

}

StudentPojo studentPojo = new StudentPojo(studentnum, studentname, studenttel, studentsex,

studentcode1, studenthouse1, studentclass, studentqq, studentpassword);

boolean i=studentService.insertStudent(studentPojo);

if (!i) return false;

return true;

}else if (flag1==2) {

//宿管

String housemannum = req.getParameter("housemannum");

String housemanname = req.getParameter("housemanname");

String housemantel = req.getParameter("housemantel");

String housemansex = req.getParameter("housemansex");

String housemancode = req.getParameter("housemancode");

String housemanqq = req.getParameter("housemanqq");

String housemanpassword = req.getParameter("housemanpassword");

Integer housemancode1;

try {

housemancode1 = Integer.valueOf(housemancode);

} catch (Exception e) {

return false;

}

HousemanPojo housemanPojo = new HousemanPojo(housemannum, housemanname, housemantel,

housemansex, housemancode1, housemanqq, housemanpassword);

String desc = JsonUtils.objectToJson(housemanPojo);

ApplyPojo applyPojo = new ApplyPojo(null, desc, new Date(), flag1,0);

boolean i = applyService.insertApply(applyPojo);

if (!i) return false;

return true;

}else if (flag1==3) {

//维修

String updatmannum = req.getParameter("updatmannum");

String updatmanname = req.getParameter("updatmanname");

String updatmantel = req.getParameter("updatmantel");

String updatmansex = req.getParameter("updatmansex");

String updatmanqq = req.getParameter("updatmanqq");

String updatmanpassword = req.getParameter("updatmanpassword");

UpdatmanPojo updatmanPojo = new UpdatmanPojo(updatmannum, updatmanname, updatmantel,

updatmansex, updatmanqq, updatmanpassword);

String desc = JsonUtils.objectToJson(updatmanPojo);

ApplyPojo applyPojo = new ApplyPojo(null, desc, new Date(), flag1,0);

boolean i = applyService.insertApply(applyPojo);

if (!i) return false;

return true;

}else {

return false;

}

}

}

4.1.3 ApplyController.java类

package com.jx.controller;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.ApplyPojo;

import com.jx.pojo.HousemanPojo;

import com.jx.pojo.UpdatmanPojo;

import com.jx.service.ApplyService;

import com.jx.service.HousemanService;

import com.jx.service.UpdatmanService;

import com.jx.util.DateUtil;

import com.jx.util.JsonUtils;

@Controller

public class ApplyController {

@Autowired

private ApplyService applyService;

@Autowired

private HousemanService housemanService;

@Autowired

private UpdatmanService updatmanService;

/\*\*

\* 根据id查询申请信息

\* @param req

\* @return

\*/

@RequestMapping("/findbyid")

@ResponseBody

public ApplyPojo findByid(HttpServletRequest req) {

String applyid = req.getParameter("applyid");

Long applyid1;

try {

applyid1 = Long.valueOf(applyid);

} catch (Exception e) {

return null;

}

return applyService.findById(applyid1);

}

/\*\*

\* 查询所有申请信息

\* @param req

\* @return

\*/

@RequestMapping("/findall")

@ResponseBody

public List<ApplyPojo> findAll(HttpServletRequest req) {

return applyService.findAll();

}

/\*\*

\* 根据id删除申请信息

\* @param req

\* @return

\*/

@RequestMapping("/deletebyid")

@ResponseBody

public boolean deleteByid(HttpServletRequest req) {

String applyid = req.getParameter("applyid");

Long applyid1;

try {

applyid1 = Long.valueOf(applyid);

} catch (Exception e) {

return false;

}

return applyService.deleteApply(applyid1);

}

/\*\*

\* 修改申请信息

\* @param req

\* @return

\*/

@RequestMapping("/updateApply")

@ResponseBody

public boolean updateApply(HttpServletRequest req) {

String applyid = req.getParameter("applyid");

String applydesc = req.getParameter("applydesc");

String createtime = req.getParameter("createtime");

String flag = req.getParameter("flag");

String statu = req.getParameter("statu");

Long id = null;

Integer flag1=null,statu1=null;

Date time=null;

try {

id = Long.valueOf(applyid);

flag1 = Integer.valueOf(flag);

statu1 = Integer.valueOf(statu);

time = DateUtil.formatDate(createtime, "yyyy-mm-dd hh:mm:ss");

} catch (Exception e) {

return false;

}

ApplyPojo applyPojo = new ApplyPojo(id, applydesc, time, flag1, statu1);

return applyService.updateApply(applyPojo);

}

/\*\*

\* 更具申请状态查询申请信息

\* @param req

\* @return

\*/

@RequestMapping("/findBystatu")

@ResponseBody

public List<ApplyPojo> findBystatu(HttpServletRequest req){

String statu = req.getParameter("statu");

Integer statu1;

try {

statu1 = Integer.valueOf(statu);

} catch (Exception e) {

return null;

}

return applyService.findByStatu(statu1);

}

/\*\*

\* 申请同意

\* @param req

\* @return

\*/

@RequestMapping("/agree")

@ResponseBody

public boolean agreeApply(HttpServletRequest req) {

String applyid = req.getParameter("applyid");

Long applyid1;

try {

applyid1 = Long.valueOf(applyid);

} catch (Exception e) {

return false;

}

ApplyPojo applyPojo = applyService.findById(applyid1);

Integer flag = applyPojo.getFlag();

if(flag == 2) {

//宿管

try {

HousemanPojo housemanPojo = JsonUtils.jsonToPojo(applyPojo.getApplydesc(), HousemanPojo.class);

housemanService.insertHouseman(housemanPojo);

applyPojo.setFlag(1);

applyService.updateApply(applyPojo);

} catch (Exception e) {

return false;

}

return true;

}else if (flag==3) {

//维修

try {

UpdatmanPojo updatmanPojo = JsonUtils.jsonToPojo(applyPojo.getApplydesc(), UpdatmanPojo.class);

updatmanService.insertUpdatman(updatmanPojo);

applyPojo.setFlag(1);

applyService.updateApply(applyPojo);

} catch (Exception e) {

return false;

}

return true;

}

return false;

}

}

4.1.4 AssessController.java类

package com.jx.controller;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.request;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

//import org.apache.catalina.tribes.group.GroupChannel.HeartbeatThread;

import org.omg.CosNaming.NamingContextExtPackage.StringNameHelper;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.AssessPojo;

import com.jx.service.AssessService;

@Controller

public class AssessController{

@Autowired

private AssessService assessService;

/\*\*

\* 查询所有评价

\* @return

\*/

@RequestMapping("/findassessall")

@ResponseBody

public List<AssessPojo> findAssessAll() {

return assessService.findAssessAll();

}

/\*\*

\* 添加评价

\* @param req

\* @return

\*/

@RequestMapping("/insertassess")

@ResponseBody

public boolean insertAsssess(HttpServletRequest req) {

String studentnum = req.getParameter("studentnum");

String studentname = req.getParameter("studentname");

String assessdesc = req.getParameter("assessdesc");

String assesscode = req.getParameter("assesscode");

String taskid = req.getParameter("taskid");

Long taskid1;

Integer assesscode1;

try {

assesscode1 = Integer.valueOf(assesscode);

taskid1 = Long.valueOf(taskid);

} catch (Exception e) {

return false;

}

AssessPojo assessPojo = new AssessPojo(null, studentnum, studentname, assessdesc, new Date(), assesscode1, taskid1);

return assessService.insertAsssess(assessPojo);

}

/\*\*

\* 根据id删除评价

\* @param req

\* @return

\*/

@RequestMapping("/deleteassess")

@ResponseBody

public boolean deleteAssess(HttpServletRequest req) {

String assessid = req.getParameter("assessid");

Long assessid1;

try {

assessid1 = Long.valueOf(assessid);

} catch (Exception e) {

return false;

}

return assessService.deleteAssess(assessid1);

}

/\*\*

\* 根据评价星级查询评价

\* @param req

\* @return

\*/

@RequestMapping("/findbycode")

@ResponseBody

public List<AssessPojo> findByAsCode(HttpServletRequest req) {

String assesscode = req.getParameter("assesscode");

Integer assesscode1;

try {

assesscode1 = Integer.valueOf(assesscode);

} catch (Exception e) {

return null;

}

return assessService.findByAsCode(assesscode1);

}

/\*\*

\* 根据任务id查询评价

\* @param req

\* @return

\*/

@RequestMapping("/findByastaskid")

@ResponseBody

public List<AssessPojo> findByTaskid(HttpServletRequest req){

String taskid = req.getParameter("taskid");

Long taskid1;

try {

taskid1 = Long.valueOf(taskid);

} catch (Exception e) {

return null;

}

return assessService.findByTaskid(taskid1);

}

}

4.1.5 CompalainController.java类

package com.jx.controller;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.ComplainPojo;

import com.jx.service.ComplainService;

@Controller

public class CompalainController{

@Autowired

private ComplainService complainService;

/\*\*

\* 查询所有投诉信息

\* @return

\*/

@RequestMapping("/finfcomall")

@ResponseBody

public List<ComplainPojo> findComall() {

return complainService.findComall();

}

/\*\*

\* 增加投诉信息

\* @param req

\* @return

\*/

@RequestMapping("/insertcom")

@ResponseBody

public boolean insertCom(HttpServletRequest req) {

String studentnum = req.getParameter("studentnum");

String number = req.getParameter("number");

String username = req.getParameter("studentname");

String complaindesc = req.getParameter("complaindesc");

String studentname = req.getParameter("studentname");

ComplainPojo cpojo = new ComplainPojo(null, studentnum, number, username, studentname, complaindesc, new Date());

return complainService.insertCom(cpojo);

}

/\*\*

\* 添加投诉信息

\* @param req

\* @return

\*/

@RequestMapping("/deletecom")

@ResponseBody

public boolean deleteCom(HttpServletRequest req) {

String complaindescid = req.getParameter("complaindescid");

Long complaindescid1;

try {

complaindescid1 = Long.valueOf(complaindescid);

} catch (Exception e) {

return false;

}

return complainService.deleteCom(complaindescid1);

}

}

4.1.6 DeleteController.java类

package com.jx.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.service.HousemanService;

import com.jx.service.StudentService;

import com.jx.service.UpdatmanService;

@Controller

public class DeleteController {

@Autowired

private StudentService studentService;

@Autowired

private HousemanService housemanService;

@Autowired

private UpdatmanService updatmanService;

@RequestMapping("/delete")

@ResponseBody

public boolean delete(HttpServletRequest req) {

String flag = req.getParameter("flag");

Integer flag1;

try {

flag1 = Integer.valueOf(flag);

} catch (Exception e) {

return false;

}

if(flag1==1) {

//学生

String studentnum = req.getParameter("studentnum");

if (studentnum ==null ) return false;

boolean i=studentService.deleteStudent(studentnum);

if (!i) return false;

return true;

}else if (flag1==2) {

//宿管员

String housemannum = req.getParameter("housemannum");

if (housemannum ==null ) return false;

boolean i = housemanService.deleteHouseman(housemannum);

if (!i) return false;

return true;

}else if (flag1==3) {

//维修员

String updatmannum = req.getParameter("updatmannum");

if (updatmannum ==null ) return false;

boolean i = updatmanService.deleteUpdatman(updatmannum);

if (!i) return false;

return true;

}else {

return false;

}

}

}

4.1.7 FeedbackController.java类

package com.jx.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.FeedbackPojo;

import com.jx.service.FeedbackService;

@Controller

public class FeedbackController {

@Autowired

private FeedbackService feedbackService;

/\*\*

\* 添加反馈信息

\* @param req

\* @return

\*/

@RequestMapping("/insertfeedback")

@ResponseBody

public boolean insertFeedback(HttpServletRequest req) {

String complaindescid = req.getParameter("complaindescid");

String feedbackdesc = req.getParameter("feedbackdesc");

String number = req.getParameter("number");

String username = req.getParameter("username");

Long complaindescid1;

try {

complaindescid1 = Long.valueOf(complaindescid);

} catch (Exception e) {

return false;

}

FeedbackPojo fPojo = new FeedbackPojo(null, complaindescid1, feedbackdesc, number, username);

return feedbackService.insertFeedback(fPojo);

}

/\*\*

\* 根据投诉id查询投诉反馈

\* @param req

\* @return

\*/

@RequestMapping("/findcomplainId")

@ResponseBody

public List<FeedbackPojo> findcomplainId(HttpServletRequest req){

String complaindescid = req.getParameter("complaindescid");

return feedbackService.findcomplainId(complaindescid);

}

}

4.1.8 FindUserContoller.java类

package com.jx.controller;

import static org.hamcrest.CoreMatchers.nullValue;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.service.HousemanService;

import com.jx.service.StudentService;

import com.jx.service.SupermanService;

import com.jx.service.UpdatmanService;

@Controller

public class FindUserContoller {

@Autowired

private HousemanService housemanService;

@Autowired

private StudentService studentService;

@Autowired

private SupermanService supermanService;

@Autowired

private UpdatmanService updatmanService;

/\*\*

\* 根据用户账号查询用户信息

\* @param req

\* @return

\*/

@RequestMapping("/findbyuserid")

@ResponseBody

public Object findByUserid(HttpServletRequest req) {

String flag = req.getParameter("flag");

Integer flag1;

try {

flag1 = Integer.valueOf(flag);

} catch (Exception e) {

return null;

}

if (flag1==1) {

//学生

String studentnum = req.getParameter("studentnum");

return studentService.findById(studentnum);

}else if (flag1==2) {

//宿管

String housemannum = req.getParameter("housemannum");

return housemanService.findById(housemannum);

}else if (flag1==3) {

//维修员

String updatmannum = req.getParameter("updatmannum");

return updatmanService.findById(updatmannum);

}else {

return null;

}

}

/\*\*

\* 根据用户姓名查询用户信息

\* @param req

\* @return

\*/

@RequestMapping("/findbyusername")

@ResponseBody

public Object findByUserName(HttpServletRequest req) {

String flag = req.getParameter("flag");

Integer flag1;

try {

flag1 = Integer.valueOf(flag);

} catch (Exception e) {

return null;

}

if (flag1==1) {

//学生

String studentname = req.getParameter("studentname");

return studentService.findByStudentName(studentname);

}else if (flag1==2) {

//宿管

String housemanname = req.getParameter("housemanname");

return housemanService.findByUserName(housemanname);

}else if (flag1==3) {

//维修员

String updatmanname = req.getParameter("updatmanname");

return updatmanService.findByUpdateName(updatmanname);

}else {

return null;

}

}

/\*\*

\* 查询所有用户信息

\* @param req

\* @return

\*/

@RequestMapping("/findbyuserall")

@ResponseBody

public Object findByUserAll(HttpServletRequest req) {

String flag = req.getParameter("flag");

Integer flag1;

try {

flag1 = Integer.valueOf(flag);

} catch (Exception e) {

return null;

}

if (flag1==1) {

//学生

return studentService.findSudentAll();

}else if (flag1==2) {

//宿管

return housemanService.findUserAll();

}else if (flag1==3) {

//维修员

return updatmanService.findUpdateAll();

}else {

return null;

}

}

}

4.1.9 FixController.java类

package com.jx.controller;

import java.text.ParseException;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.jx.pojo.FixPojo;

import com.jx.service.FixService;

import com.jx.util.DateUtil;

@Controller

public class FixController{

@Autowired

private FixService fixService;

/\*\*

\* 添加维修任务

\* @param fixPojo

\* @return

\* @throws ParseException

\*/

@RequestMapping("/insertfix")

@ResponseBody

public boolean insertFix(HttpServletRequest req,@RequestParam("file") CommonsMultipartFile file) throws ParseException {

String studentnum = req.getParameter("studentnum");

String fixdesc = req.getParameter("fixdesc");

String fixtime = req.getParameter("fixtime");

String studentcode = req.getParameter("studentcode");

String studenthouse = req.getParameter("studenthouse");

String studentname = req.getParameter("studentname");

String fixflag = req.getParameter("fixflag");

Integer studentcode1=null,studenthouse1=null,fixflag1=null;

Date fixtime1=null;

try {

studentcode1 = Integer.valueOf(studentcode);

studenthouse1 = Integer.valueOf(studenthouse);

fixflag1 = Integer.valueOf(fixflag);

fixtime1 = DateUtil.formatDate(fixtime, "yyyy-mm-dd hh:mm:mm");

} catch (Exception e) {

return false;

}

FixPojo fixPojo = new FixPojo(null, studentnum, fixdesc, fixtime1, null, studentcode1, studenthouse1, studentname, 0);

return fixService.insertFix(fixPojo, file);

}

/\*\*

\* 查看历史维修任务

\* @return

\*/

@RequestMapping("/findallfix")

@ResponseBody

public List<FixPojo> findFixall(HttpServletRequest req) {

return fixService.findFixall();

}

/\*\*

\* 根据处理状态查看维修任务

\* @param fixflag

\* @return

\*/

@RequestMapping("/findfixbystaut")

@ResponseBody

public List<FixPojo> findFixBystaut(HttpServletRequest req) {

return fixService.findFixBystaut(0);

}

/\*\*

\* 删除维修任务

\* @param fixid

\* @return

\*/

@RequestMapping("/deletefix")

@ResponseBody

public boolean deleteFix(HttpServletRequest req) {

String fixid = req.getParameter("fixid");

Long fixid1;

try {

fixid1 = Long.valueOf(fixid);

} catch (Exception e) {

return false;

}

return fixService.deleteFix(fixid1);

}

/\*\*

\* 改变维修任务状态

\* @param fixid

\* @param fixflag

\* @return

\*/

@RequestMapping("/updatefix")

@ResponseBody

public boolean updateStaut(HttpServletRequest req) {

String fixid = req.getParameter("fixid");

String fixflag = req.getParameter("fixflag");

Long fixid1;

Integer fixflag1;

try {

fixid1 = Long.valueOf(fixid);

fixflag1 = Integer.valueOf(fixflag);

} catch (Exception e) {

return false;

}

return fixService.updateStaut(fixid1, fixflag1);

}

}

4.1.10 NoticeController.Java类

package com.jx.controller;

import static org.hamcrest.CoreMatchers.nullValue;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.Noticepojo;

import com.jx.service.NoticeService;

@Controller

public class NoticeController {

@Autowired

private NoticeService noticeService;

/\*\*

\* 发布通知

\* @param req

\* @return

\*/

@RequestMapping("/releasenotice")

@ResponseBody

public boolean releaseNotice(HttpServletRequest req) {

String housemannum = req.getParameter("housemannum");

String housemanname = req.getParameter("housemanname");

String noticedesc = req.getParameter("noticedesc");

String noticehose = req.getParameter("noticehose");

Integer noticehose1;

try {

noticehose1 = Integer.valueOf(noticehose);

} catch (Exception e) {

return false;

}

Noticepojo noticepojo = new Noticepojo(null, housemannum, housemanname, noticedesc, new Date(), noticehose1);

return noticeService.insertNotice(noticepojo);

}

/\*\*

\* 删除通知

\* @param req

\* @return

\*/

@RequestMapping("/deletenotice")

@ResponseBody

public boolean deleteNotice(HttpServletRequest req) {

String noticid = req.getParameter("noticid");

Long noticid1;

try {

noticid1 = Long.valueOf(noticid);

} catch (Exception e) {

return false;

}

return noticeService.deleteNotice(noticid1);

}

/\*\*

\* 查询最新通知

\* @param req

\* @return

\*/

@RequestMapping("/findonenotice")

@ResponseBody

public Noticepojo findOneNotice(HttpServletRequest req) {

String noticehose = req.getParameter("noticehose");

Integer noticehose1;

try {

noticehose1 = Integer.valueOf(noticehose);

} catch (Exception e) {

return null;

}

return noticeService.findOneNotice(noticehose1);

}

/\*\*

\* 查询所有通知

\* @param req

\* @return

\*/

@RequestMapping("/findallnotice")

@ResponseBody

public List<Noticepojo> findAllNotice(HttpServletRequest req){

return noticeService.findAllNotice();

}

}

4.1.11 ResourceController.java类

package com.jx.controller;

import java.io.File;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.jx.pojo.SharePojo;

import com.jx.service.ShareService;

import com.jx.util.FileUtil;

@Controller

public class ResourceController {

@Autowired

private ShareService shareService;

/\*\*

\* 资源上传功能

\* @param file

\* @param req

\* @return

\*/

@RequestMapping("/uploadfile")

@ResponseBody

public boolean upResource( @RequestParam("file") CommonsMultipartFile file,HttpServletRequest req) {

String studentnum = req.getParameter("studentnum");

String studentname = req.getParameter("studentname");

String sharetype = req.getParameter("sharetype");

System.out.println(studentnum+studentname+sharetype);

SharePojo sharePojo = new SharePojo(null, studentnum, studentname, null, sharetype);

return shareService.insertShare(sharePojo, file);

}

/\*\*

\* 查看所有资源

\* @param req

\* @return

\*/

@RequestMapping("/seefile")

@ResponseBody

public List<SharePojo> seeResource(HttpServletRequest req) {

return shareService.findShareAll();

}

/\*\*

\* 删除资源

\* @param request

\* @return

\*/

@RequestMapping("/deletefile")

@ResponseBody

public boolean deleteFile(HttpServletRequest request) {

String shareid = request.getParameter("shareid");

Long shareid1;

try {

shareid1 = Long.valueOf(shareid);

} catch (Exception e) {

return false;

}

return shareService.deleteShare(shareid1);

}

/\*\*

\* 文件下载

\* @param req

\* @return

\*/

@RequestMapping("/downfile")

@ResponseBody

public boolean downFile(HttpServletRequest req,HttpServletResponse response) {

String sharepath = req.getParameter("sharepath");

String filename = req.getParameter("filename");

return FileUtil.fileDown(response, sharepath, filename);

}

}

4.1.12 TaskController.java类

package com.jx.controller;

import java.util.Date;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.cglib.reflect.FastClass;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.TaskPojo;

import com.jx.service.TaskService;

import com.jx.util.DateUtil;

@Controller

public class TaskController {

@Autowired

private TaskService taskService;

/\*\*

\* 添加维修任务

\* @param req

\* @return

\*/

@RequestMapping("/addtask")

@ResponseBody

public boolean insertTask(HttpServletRequest req) {

String taskdesc = req.getParameter("taskdesc");

String taskman = req.getParameter("taskman");

String taskaddress = req.getParameter("taskaddress");

String tasktime = req.getParameter("tasktime");

String taskfirst = req.getParameter("taskfirst");

Integer taskfirst1;

Date tasktime1;

try {

taskfirst1 = Integer.valueOf(taskfirst);

tasktime1 = DateUtil.formatDate(tasktime, "yyyy-mm-dd hh:mm:mm");

} catch (Exception e) {

return false;

}

TaskPojo tPojo = new TaskPojo(null, taskdesc, taskman, taskaddress, tasktime1, new Date(), 0, taskfirst1);

return taskService.insertTask(tPojo);

}

/\*\*

\* 查询所有维修任务

\* @param req

\* @return

\*/

@RequestMapping("/findtaskall")

@ResponseBody

public List<TaskPojo> findTaskall(HttpServletRequest req) {

return taskService.findTaskall();

}

/\*\*

\* 根据维修任务id删除维修任务

\* @param req

\* @return

\*/

@RequestMapping("/deletetask")

@ResponseBody

public boolean deleteTask(HttpServletRequest req) {

String taskid = req.getParameter("taskid");

Long taskid1;

try {

taskid1 = Long.valueOf(taskid);

} catch (Exception e) {

return false;

}

return taskService.deleteTask(taskid1);

}

/\*\*

\* 根据维修任务状态查询维修任务

\* @param req

\* @return

\*/

@RequestMapping("/findtaskbystaut")

@ResponseBody

public List<TaskPojo> findTaskBystaut(HttpServletRequest req) {

String taskflag = req.getParameter("taskflag");

Integer taskflag1;

try {

taskflag1 = Integer.valueOf(taskflag);

} catch (Exception e) {

return null;

}

return taskService.findTaskBystaut(taskflag1);

}

/\*\*

\* 修改维修任务状态

\* @param req

\* @return

\*/

@RequestMapping("/updatetaskstaut")

@ResponseBody

public boolean updateTaskStaut(HttpServletRequest req) {

String taskflag = req.getParameter("taskflag");

String taskid = req.getParameter("taskid");

Long taskid1;

Integer taskflag1;

try {

taskid1 = Long.valueOf(taskid);

taskflag1 = Integer.valueOf(taskflag);

} catch (Exception e) {

return false;

}

return taskService.updateTaskStaut(taskid1, taskflag1);

}

}

4.1.13 UpdateController.java类

package com.jx.controller;

import java.util.Date;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import com.jx.pojo.HousemanPojo;

import com.jx.pojo.StudentPojo;

import com.jx.pojo.UpdatmanPojo;

import com.jx.service.HousemanService;

import com.jx.service.StudentService;

import com.jx.service.UpdatmanService;

@Controller

public class UpdateController {

@Autowired

private StudentService studentService;

@Autowired

private UpdatmanService updatmanService;

@Autowired

private HousemanService housemanService;

@RequestMapping("/update")

@ResponseBody

public boolean registre(HttpServletRequest req) {

String flag = req.getParameter("flag");

Integer flag1;

try {

flag1 = Integer.valueOf(flag);

} catch (Exception e) {

return false;

}

if(flag1==1) {

//学生

String studentnum = req.getParameter("studentnum");

String studentname = req.getParameter("studentname");

String studenttel = req.getParameter("studenttel");

String studentsex = req.getParameter("studentsex");

String studentcode = req.getParameter("studentcode");

String studenthouse = req.getParameter("studenthouse");

String studentclass = req.getParameter("studentclass");

String studentqq = req.getParameter("studentqq");

String studentpassword = req.getParameter("studentpassword");

System.out.println(studentname+studenttel+"::"+studenthouse+",,"+studentpassword);

Integer studentcode1,studenthouse1;

try {

studentcode1 = Integer.valueOf(studentcode);

studenthouse1 = Integer.valueOf(studenthouse);

System.out.println(studenthouse1);

} catch (Exception e) {

return false;

}

StudentPojo studentPojo = new StudentPojo(studentnum, studentname, studenttel, studentsex,

studentcode1, studenthouse1, studentclass, studentqq, studentpassword);

boolean i=studentService.updateStudent(studentPojo);

if (!i) return false;

return true;

}else if (flag1==2) {

//宿管

String housemannum = req.getParameter("housemannum");

String housemanname = req.getParameter("housemanname");

String housemantel = req.getParameter("housemantel");

String housemansex = req.getParameter("housemansex");

String housemancode = req.getParameter("housemancode");

String housemanqq = req.getParameter("housemanqq");

String housemanpassword = req.getParameter("housemanpassword");

Integer housemancode1;

try {

housemancode1 = Integer.valueOf(housemancode);

} catch (Exception e) {

return false;

}

HousemanPojo housemanPojo = new HousemanPojo(housemannum, housemanname, housemantel,

housemansex, housemancode1, housemanqq, housemanpassword);

boolean i = housemanService.updateHouseman(housemanPojo);

if (!i) return false;

return true;

}else if (flag1==3) {

//维修

String updatmannum = req.getParameter("updatmannum");

String updatmanname = req.getParameter("updatmanname");

String updatmantel = req.getParameter("updatmantel");

String updatmansex = req.getParameter("updatmansex");

String updatmanqq = req.getParameter("updatmanqq");

String updatmanpassword = req.getParameter("updatmanpassword");

UpdatmanPojo updatmanPojo = new UpdatmanPojo(updatmannum, updatmanname, updatmantel,

updatmansex, updatmanqq, updatmanpassword);

boolean i = updatmanService.updateUpdatman(updatmanPojo);

if (!i) return false;

return true;

}else {

return false;

}

}

}

## 4.2 Dao层

4.2.1 ApplyMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.ApplyPojo;

public interface ApplyMapper {

public ApplyPojo findById(Long applyid);

public List<ApplyPojo> findAll();

public void updateApply(@Param("applyPojo")ApplyPojo applyPojo);

public void insertApply(@Param("applyPojo")ApplyPojo applyPojo);

public void deleteApply(Long applyid);

public List<ApplyPojo> findByStatu(Integer statu);

}

4.2.2 AssessMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.AssessPojo;

public interface AssessMapper {

public List<AssessPojo> findAssessAll();

public void insertAsssess(@Param("asp")AssessPojo asp);

public void deleteAssess(Long assessid);

public List<AssessPojo> findByAsCode(Integer assesscode);

public List<AssessPojo> findByTaskid(Long taskid);

}

4.2.3 ComplainMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.ComplainPojo;

public interface ComplainMapper {

public List<ComplainPojo> findComall();

public void insertCom(@Param("cpojo")ComplainPojo cpojo);

public void deleteCom(Long complaindescid);

}

4.2.4 FeedbackMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.FeedbackPojo;

public interface FeedbackMapper {

public void insertFeedback(@Param("fPojo")FeedbackPojo fPojo);

public List<FeedbackPojo> findcomplainId(String complaindescid);

}

4.2.5 FixMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.FixPojo;

public interface FixMapper {

public void insertFix(@Param("fixPojo")FixPojo fixPojo);

public List<FixPojo> findFixall();

public List<FixPojo> findFixBystaut(Integer fixflag);

public void deleteFix(Long fixid);

public void updateStaut(@Param("fixid")Long fixid,@Param("fixflag")Integer fixflag);

}

4.2.6 HousemanMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.HousemanPojo;

public interface HousemanMapper {

public HousemanPojo findById(String housemannum);

public void insertHouseman(@Param("houseman")HousemanPojo houseman);

public void updateHouseman(@Param("houseman")HousemanPojo houseman);

public void deleteHouseman(String housemannum);

public List<HousemanPojo> findByUserName(String housemanname);

public List<HousemanPojo> findUserAll();

}

4.2.7 NoticeMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.Noticepojo;

public interface NoticeMapper {

public void insertNotice(@Param("notice")Noticepojo notice);

public void deleteNotice(Long noticid);

public Noticepojo findOneNotice(Integer noticehose);

public List<Noticepojo> findAllNotice();

}、

4.2.8 ShareMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.SharePojo;

public interface ShareMapper {

public List<SharePojo> findShareAll();

public void insertShare(@Param("sp")SharePojo sp);

public void deleteShare(Long shareid);

}

4.2.9 StudentMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.StudentPojo;

public interface StudentMapper {

public StudentPojo findById(String studentnum);

public void insertStudent(@Param("stu")StudentPojo stu);

public void updateStudent(@Param("stu")StudentPojo stu);

public void deleteStudent(String studentnum);

public List<StudentPojo> findByStudentName(String studentname);

public List<StudentPojo> findSudentAll();

}

4.2.10 SupermanMapper.java类

package com.jx.dao;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.StudentPojo;

import com.jx.pojo.SupermanPojo;

public interface SupermanMapper {

public SupermanPojo findById(String supernum);

public void insertSuperman(@Param("stp")SupermanPojo stp);

public void updateSuperman(@Param("stp")SupermanPojo stp);

public void deleteSuperman(String supernum);

}

4.2.11 TaskMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.TaskPojo;

public interface TaskMapper {

public void insertTask(@Param("taskPojo")TaskPojo taskPojo);

public List<TaskPojo> findTaskall();

public void deleteTask(Long taskid);

public List<TaskPojo> findTaskBystaut(Integer taskflag);

public void updateTaskStaut(@Param("taskid")Long taskid,@Param("taskflag")Integer taskflag);

}

4.2.12 UpdatmanMapper.java类

package com.jx.dao;

import java.util.List;

import org.apache.ibatis.annotations.Param;

import com.jx.pojo.StudentPojo;

import com.jx.pojo.UpdatmanPojo;

public interface UpdatmanMapper {

public UpdatmanPojo findById(String updatmannum);

public void insertUpdatman(@Param("upm")UpdatmanPojo upm);

public void updateUpdatman(@Param("upm")UpdatmanPojo upm);

public void deleteUpdatman(String updatmannum);

public List<UpdatmanPojo> findByUpdateName(String updatmanname);

public List<UpdatmanPojo> findUpdateAll();

}

## 4.3 Mapping层

4.3.1 ApplyMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.ApplyMapper">

<!-- 1.添加申请记录信息 -->

<insert id="insertApply" parameterType="com.jx.pojo.ApplyPojo">

insert into tbapply(applyid,applydesc,createtime,flag,statu) values

(#{applyPojo.applyid},#{applyPojo.applydesc},#{applyPojo.createtime},#{applyPojo.flag},#{applyPojo.statu})

</insert>

<!-- 2。修改申请记录信息 -->

<update id="updateApply" parameterType="com.jx.pojo.ApplyPojo">

update tbapply set

applyid = #{applyPojo.applyid},

applydesc = #{applyPojo.applydesc},

createtime = #{applyPojo.createtime},

flag = #{applyPojo.flag},

statu = #{applyPojo.statu}

where applyid = #{applyPojo.applyid}

</update>

<!-- 3.删除申请记录信息 -->

<delete id="deleteApply" parameterType="Long">

delete from tbapply where applyid = #{applyPojo.applyid}

</delete>

<!-- 4.根据id查询申请信息 -->

<select id="findById" parameterType="Long" resultType="com.jx.pojo.ApplyPojo">

select \* from tbapply where applyid = #{applyPojo.applyid}

</select>

<!-- 5.查询所有申请信息 -->

<select id="findAll" parameterType="Long" resultType="com.jx.pojo.ApplyPojo">

select \* from tbapply

</select>

<!-- 6.查询未处理的申请信息 -->

<select id="findByStatu" resultType="com.jx.pojo.ApplyPojo">

select \* from tbapply where statu = #{applyPojo.statu}

</select>

</mapper>

4.3.2 AssessMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.AssessMapper">

<!-- 1：查询所有评价 -->

<select id="findAssessAll" resultType="com.jx.pojo.AssessPojo">

select \* from assess order by assessid desc

</select>

<!-- 2.添加报修任务 -->

<insert id="insertAsssess" parameterType="com.jx.pojo.AssessPojo">

insert into assess(studentnum,studentname,assessdesc,assesstime,assesscode,taskid) values

(#{asp.studentnum},#{asp.studentname},#{asp.assessdesc},#{asp.assesstime},#{asp.assesscode},#{asp.taskid})

</insert>

<!-- 3.删除报修任务 -->

<delete id="deleteAssess" >

delete from assess where assessid = #{assessid}

</delete>

<!-- 4.根据评价星级查询评价 -->

<select id="findByAsCode" resultType="com.jx.pojo.AssessPojo">

select \* from assess where assesscode=#{assesscode}

</select>

<!-- 5.根据任务id查询评价 -->

<select id="findByTaskid" resultType="com.jx.pojo.AssessPojo">

select \* from assess where taskid=#{taskid} order by assessid desc

</select>

</mapper>

4.3.3 ComplainMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.ComplainMapper">

<!-- 1：查询所有投诉 -->

<select id="findComall" resultType="com.jx.pojo.ComplainPojo">

select \* from complain

</select>

<!-- 2.添加投诉信息 -->

<insert id="insertCom" parameterType="com.jx.pojo.ComplainPojo">

insert into complain(studentnum,number,username,studentname,complaindesc,commdate) values

(#{cpojo.studentnum},#{cpojo.number},#{cpojo.username},#{cpojo.studentname},#{cpojo.complaindesc},#{cpojo.commdate})

</insert>

<!-- 3.删除投诉 -->

<delete id="deleteCom" >

delete from complain where complaindescid = #{complaindescid}

</delete>

</mapper>

4.3.4 FeedbackMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.FeedbackMapper">

<!--添加投诉反馈 -->

<insert id="insertFeedback" parameterType="com.jx.pojo.FeedbackPojo">

insert into feedback(complaindescid,feedbackdesc,number,username)

values (#{fPojo.complaindescid},#{fPojo.feedbackdesc},#{fPojo.number},#{fPojo.username})

</insert>

<!-- 根据投诉id查询投诉反馈 -->

<select id="findcomplainId" resultType="com.jx.pojo.FeedbackPojo">

select \* from feedback where number=#{complaindescid}

</select>

</mapper>

4.3.5 FixMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.FixMapper">

<!-- 1：查询所有报修任务 -->

<select id="findFixall" resultType="com.jx.pojo.FixPojo">

select \* from fix

</select>

<!-- 2.添加报修任务 -->

<insert id="insertFix" parameterType="com.jx.pojo.FixPojo">

insert into fix(studentnum,fixdesc,fixtime,fiximage,studentcode,studenthouse,

studentname,fixflag) values

(#{fixPojo.studentnum},#{fixPojo.fixdesc},#{fixPojo.fixtime},#{fixPojo.fiximage},

#{fixPojo.studentcode},#{fixPojo.studenthouse},#{fixPojo.studentname},#{fixPojo.fixflag})

</insert>

<!-- 3.删除报修任务 -->

<delete id="deleteFix" >

delete from fix where fixid = #{fixid}

</delete>

<!-- 4：根据状态查询报修任务 -->

<select id="findFixBystaut" resultType="com.jx.pojo.FixPojo">

select \* from fix where fixflag=#{fixflag}

</select>

<!-- 4:改变报修任务状态 -->

<update id="updateStaut">

update fix set fixflag=#{fixflag} where fixid = #{fixid}

</update>

</mapper>

4.3.6 HousemanMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.HousemanMapper">

<!-- 1：根据宿管员编号查询 -->

<select id="findById" parameterType="String" resultType="com.jx.pojo.HousemanPojo">

select \* from houseman where housemannum=#{housemannum}

</select>

<!-- 2.添加宿管员信息 -->

<insert id="insertHouseman" parameterType="com.jx.pojo.HousemanPojo">

insert into houseman(housemannum,housemanname,housemantel,housemansex,

housemancode,housemanqq,housemanpassword) values

(#{houseman.housemannum},#{houseman.housemanname},#{houseman.housemantel},

#{houseman.housemansex},#{houseman.housemancode},#{houseman.housemanqq},

#{houseman.housemanpassword})

</insert>

<!-- 3。修改宿管员信息 -->

<update id="updateHouseman" parameterType="com.jx.pojo.HousemanPojo">

update houseman set

housemannum = #{houseman.housemannum},housemanname = #{houseman.housemanname}, housemantel = #{houseman.housemantel},

housemansex = #{houseman.housemansex},housemancode = #{houseman.housemancode},housemanqq = #{houseman.housemanqq},

housemanpassword = #{houseman.housemanpassword}

where housemannum = #{houseman.housemannum}

</update>

<!-- 4.删除宿管员信息 -->

<delete id="deleteHouseman" parameterType="String">

delete from houseman where housemannum = #{housemannum}

</delete>

<!-- 5：根据宿管员姓名查询 -->

<select id="findByUserName" parameterType="String" resultType="com.jx.pojo.HousemanPojo">

select \* from houseman where housemanname = #{housemanname}

</select>

<!-- 6：查询所有宿管员信息 -->

<select id="findUserAll" resultType="com.jx.pojo.HousemanPojo">

select \* from houseman

</select>

</mapper>

4.3.7 NoticeMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.NoticeMapper">

<!-- 1.添加新通知 -->

<insert id="insertNotice" parameterType="com.jx.pojo.Noticepojo">

insert into notice(housemannum,housemanname,noticedesc,noticetime,noticehose) values

(#{notice.housemannum},#{notice.housemanname},#{notice.noticedesc},#{notice.noticetime},#{notice.noticehose})

</insert>

<!-- 2.删除历史通知 -->

<delete id="deleteNotice" >

delete from notice where noticid = #{noticid}

</delete>

<!-- 4：查询最新通知 -->

<select id="findFixBystaut" resultType="com.jx.pojo.Noticepojo">

select \* from notice where noticehose=#{noticehose} order by noticid desc limit 0,1

</select>

<!-- 5:查询所有历史通知-->

<select id="findAllNotice" resultType="com.jx.pojo.Noticepojo">

select \* from notice

</select>

</mapper>

4.3.8 ShareMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.ShareMapper">

<!-- 1：查询所有资源 -->

<select id="findShareAll" resultType="com.jx.pojo.SharePojo">

select \* from share

</select>

<!-- 2.添加资源 -->

<insert id="insertShare" parameterType="com.jx.pojo.SharePojo">

insert into share(studentnum,studentname,sharepath,sharetype) values

(#{sp.studentnum},#{sp.studentname},#{sp.sharepath},#{sp.sharetype})

</insert>

<!-- 3.删除资源 -->

<delete id="deleteShare" >

delete from share where shareid = #{shareid}

</delete>

</mapper>

4.3.9 StudentMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.StudentMapper">

<!-- 1：根据学生学号查询 -->

<select id="findById" parameterType="String" resultType="com.jx.pojo.StudentPojo">

select \* from student where studentnum=#{studentnum}

</select>

<!-- 2.添加学生信息 -->

<insert id="insertStudent" parameterType="com.jx.pojo.StudentPojo">

insert into student(studentnum,studentname,studenttel,studentsex,studentcode,

studenthouse,studentclass,studentqq,studentpassword) values

(#{stu.studentnum},#{stu.studentname},#{stu.studenttel},#{stu.studentsex},#{stu.studentcode},

#{stu.studenthouse},#{stu.studentclass},#{stu.studentqq},#{stu.studentpassword})

</insert>

<!-- 3。修改学生信息 -->

<update id="updateStudent" parameterType="com.jx.pojo.StudentPojo">

update student set

studentnum = #{stu.studentnum},studentname = #{stu.studentname}, studenttel = #{stu.studenttel},

studentsex = #{stu.studentsex},studentcode = #{stu.studentcode},studenthouse = #{stu.studenthouse},

studentclass = #{stu.studentclass},studentqq = #{stu.studentqq},studentpassword = #{stu.studentpassword}

where studentnum = #{stu.studentnum}

</update>

<!-- 4.删除学生信息 -->

<delete id="deleteStudent" parameterType="String">

delete from student where studentnum = #{studentnum}

</delete>

<!-- 5：根据学生姓名查询 -->

<select id="findByStudentName" parameterType="String" resultType="com.jx.pojo.StudentPojo">

select \* from student where studentname=#{studentname}

</select>

<!-- 6：查询所有学生信息 -->

<select id="findSudentAll" resultType="com.jx.pojo.StudentPojo">

select \* from student

</select>

</mapper>

4.3.10 SupermanMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.SupermanMapper">

<!-- 1：根据学生学号查询 -->

<select id="findById" parameterType="String" resultType="com.jx.pojo.SupermanPojo">

select \* from superman where supernum=#{supernum}

</select>

<!-- 2.添加学生信息 -->

<insert id="insertStudent" parameterType="com.jx.pojo.SupermanPojo">

insert into superman(supernum,superpwd) values

(#{stp.supernum},#{stp.superpwd})

</insert>

<!-- 3。修改学生信息 -->

<update id="updateStudent" parameterType="com.jx.pojo.SupermanPojo">

update superman set

supernum = #{stp.supernum},

superpwd = #{stp.supernum}

where supernum = #{stp.supernum}

</update>

<!-- 4.删除学生信息 -->

<delete id="deleteStudent" parameterType="String">

delete from superman where supernum=#{supernum}

</delete>

</mapper>

4.3.11 TaskMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.TaskMapper">

<!-- 1：查询所有维修任务 -->

<select id="findTaskall" resultType="com.jx.pojo.TaskPojo">

select \* from task

</select>

<!-- 2.添加维修任务 -->

<insert id="insertTask" parameterType="com.jx.pojo.TaskPojo">

insert into task(taskdesc,taskman,taskaddress,tasktime,taskcreatetime,taskflag,taskfirst) values

(#{taskPojo.taskdesc},#{taskPojo.taskman},#{taskPojo.taskaddress},#{taskPojo.tasktime},

#{taskPojo.taskcreatetime},#{taskPojo.taskflag},#{taskPojo.taskfirst})

</insert>

<!-- 3.删除维修任务 -->

<delete id="deleteTask" >

delete from task taskid = #{taskid}

</delete>

<!-- 4：根据状态查询维修任务 -->

<select id="findTaskBystaut" resultType="com.jx.pojo.TaskPojo">

select \* from task where taskflag=#{taskflag}

</select>

<!-- 5:改变维修任务状态 -->

<update id="updateTaskStaut">

update task set taskid=#{taskid} where taskflag = #{taskflag}

</update>

</mapper>

4.3.12 UpdatmanMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jx.dao.UpdatmanMapper">

<!-- 1：根据维修员编号查询 -->

<select id="findById" parameterType="String" resultType="com.jx.pojo.UpdatmanPojo">

select \* from updatman where updatmannum=#{updatmannum}

</select>

<!-- 2.添加维修员信息 -->

<insert id="insertUpdatman" parameterType="com.jx.pojo.UpdatmanPojo">

insert into updatman(updatmannum,updatmanname,updatmantel,updatmansex,

updatmanqq,updatmanpassword) values

(#{upm.updatmannum},#{upm.updatmanname},#{upm.updatmantel},#{upm.updatmansex},

#{upm.updatmanqq},#{upm.updatmanpassword})

</insert>

<!-- 3。修改维修员信息 -->

<update id="updateUpdatman" parameterType="com.jx.pojo.UpdatmanPojo">

update updatman set

updatmannum = #{upm.updatmannum},

updatmanname = #{upm.updatmanname},

updatmantel = #{upm.updatmantel},

updatmansex = #{upm.updatmansex},

updatmanqq = #{upm.updatmanqq},

updatmanpassword = #{upm.updatmanpassword}

where updatmannum = #{upm.updatmannum}

</update>

<!-- 4.删除维修员信息 -->

<delete id="deleteUpdatman" parameterType="String">

delete from updatman where updatmannum = #{updatmannum}

</delete>

<!-- 5：根据维修员姓名查询 -->

<select id="findByUpdateName" parameterType="String" resultType="com.jx.pojo.UpdatmanPojo">

select \* from updatman where updatmanname=#{updatmanname}

</select>

<!-- 6：查询维所有维修人员信息-->

<select id="findUpdateAll" resultType="com.jx.pojo.UpdatmanPojo">

select \* from updatman

</select>

</mapper>

## 4.4 pojo层

4.4.1 ApplyPojo.java类

package com.jx.pojo;

import java.util.Date;

public class ApplyPojo {

private Long applyid;

private String applydesc;

private Date createtime;

private Integer flag;

private Integer statu;

public Long getApplyid() {

return applyid;

}

public void setApplyid(Long applyid) {

this.applyid = applyid;

}

public String getApplydesc() {

return applydesc;

}

public void setApplydesc(String applydesc) {

this.applydesc = applydesc;

}

public Date getCreatetime() {

return createtime;

}

public void setCreatetime(Date createtime) {

this.createtime = createtime;

}

public Integer getFlag() {

return flag;

}

public void setFlag(Integer flag) {

this.flag = flag;

}

public Integer getStatu() {

return statu;

}

public void setStatu(Integer statu) {

this.statu = statu;

}

@Override

public String toString() {

return "ApplyPojo [applyid=" + applyid + ", applydesc=" + applydesc + ", createtime=" + createtime + ", flag="

+ flag + ", statu=" + statu + "]";

}

public ApplyPojo(Long applyid, String applydesc, Date createtime, Integer flag, Integer statu) {

super();

this.applyid = applyid;

this.applydesc = applydesc;

this.createtime = createtime;

this.flag = flag;

this.statu = statu;

}

public ApplyPojo() {

super();

}

}

4.4.2 AssessPojo.java类

package com.jx.pojo;

import java.util.Date;

public class AssessPojo {

private Long assessid;

private String studentnum;

private String studentname;

private String assessdesc;

private Date assesstime;

private Integer assesscode;

private Long taskid;

public Long getAssessid() {

return assessid;

}

public void setAssessid(Long assessid) {

this.assessid = assessid;

}

public String getStudentnum() {

return studentnum;

}

public void setStudentnum(String studentnum) {

this.studentnum = studentnum;

}

public String getStudentname() {

return studentname;

}

public void setStudentname(String studentname) {

this.studentname = studentname;

}

public String getAssessdesc() {

return assessdesc;

}

public void setAssessdesc(String assessdesc) {

this.assessdesc = assessdesc;

}

public Date getAssesstime() {

return assesstime;

}

public void setAssesstime(Date assesstime) {

this.assesstime = assesstime;

}

public Integer getAssesscode() {

return assesscode;

}

public void setAssesscode(Integer assesscode) {

this.assesscode = assesscode;

}

public Long getTaskid() {

return taskid;

}

public void setTaskid(Long taskid) {

this.taskid = taskid;

}

@Override

public String toString() {

return "AssessPojo [assessid=" + assessid + ", studentnum=" + studentnum + ", studentname=" + studentname

+ ", assessdesc=" + assessdesc + ", assesstime=" + assesstime + ", assesscode=" + assesscode

+ ", taskid=" + taskid + "]";

}

public AssessPojo(Long assessid, String studentnum, String studentname, String assessdesc, Date assesstime,

Integer assesscode, Long taskid) {

super();

this.assessid = assessid;

this.studentnum = studentnum;

this.studentname = studentname;

this.assessdesc = assessdesc;

this.assesstime = assesstime;

this.assesscode = assesscode;

this.taskid = taskid;

}

public AssessPojo() {

super();

// TODO Auto-generated constructor stub

}

}

4.4.3 ComplainPojo.java类

package com.jx.pojo;

import java.util.Date;

public class ComplainPojo {

private Long complaindescid;

private String studentnum;

private String number;

private String username;

private String studentname;

private String complaindesc;

private Date commdate;

public Long getComplaindescid() {

return complaindescid;

}

public void setComplaindescid(Long complaindescid) {

this.complaindescid = complaindescid;

}

public String getStudentnum() {

return studentnum;

}

public void setStudentnum(String studentnum) {

this.studentnum = studentnum;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getStudentname() {

return studentname;

}

public void setStudentname(String studentname) {

this.studentname = studentname;

}

public String getComplaindesc() {

return complaindesc;

}

public void setComplaindesc(String complaindesc) {

this.complaindesc = complaindesc;

}

public Date getCommdate() {

return commdate;

}

public void setCommdate(Date commdate) {

this.commdate = commdate;

}

public ComplainPojo(Long complaindescid, String studentnum, String number, String username, String studentname,

String complaindesc, Date commdate) {

super();

this.complaindescid = complaindescid;

this.studentnum = studentnum;

this.number = number;

this.username = username;

this.studentname = studentname;

this.complaindesc = complaindesc;

this.commdate = commdate;

}

public ComplainPojo() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "ComplainPojo [complaindescid=" + complaindescid + ", studentnum=" + studentnum + ", number=" + number

+ ", username=" + username + ", studentname=" + studentname + ", complaindesc=" + complaindesc

+ ", commdate=" + commdate + "]";

}

}

4.4.4 FeedbackPojo.java类

package com.jx.pojo;

public class FeedbackPojo {

public Long feedbackid;

public Long complaindescid;

public String feedbackdesc;

public String number;

public String username;

public Long getFeedbackid() {

return feedbackid;

}

public void setFeedbackid(Long feedbackid) {

this.feedbackid = feedbackid;

}

public Long getComplaindescid() {

return complaindescid;

}

public void setComplaindescid(Long complaindescid) {

this.complaindescid = complaindescid;

}

public String getFeedbackdesc() {

return feedbackdesc;

}

public void setFeedbackdesc(String feedbackdesc) {

this.feedbackdesc = feedbackdesc;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public FeedbackPojo(Long feedbackid, Long complaindescid, String feedbackdesc, String number, String username) {

super();

this.feedbackid = feedbackid;

this.complaindescid = complaindescid;

this.feedbackdesc = feedbackdesc;

this.number = number;

this.username = username;

}

public FeedbackPojo() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "FeedbackPojo [feedbackid=" + feedbackid + ", complaindescid=" + complaindescid + ", feedbackdesc="

+ feedbackdesc + ", number=" + number + ", username=" + username + "]";

}

}

4.4.5 FixPojo.java类

package com.jx.pojo;

import java.util.Date;

public class FixPojo {

private Long fixid;

private String studentnum;

private String fixdesc;

private Date fixtime;

private String fiximage;

private Integer studentcode;

private Integer studenthouse;

private String studentname;

private Integer fixflag;

public Long getFixid() {

return fixid;

}

public void setFixid(Long fixid) {

this.fixid = fixid;

}

public String getStudentnum() {

return studentnum;

}

public void setStudentnum(String studentnum) {

this.studentnum = studentnum;

}

public String getFixdesc() {

return fixdesc;

}

public void setFixdesc(String fixdesc) {

this.fixdesc = fixdesc;

}

public Date getFixtime() {

return fixtime;

}

public void setFixtime(Date fixtime) {

this.fixtime = fixtime;

}

public String getFiximage() {

return fiximage;

}

public void setFiximage(String fiximage) {

this.fiximage = fiximage;

}

public Integer getStudentcode() {

return studentcode;

}

public void setStudentcode(Integer studentcode) {

this.studentcode = studentcode;

}

public Integer getStudenthouse() {

return studenthouse;

}

public void setStudenthouse(Integer studenthouse) {

this.studenthouse = studenthouse;

}

public String getStudentname() {

return studentname;

}

public void setStudentname(String studentname) {

this.studentname = studentname;

}

public Integer getFixflag() {

return fixflag;

}

public void setFixflag(Integer fixflag) {

this.fixflag = fixflag;

}

@Override

public String toString() {

return "FixPojo [fixid=" + fixid + ", studentnum=" + studentnum + ", fixdesc=" + fixdesc + ", fixtime="

+ fixtime + ", fiximage=" + fiximage + ", studentcode=" + studentcode + ", studenthouse=" + studenthouse

+ ", studentname=" + studentname + ", fixflag=" + fixflag + "]";

}

public FixPojo(Long fixid, String studentnum, String fixdesc, Date fixtime, String fiximage, Integer studentcode,

Integer studenthouse, String studentname, Integer fixflag) {

super();

this.fixid = fixid;

this.studentnum = studentnum;

this.fixdesc = fixdesc;

this.fixtime = fixtime;

this.fiximage = fiximage;

this.studentcode = studentcode;

this.studenthouse = studenthouse;

this.studentname = studentname;

this.fixflag = fixflag;

}

public FixPojo() {

super();

// TODO Auto-generated constructor stub

}

}

4.4.6 HousemanPojo.java类

package com.jx.pojo;

public class HousemanPojo {

private String housemannum;

private String housemanname;

private String housemantel;

private String housemansex;

private Integer housemancode;

private String housemanqq;

private String housemanpassword;

public String getHousemannum() {

return housemannum;

}

public void setHousemannum(String housemannum) {

this.housemannum = housemannum;

}

public String getHousemanname() {

return housemanname;

}

public void setHousemanname(String housemanname) {

this.housemanname = housemanname;

}

public String getHousemantel() {

return housemantel;

}

public void setHousemantel(String housemantel) {

this.housemantel = housemantel;

}

public String getHousemansex() {

return housemansex;

}

public void setHousemansex(String housemansex) {

this.housemansex = housemansex;

}

public Integer getHousemancode() {

return housemancode;

}

public void setHousemancode(Integer housemancode) {

this.housemancode = housemancode;

}

public String getHousemanqq() {

return housemanqq;

}

public void setHousemanqq(String housemanqq) {

this.housemanqq = housemanqq;

}

public String getHousemanpassword() {

return housemanpassword;

}

public void setHousemanpassword(String housemanpassword) {

this.housemanpassword = housemanpassword;

}

@Override

public String toString() {

return "HousemanPojo [housemannum=" + housemannum + ", housemanname=" + housemanname + ", housemantel="

+ housemantel + ", housemansex=" + housemansex + ", housemancode=" + housemancode + ", housemanqq="

+ housemanqq + ", housemanpassword=" + housemanpassword + "]";

}

public HousemanPojo(String housemannum, String housemanname, String housemantel, String housemansex,

Integer housemancode, String housemanqq, String housemanpassword) {

super();

this.housemannum = housemannum;

this.housemanname = housemanname;

this.housemantel = housemantel;

this.housemansex = housemansex;

this.housemancode = housemancode;

this.housemanqq = housemanqq;

this.housemanpassword = housemanpassword;

}

public HousemanPojo() {

super();

// TODO Auto-generated constructor stub

}

}

4.4.7 Noticepojo.java类

package com.jx.pojo;

import java.util.Date;

public class Noticepojo {

private Long noticid;

private String housemannum;

private String housemanname;

private String noticedesc;

private Date noticetime;

private Integer noticehose;

public Long getNoticid() {

return noticid;

}

public void setNoticid(Long noticid) {

this.noticid = noticid;

}

public String getHousemannum() {

return housemannum;

}

public void setHousemannum(String housemannum) {

this.housemannum = housemannum;

}

public String getHousemanname() {

return housemanname;

}

public void setHousemanname(String housemanname) {

this.housemanname = housemanname;

}

public String getNoticedesc() {

return noticedesc;

}

public void setNoticedesc(String noticedesc) {

this.noticedesc = noticedesc;

}

public Date getNoticetime() {

return noticetime;

}

public void setNoticetime(Date noticetime) {

this.noticetime = noticetime;

}

public Integer getNoticehose() {

return noticehose;

}

public void setNoticehose(Integer noticehose) {

this.noticehose = noticehose;

}

public Noticepojo(Long noticid, String housemannum, String housemanname, String noticedesc, Date noticetime,

Integer noticehose) {

super();

this.noticid = noticid;

this.housemannum = housemannum;

this.housemanname = housemanname;

this.noticedesc = noticedesc;

this.noticetime = noticetime;

this.noticehose = noticehose;

}

public Noticepojo() {

super();

// TODO Auto-generated constructor stub

}

@Override

public String toString() {

return "Noticepojo [noticid=" + noticid + ", housemannum=" + housemannum + ", housemanname=" + housemanname

+ ", noticedesc=" + noticedesc + ", noticetime=" + noticetime + ", noticehose=" + noticehose + "]";

}

}

4.4.8 SharePojo.java类

package com.jx.pojo;

public class SharePojo {

private Long shareid;

private String studentnum;

private String studentname;

private String sharepath;

private String sharetype;

public Long getShareid() {

return shareid;

}

public void setShareid(Long shareid) {

this.shareid = shareid;

}

public String getStudentnum() {

return studentnum;

}

public void setStudentnum(String studentnum) {

this.studentnum = studentnum;

}

public String getStudentname() {

return studentname;

}

public void setStudentname(String studentname) {

this.studentname = studentname;

}

public String getSharepath() {

return sharepath;

}

public void setSharepath(String sharepath) {

this.sharepath = sharepath;

}

public String getSharetype() {

return sharetype;

}

public void setSharetype(String sharetype) {

this.sharetype = sharetype;

}

@Override

public String toString() {

return "SharePojo [shareid=" + shareid + ", studentnum=" + studentnum + ", studentname=" + studentname

+ ", sharepath=" + sharepath + ", sharetype=" + sharetype + "]";

}

public SharePojo(Long shareid, String studentnum, String studentname, String sharepath, String sharetype) {

super();

this.shareid = shareid;

this.studentnum = studentnum;

this.studentname = studentname;

this.sharepath = sharepath;

this.sharetype = sharetype;

}

public SharePojo() {

super();

// TODO Auto-generated constructor stub

}

}

4.4.9 StudentPojo.java类

package com.jx.pojo;

public class StudentPojo {

private String studentnum;

private String studentname;

private String studenttel;

private String studentsex;

private Integer studentcode;

private Integer studenthouse;

private String studentclass;

private String studentqq;

private String studentpassword;

public String getStudentnum() {

return studentnum;

}

public void setStudentnum(String studentnum) {

this.studentnum = studentnum;

}

public String getStudentname() {

return studentname;

}

public void setStudentname(String studentname) {

this.studentname = studentname;

}

public String getStudenttel() {

return studenttel;

}

public void setStudenttel(String studenttel) {

this.studenttel = studenttel;

}

public String getStudentsex() {

return studentsex;

}

public void setStudentsex(String studentsex) {

this.studentsex = studentsex;

}

public Integer getStudentcode() {

return studentcode;

}

public void setStudentcode(Integer studentcode) {

this.studentcode = studentcode;

}

public Integer getStudenthouse() {

return studenthouse;

}

public void setTudenthouse(Integer studenthouse) {

this.studenthouse = studenthouse;

}

public String getStudentclass() {

return studentclass;

}

public void setStudentclass(String studentclass) {

this.studentclass = studentclass;

}

public String getStudentqq() {

return studentqq;

}

public void setStudentqq(String studentqq) {

this.studentqq = studentqq;

}

public String getStudentpassword() {

return studentpassword;

}

public void setStudentpassword(String studentpassword) {

this.studentpassword = studentpassword;

}

@Override

public String toString() {

return "StudentPojo [studentnum=" + studentnum + ", studentname=" + studentname + ", studenttel=" + studenttel

+ ", studentsex=" + studentsex + ", studentcode=" + studentcode + ", tudenthouse=" + studenthouse

+ ", studentclass=" + studentclass + ", studentqq=" + studentqq + ", studentpassword=" + studentpassword

+ "]";

}

public StudentPojo(String studentnum, String studentname, String studenttel, String studentsex, Integer studentcode,

Integer studenthouse, String studentclass, String studentqq, String studentpassword) {

super();

this.studentnum = studentnum;

this.studentname = studentname;

this.studenttel = studenttel;

this.studentsex = studentsex;

this.studentcode = studentcode;

this.studenthouse = studenthouse;

this.studentclass = studentclass;

this.studentqq = studentqq;

this.studentpassword = studentpassword;

}

public StudentPojo() {

super();

// TODO Auto-generated constructor stub

}

}

4.4.10 SupermanPojo.java类

package com.jx.pojo;

public class SupermanPojo {

private String supernum;

private String superpwd;

public String getSupernum() {

return supernum;

}

public void setSupernum(String supernum) {

this.supernum = supernum;

}

public String getSuperpwd() {

return superpwd;

}

public void setSuperpwd(String superpwd) {

this.superpwd = superpwd;

}

@Override

public String toString() {

return "superman [supernum=" + supernum + ", superpwd=" + superpwd + "]";

}

public SupermanPojo(String supernum, String superpwd) {

super();

this.supernum = supernum;

this.superpwd = superpwd;

}

public SupermanPojo() {

super();

// TODO Auto-generated constructor stub

}

}

4.4.11 TaskPojo.java类

package com.jx.pojo;

import java.util.Date;

public class TaskPojo {

private Long taskid;

private String taskdesc;

private String taskman;

private String taskaddress;

private Date tasktime;

private Date taskcreatetime;

private Integer taskflag;

private Integer taskfirst;

public Long getTaskid() {

return taskid;

}

public void setTaskid(Long taskid) {

this.taskid = taskid;

}

public String getTaskdesc() {

return taskdesc;

}

public void setTaskdesc(String taskdesc) {

this.taskdesc = taskdesc;

}

public String getTaskman() {

return taskman;

}

public void setTaskman(String taskman) {

this.taskman = taskman;

}

public String getTaskaddress() {

return taskaddress;

}

public void setTaskaddress(String taskaddress) {

this.taskaddress = taskaddress;

}

public Date getTasktime() {

return tasktime;

}

public void setTasktime(Date tasktime) {

this.tasktime = tasktime;

}

public Date getTaskcreatetime() {

return taskcreatetime;

}

public void setTaskcreatetime(Date taskcreatetime) {

this.taskcreatetime = taskcreatetime;

}

public Integer getTaskflag() {

return taskflag;

}

public void setTaskflag(Integer taskflag) {

this.taskflag = taskflag;

}

public Integer getTaskfirst() {

return taskfirst;

}

public void setTaskfirst(Integer taskfirst) {

this.taskfirst = taskfirst;

}

@Override

public String toString() {

return "TaskPojo [taskid=" + taskid + ", taskdesc=" + taskdesc + ", taskman=" + taskman + ", taskaddress="

+ taskaddress + ", tasktime=" + tasktime + ", taskcreatetime=" + taskcreatetime + ", taskflag="

+ taskflag + ", taskfirst=" + taskfirst + "]";

}

public TaskPojo(Long taskid, String taskdesc, String taskman, String taskaddress, Date tasktime,

Date taskcreatetime, Integer taskflag, Integer taskfirst) {

super();

this.taskid = taskid;

this.taskdesc = taskdesc;

this.taskman = taskman;

this.taskaddress = taskaddress;

this.tasktime = tasktime;

this.taskcreatetime = taskcreatetime;

this.taskflag = taskflag;

this.taskfirst = taskfirst;

}

public TaskPojo() {

super();

// TODO Auto-generated constructor stub

}

}

4.4.12 UpdatmanPojo.java类

package com.jx.pojo;

public class UpdatmanPojo {

private String updatmannum;

private String updatmanname;

private String updatmantel;

private String updatmansex;

private String updatmanqq;

private String updatmanpassword;

public String getUpdatmannum() {

return updatmannum;

}

public void setUpdatmannum(String updatmannum) {

this.updatmannum = updatmannum;

}

public String getUpdatmanname() {

return updatmanname;

}

public void setUpdatmanname(String updatmanname) {

this.updatmanname = updatmanname;

}

public String getUpdatmantel() {

return updatmantel;

}

public void setUpdatmantel(String updatmantel) {

this.updatmantel = updatmantel;

}

public String getUpdatmansex() {

return updatmansex;

}

public void setUpdatmansex(String updatmansex) {

this.updatmansex = updatmansex;

}

public String getUpdatmanqq() {

return updatmanqq;

}

public void setUpdatmanqq(String updatmanqq) {

this.updatmanqq = updatmanqq;

}

public String getUpdatmanpassword() {

return updatmanpassword;

}

public void setUpdatmanpassword(String updatmanpassword) {

this.updatmanpassword = updatmanpassword;

}

@Override

public String toString() {

return "UpdatmanPojo [updatmannum=" + updatmannum + ", updatmanname=" + updatmanname + ", updatmantel="

+ updatmantel + ", updatmansex=" + updatmansex + ", updatmanqq=" + updatmanqq + ", updatmanpassword="

+ updatmanpassword + "]";

}

public UpdatmanPojo(String updatmannum, String updatmanname, String updatmantel, String updatmansex,

String updatmanqq, String updatmanpassword) {

super();

this.updatmannum = updatmannum;

this.updatmanname = updatmanname;

this.updatmantel = updatmantel;

this.updatmansex = updatmansex;

this.updatmanqq = updatmanqq;

this.updatmanpassword = updatmanpassword;

}

public UpdatmanPojo() {

super();

// TODO Auto-generated constructor stub

}

}

## 4.5 Service接口层

4.5.1 ApplyService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.ApplyPojo;

public interface ApplyService {

public ApplyPojo findById(Long applyid);

public List<ApplyPojo> findAll();

public Boolean updateApply(ApplyPojo applyPojo);

public Boolean insertApply(ApplyPojo applyPojo);

public Boolean deleteApply(Long applyid);

public List<ApplyPojo> findByStatu(Integer statu);

}

4.5.2 AssessService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.AssessPojo;

public interface AssessService {

public List<AssessPojo> findAssessAll();

public boolean insertAsssess(AssessPojo asp);

public boolean deleteAssess(Long assessid);

public List<AssessPojo> findByAsCode(Integer assesscode);

public List<AssessPojo> findByTaskid(Long taskid);

}

4.5.3 ComplainService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.ComplainPojo;

public interface ComplainService {

public List<ComplainPojo> findComall();

public boolean insertCom(ComplainPojo cpojo);

public boolean deleteCom(Long complaindescid);

}

4.5.4 FeedbackService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.FeedbackPojo;

public interface FeedbackService {

public boolean insertFeedback(FeedbackPojo fPojo);

public List<FeedbackPojo> findcomplainId(String complaindescid);

}

4.5.5 FixService .java类

package com.jx.service;

import java.util.List;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.jx.pojo.FixPojo;

public interface FixService {

public boolean insertFix(FixPojo fixPojo, CommonsMultipartFile file);

public List<FixPojo> findFixall();

public List<FixPojo> findFixBystaut(Integer fixflag);

public boolean deleteFix(Long fixid);

public boolean updateStaut(Long fixid,Integer fixflag);

}

4.5.6 HousemanService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.HousemanPojo;

public interface HousemanService {

public HousemanPojo findById(String housemannum);

public boolean insertHouseman(HousemanPojo houseman);

public boolean updateHouseman(HousemanPojo houseman);

public boolean deleteHouseman(String housemannum);

public List<HousemanPojo> findByUserName(String housemanname);

public List<HousemanPojo> findUserAll();

}

4.5.7 NoticeService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.Noticepojo;

public interface NoticeService {

public boolean insertNotice(Noticepojo notice);

public boolean deleteNotice(Long noticid);

public Noticepojo findOneNotice(Integer noticehose);

public List<Noticepojo> findAllNotice();

}

4.5.8 ShareService.java类

package com.jx.service;

import java.util.List;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.jx.pojo.SharePojo;

public interface ShareService {

public List<SharePojo> findShareAll();

public boolean insertShare(SharePojo sp,CommonsMultipartFile file);

public boolean deleteShare(Long shareid);

}

4.5.9 StudentService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.StudentPojo;

public interface StudentService {

public StudentPojo findById(String studentnum);

public Boolean insertStudent(StudentPojo stu);

public Boolean updateStudent(StudentPojo stu);

public Boolean deleteStudent(String studentnum);

public List<StudentPojo> findByStudentName(String studentname);

public List<StudentPojo> findSudentAll();

}

4.5.10 SupermanService.java类

package com.jx.service;

import com.jx.pojo.SupermanPojo;

public interface SupermanService {

public SupermanPojo findById(String supernum);

public boolean insertSuperman(SupermanPojo stp);

public boolean updateSuperman(SupermanPojo stp);

public boolean deleteSuperman(String supernum);

}

4.5.11 TaskService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.TaskPojo;

public interface TaskService {

public boolean insertTask(TaskPojo taskPojo);

public List<TaskPojo> findTaskall();

public boolean deleteTask(Long taskid);

public List<TaskPojo> findTaskBystaut(Integer taskflag);

public boolean updateTaskStaut(Long taskid,Integer taskflag);

}

4.5.12 UpdatmanService.java类

package com.jx.service;

import java.util.List;

import com.jx.pojo.UpdatmanPojo;

public interface UpdatmanService {

public UpdatmanPojo findById(String updatmannum);

public boolean insertUpdatman(UpdatmanPojo upm);

public boolean updateUpdatman(UpdatmanPojo upm);

public boolean deleteUpdatman(String updatmannum);

public List<UpdatmanPojo> findByUpdateName(String updatmanname);

public List<UpdatmanPojo> findUpdateAll();

}

## 4.6 Serviceimp实现层

4.6.1 ApplyServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.w3c.dom.ls.LSInput;

import com.jx.dao.ApplyMapper;

import com.jx.pojo.ApplyPojo;

import com.jx.service.ApplyService;

@Service

public class ApplyServiceImp implements ApplyService{

@Autowired

private ApplyMapper applyMapper;

/\*\*

\* 根据id查询申请信息

\*/

@Override

public ApplyPojo findById(Long applyid) {

ApplyPojo applyPojo;

try {

applyPojo = applyMapper.findById(applyid);

} catch (Exception e) {

return null;

}

return applyPojo;

}

/\*\*

\* 查询所有申请信息

\*/

@Override

public List<ApplyPojo> findAll() {

List<ApplyPojo> list;

try {

list = applyMapper.findAll();

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 修改申请信息

\*/

@Override

public Boolean updateApply(ApplyPojo applyPojo) {

try {

applyMapper.updateApply(applyPojo);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 插入申请信息

\*/

@Override

public Boolean insertApply(ApplyPojo applyPojo) {

try {

applyMapper.insertApply(applyPojo);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 删除申请信息

\*/

@Override

public Boolean deleteApply(Long applyid) {

try {

applyMapper.deleteApply(applyid);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据申请状态查询申请信息

\*/

@Override

public List<ApplyPojo> findByStatu(Integer statu) {

List<ApplyPojo> list;

try {

list = applyMapper.findByStatu(statu);

} catch (Exception e) {

return null;

}

return list;

}

}

4.6.2 AssessServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.AssessMapper;

import com.jx.pojo.AssessPojo;

import com.jx.service.AssessService;

@Service

public class AssessServiceImp implements AssessService{

@Autowired

private AssessMapper assessMapper;

/\*\*

\* 查询所有评价

\*/

@Override

public List<AssessPojo> findAssessAll() {

List<AssessPojo> list;

try {

list = assessMapper.findAssessAll();

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 添加评价信息

\*/

@Override

public boolean insertAsssess(AssessPojo asp) {

try {

assessMapper.insertAsssess(asp);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据id删除评价

\*/

@Override

public boolean deleteAssess(Long assessid) {

try {

assessMapper.deleteAssess(assessid);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据评价星级查询评价

\*/

@Override

public List<AssessPojo> findByAsCode(Integer assesscode) {

List<AssessPojo> list;

try {

list = findByAsCode(assesscode);

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 根据任务id查询评价

\*/

@Override

public List<AssessPojo> findByTaskid(Long taskid) {

List<AssessPojo> list;

try {

list = assessMapper.findByTaskid(taskid);

} catch (Exception e) {

return null;

}

return list;

}

}

4.6.3 ComplainServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.ComplainMapper;

import com.jx.pojo.ComplainPojo;

import com.jx.service.ComplainService;

@Service

public class ComplainServiceImp implements ComplainService{

@Autowired

private ComplainMapper compainMapper;

/\*\*

\* 查询所有投诉信息

\*/

@Override

public List<ComplainPojo> findComall() {

List<ComplainPojo> list;

try {

list = compainMapper.findComall();

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 添加投诉信息

\*/

@Override

public boolean insertCom(ComplainPojo cpojo) {

// try {

compainMapper.insertCom(cpojo);

// } catch (Exception e) {

// return false;

// }

return true;

}

/\*\*

\* 删除投诉信息

\*/

@Override

public boolean deleteCom(Long complaindescid) {

try {

compainMapper.deleteCom(complaindescid);

} catch (Exception e) {

return false;

}

return true;

}

}

4.6.4 FeedbackServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.FeedbackMapper;

import com.jx.pojo.FeedbackPojo;

import com.jx.service.FeedbackService;

@Service

public class FeedbackServiceImp implements FeedbackService{

@Autowired

private FeedbackMapper feedbackMapper;

/\*\*

\* 添加投诉反馈

\*/

@Override

public boolean insertFeedback(FeedbackPojo fPojo) {

try {

feedbackMapper.insertFeedback(fPojo);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据投诉id查询投诉反馈

\*/

@Override

public List<FeedbackPojo> findcomplainId(String complaindescid) {

List<FeedbackPojo> list;

try {

list = feedbackMapper.findcomplainId(complaindescid);

} catch (Exception e) {

return null;

}

return list;

}

}

4.6.5 FixServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.jx.dao.FixMapper;

import com.jx.pojo.FixPojo;

import com.jx.service.FixService;

import com.jx.util.FileUtil;

@Service

public class FixServiceImp implements FixService{

@Autowired

private FixMapper fixMapper;

private String Path = "D:\\file\\fiximg\\";

/\*\*

\* 添加维修任务

\*/

@Override

public boolean insertFix(FixPojo fixPojo,@RequestParam("file") CommonsMultipartFile file) {

try {

Map map = FileUtil.fileUp(file, Path);

String imgpath = (String)map.get("path")+(String)map.get("newFileName");

fixPojo.setFiximage(imgpath);

fixMapper.insertFix(fixPojo);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 查看历史维修任务

\*/

@Override

public List<FixPojo> findFixall() {

List<FixPojo> list;

try {

list = fixMapper.findFixall();

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 根据处理状态查看维修任务

\*/

@Override

public List<FixPojo> findFixBystaut(Integer fixflag) {

List<FixPojo> list;

try {

list = fixMapper.findFixBystaut(fixflag);

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 删除维修任务

\*/

@Override

public boolean deleteFix(Long fixid) {

try {

fixMapper.deleteFix(fixid);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 改变维修任务状态

\*/

@Override

public boolean updateStaut(Long fixid, Integer fixflag) {

try {

fixMapper.updateStaut(fixid, fixflag);

} catch (Exception e) {

return false;

}

return true;

}

}

4.6.6 HousemanServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.HousemanMapper;

import com.jx.pojo.HousemanPojo;

import com.jx.service.HousemanService;

@Service

public class HousemanServiceImp implements HousemanService{

@Autowired

private HousemanMapper housemanMapper;

/\*\*

\* 根据宿管员账号查找宿管员信息

\*/

@Override

public HousemanPojo findById(String housemannum) {

HousemanPojo housemanPojo;

try {

housemanPojo = housemanMapper.findById(housemannum);

} catch (Exception e) {

return null;

}

return housemanPojo;

}

/\*\*

\*添加宿管员信息

\*/

@Override

public boolean insertHouseman(HousemanPojo houseman) {

// try {

housemanMapper.insertHouseman(houseman);

// } catch (Exception e) {

// return false;

// }

return true;

}

/\*\*

\* 修改宿管员信息

\*/

@Override

public boolean updateHouseman(HousemanPojo houseman) {

try {

housemanMapper.updateHouseman(houseman);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 删除宿管员信息

\*/

@Override

public boolean deleteHouseman(String housemannum) {

try {

housemanMapper.deleteHouseman(housemannum);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据宿管员姓名查询

\*/

@Override

public List<HousemanPojo> findByUserName(String housemanname) {

List<HousemanPojo> list;

try {

list = housemanMapper.findByUserName(housemanname);

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 查询所有

\*/

@Override

public List<HousemanPojo> findUserAll() {

List<HousemanPojo> list;

try {

list = housemanMapper.findUserAll();

} catch (Exception e) {

return null;

}

return list;

}

}

4.6.7 NoticeServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.NoticeMapper;

import com.jx.pojo.Noticepojo;

import com.jx.service.NoticeService;

@Service

public class NoticeServiceImp implements NoticeService{

@Autowired

private NoticeMapper noticeMapper;

/\*\*

\* 添加新通知

\*/

@Override

public boolean insertNotice(Noticepojo notice) {

try {

noticeMapper.insertNotice(notice);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 删除通知

\*/

@Override

public boolean deleteNotice(Long noticid) {

try {

noticeMapper.deleteNotice(noticid);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 查询最新通知

\*/

@Override

public Noticepojo findOneNotice(Integer noticehose) {

Noticepojo noticepojo;

try {

noticepojo = noticeMapper.findOneNotice(noticehose);

} catch (Exception e) {

return null;

}

return noticepojo;

}

/\*\*

\* 查询所有通知

\*/

@Override

public List<Noticepojo> findAllNotice() {

List<Noticepojo> list;

try {

list = noticeMapper.findAllNotice();

} catch (Exception e) {

return null;

}

return list;

}

}

4.6.8 ShareServiceImp.java类

package com.jx.service.Imp;

import java.util.Date;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.jx.dao.ShareMapper;

import com.jx.pojo.SharePojo;

import com.jx.service.ShareService;

import com.jx.util.FileUtil;

@Service

public class ShareServiceImp implements ShareService{

@Autowired

private ShareMapper shareMapper;

private String PATH="D:\\file\\resoure\\";

/\*\*

\* 查询所有资源

\*/

@Override

public List<SharePojo> findShareAll() {

List<SharePojo> list;

try {

list = shareMapper.findShareAll();

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 添加资源

\*/

@Override

public boolean insertShare(SharePojo sp,CommonsMultipartFile file) {

Map map = FileUtil.fileUp(file, PATH);

if (map == null) return false;

try {

String sharepath = (String)map.get("path")+(String)map.get("newFileName");

sp.setSharepath(sharepath);

shareMapper.insertShare(sp);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 删除资源

\*/

@Override

public boolean deleteShare(Long shareid) {

try {

shareMapper.deleteShare(shareid);

} catch (Exception e) {

return false;

}

return true;

}

}

4.6.9 StudentServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.StudentMapper;

import com.jx.pojo.StudentPojo;

import com.jx.service.StudentService;

@Service

public class StudentServiceImp implements StudentService{

@Autowired

private StudentMapper studentMapper;

/\*\*

\* 根据学号查找学生信息

\*/

@Override

public StudentPojo findById(String studentnum) {

StudentPojo stu;

try {

stu = studentMapper.findById(studentnum);

} catch (Exception e) {

return null;

}

return stu;

}

/\*\*

\* 添加学生信息

\*/

@Override

public Boolean insertStudent(StudentPojo stu) {

try {

studentMapper.insertStudent(stu);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 修改学生信息

\*/

@Override

public Boolean updateStudent(StudentPojo stu) {

try {

studentMapper.updateStudent(stu);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 删除学生信息

\*/

@Override

public Boolean deleteStudent(String studentnum) {

try {

studentMapper.deleteStudent(studentnum);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据学生姓名查询学生信息

\*/

@Override

public List<StudentPojo> findByStudentName(String studentname) {

List<StudentPojo> list;

try {

list = studentMapper.findByStudentName(studentname);

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 查询所有学生信息

\*/

@Override

public List<StudentPojo> findSudentAll() {

List<StudentPojo> list;

try {

list = studentMapper.findSudentAll();

} catch (Exception e) {

return null;

}

return list;

}

}

4.6.10 SupermanServiceImp.java类

package com.jx.service.Imp;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.SupermanMapper;

import com.jx.pojo.SupermanPojo;

import com.jx.service.SupermanService;

@Service

public class SupermanServiceImp implements SupermanService{

@Autowired

private SupermanMapper supermanMapper;

/\*\*

\* 根据管理员账号查询管理员信息

\*/

@Override

public SupermanPojo findById(String supernum) {

SupermanPojo supermanPojo;

try {

supermanPojo = supermanMapper.findById(supernum);

} catch (Exception e) {

return null;

}

return supermanPojo;

}

/\*\*

\* 添加信息管理员信息

\*/

@Override

public boolean insertSuperman(SupermanPojo stp) {

try {

supermanMapper.insertSuperman(stp);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 修改管理员信息

\*/

@Override

public boolean updateSuperman(SupermanPojo stp) {

try {

supermanMapper.updateSuperman(stp);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 删除管理员信息

\*/

@Override

public boolean deleteSuperman(String supernum) {

try {

supermanMapper.deleteSuperman(supernum);

} catch (Exception e) {

return false;

}

return true;

}

}

4.6.11 TaskServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.TaskMapper;

import com.jx.pojo.TaskPojo;

import com.jx.service.TaskService;

@Service

public class TaskServiceImp implements TaskService {

@Autowired

private TaskMapper taskMapper;

/\*\*

\* 添加维修任务

\*/

@Override

public boolean insertTask(TaskPojo taskPojo) {

try {

taskMapper.insertTask(taskPojo);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 查询所有维修任务

\*/

@Override

public List<TaskPojo> findTaskall() {

List<TaskPojo> list;

try {

list = taskMapper.findTaskall();

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 根据维修任务id删除维修任务

\*/

@Override

public boolean deleteTask(Long taskid) {

try {

taskMapper.deleteTask(taskid);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据维修任务状态查询维修任务

\*/

@Override

public List<TaskPojo> findTaskBystaut(Integer taskflag) {

List<TaskPojo> list;

try {

list = findTaskBystaut(taskflag);

} catch (Exception e) {

return null;

}

return list;

}

/\*\*

\* 修改维修任务状态

\*/

@Override

public boolean updateTaskStaut(Long taskid, Integer taskflag) {

try {

taskMapper.updateTaskStaut(taskid, taskflag);

} catch (Exception e) {

return false;

}

return true;

}

}

4.6.12 UpdatmanServiceImp.java类

package com.jx.service.Imp;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.jx.dao.UpdatmanMapper;

import com.jx.pojo.UpdatmanPojo;

import com.jx.service.UpdatmanService;

@Service

public class UpdatmanServiceImp implements UpdatmanService{

@Autowired

private UpdatmanMapper updatmanMapper;

/\*\*

\* 根据维修人员账号查询维修人员信息

\*/

@Override

public UpdatmanPojo findById(String updatmannum) {

UpdatmanPojo updatmanPojo;

try {

updatmanPojo = updatmanMapper.findById(updatmannum);

} catch (Exception e) {

return null;

}

return updatmanPojo;

}

/\*\*

\* 添加维修人员信息

\*/

@Override

public boolean insertUpdatman(UpdatmanPojo upm) {

try {

updatmanMapper.insertUpdatman(upm);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 修改维修人员信息

\*/

@Override

public boolean updateUpdatman(UpdatmanPojo upm) {

try {

updatmanMapper.updateUpdatman(upm);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 删除维修人员信息

\*/

@Override

public boolean deleteUpdatman(String updatmannum) {

try {

updatmanMapper.deleteUpdatman(updatmannum);

} catch (Exception e) {

return false;

}

return true;

}

/\*\*

\* 根据维修人员姓名查询维修人员信息

\*/

@Override

public List<UpdatmanPojo> findByUpdateName(String updatmanname) {

List<UpdatmanPojo> list;

try {

list = updatmanMapper.findByUpdateName(updatmanname);

} catch (Exception e) {

return null;

}

return list;

}

@Override

public List<UpdatmanPojo> findUpdateAll() {

List<UpdatmanPojo> list;

// try {

list = updatmanMapper.findUpdateAll();

// } catch (Exception e) {

// return null;

// }

return list;

}

}

## 4.7 Util层

4.7.1 DateUtil.java类

package com.jx.util;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class DateUtil {

/\*\*

\* 将一个日期字符串转换成日期对象

\* @param datestr

\* @param format

\* @return

\* @throws ParseException

\*/

public static Date formatDate(String datestr,String format) throws ParseException{

SimpleDateFormat simpleDateFormat=new SimpleDateFormat(format);

return simpleDateFormat.parse(datestr);

}

public static void main(String[] args) throws ParseException {

System.out.println(DateUtil.formatDate("2015-2-2 5:23:23", "yyyy-mmm-dd hh:mm:mm"));

}

}

4.7.2 EncodingFilter.java类

package com.jx.util;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* 编码过滤器

\* 从tomcat8.0开始，URIEncoding默认值不再是ISO8859-1，变成了UTF-8

\*

\*/

public class EncodingFilter implements Filter {

@Override

public void destroy() {

}

@Override

public void doFilter(ServletRequest req, ServletResponse res,

FilterChain chain) throws IOException, ServletException {

// 将请求和响应强制转换成Http形式

HttpServletRequest request = (HttpServletRequest) req;

HttpServletResponse response = (HttpServletResponse) res;

// 处理响应乱码

response.setContentType("text/html;charset=utf-8");

// 自定义一个request对象：MyRequest，对服务器原来的requset进行增强，使用装饰设计模�?

// 要增强原来的request对象，必须先获取到原来的request对象

MyRequest myRequest = new MyRequest(request);

// 注意：放行的时�?�应该传入增强后的request对象

chain.doFilter(myRequest, response);

}

@Override

public void init(FilterConfig arg0) throws ServletException {

}

}

4.7.3 FileUtil.java类

package com.jx.util;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.net.URLEncoder;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

public class FileUtil {

public static Map<String, String> fileUp(CommonsMultipartFile file,String path) {

// 获得原始文件名

String fileName = file.getOriginalFilename();

//生成新的文件名

String newFileName = IDUtils.genfileName(fileName);

File f = new File(path);

if (!f.exists())

f.mkdirs();

if (!file.isEmpty()) {

try {

FileOutputStream fos = new FileOutputStream(path + newFileName);

InputStream in = file.getInputStream();

int b = 0;

while ((b = in.read()) != -1) {

fos.write(b);

}

fos.close();

in.close();

} catch (Exception e) {

return null;

}

}

Map<String, String> map = new HashMap<>();

map.put("newFileName", newFileName);

map.put("path", path);

//返回文件全路径

return map;

}

public static boolean fileDown(HttpServletResponse response,String filepath,String filename) {

try {

//获取输入流

InputStream bis = new BufferedInputStream(new FileInputStream(new File(filepath)));

//转码，免得文件名中文乱码

filename = URLEncoder.encode(filename,"UTF-8");

//设置文件下载头

response.addHeader("Content-Disposition", "attachment;filename=" + filename);

//1.设置文件ContentType类型，这样设置，会自动判断下载文件类型

response.setContentType("multipart/form-data");

BufferedOutputStream out = new BufferedOutputStream(response.getOutputStream());

int len = 0;

while((len = bis.read()) != -1){

out.write(len);

out.flush();

}

out.close();

} catch (Exception e) {

return false;

}

return true;

}

}

4.7.4 IDUtils.java类

package com.jx.util;

import java.util.Random;

/\*\*

\* 文件名生成类

\* @author fdm

\*/

public class IDUtils {

/\*\*

\* 文件名id生成

\*/

public static String genfileName(String oldName) {

//取当前时间的长整形值包含毫秒

long millis = System.currentTimeMillis();

//long millis = System.nanoTime();

//加上三位随机数

Random random = new Random();

int end3 = random.nextInt(999);

//如果不足三位前面补0

String str = millis + String.format("%03d", end3);

String newName = str+oldName.substring(oldName.lastIndexOf("."));//加上图片扩展名

return newName;

}

}

4.7.5 JsonUtils.java类

package com.jx.util;

import java.util.List;

import com.fasterxml.jackson.core.JsonProcessingException;

import com.fasterxml.jackson.databind.JavaType;

import com.fasterxml.jackson.databind.ObjectMapper;

/\*\*

\* json转换工具类

\*/

public class JsonUtils {

// 定义jackson对象

private static final ObjectMapper MAPPER = new ObjectMapper();

/\*\*

\* 将对象转换成json字符串。

\* @param data

\* @return

\*/

public static String objectToJson(Object data) {

try {

String string = MAPPER.writeValueAsString(data);

return string;

} catch (JsonProcessingException e) {

e.printStackTrace();

}

return null;

}

/\*\*

\* 将json结果集转化为对象

\* @param jsonData json数据

\* @param clazz 对象中的object类型

\* @return

\*/

public static <T> T jsonToPojo(String jsonData, Class<T> beanType) {

try {

T t = MAPPER.readValue(jsonData, beanType);

return t;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

/\*\*

\* 将json数据转换成pojo对象list

\* @param jsonData

\* @param beanType

\* @return

\*/

public static <T>List<T> jsonToList(String jsonData, Class<T> beanType) {

JavaType javaType = MAPPER.getTypeFactory().constructParametricType(List.class, beanType);

try {

List<T> list = MAPPER.readValue(jsonData, javaType);

return list;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

4.7.6 LoginInterceptor.java类

package com.jx.util;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.web.servlet.HandlerInterceptor;

import org.springframework.web.servlet.ModelAndView;

public class LoginInterceptor implements HandlerInterceptor {

@Override

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)

throws Exception {

// Integer UserId = (Integer) request.getSession().getAttribute("UserId");

//if(UserId != null) return true;

return false;

}

@Override

public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler,

ModelAndView modelAndView) throws Exception {

}

@Override

public void afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex)

throws Exception {

}

}

# 前台源代码

## 主页面代码

* + 1. Index

<template>

<div>

<div class="columns" style="margin-top: -60px;overflow: auto;height: 120px;overflow: hidden;">

<div class="column is-12 "

style="padding: 0;background: #232332;height: 60px;margin-top: 0;color: white;">

<div class="columns is-gapless" style="font-size:20px;">

<div class="column is-1 has-text-centered"

style="font-size: 40px;text-align: center;margin-left: 20px;line-height:60px;">

<i class="fa fa-hotel" style="font-size: 40px;text-align: center;margin-left: 20px;line-height:60px;"></i>

</div>

<div class="column is-3 has-text-centered"

style="text-align: left;line-height: 60px;"> 寝室报修管理系统</div>

<div class="column">

</div>

<div class="column">

</div>

<div class="column">

<i class="fa fa-user-circle-o" style="line-height: 60px;font-size: 15px;margin-right: 5px;"></i>

<span style="line-height: 60px;font-size: 15px;margin-right: 5px;color: skyblue;">{{uid}}</span>

</div>

</div>

<div class="column is-12" style="margin-top: -35px;text-align: left;">

<div class="columns is-gapless">

<div class="column">

<p style="font-size: 20px;color: black;line-height: 60px;text-align: center;font-family: elephant;">WELCONME TO HERE</p>

</div>

<div class="column">

<div class="columns is-gapless">

<div class="column">

<p class=""></p>

</div>

<div class="column">

<p class=""></p>

</div>

</div>

</div>

<div class="column">

<div class="columns is-gapless">

<div class="column">

<p class="is-info" style="line-height: 60px;">

<a href="http://jwc.hbuas.edu.cn/content/index">

<i class="fa fa-fort-awesome"

style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>校园在线教务系统</span>

</a>

</p>

</div>

<div class="column">

<p class="is-info" style="line-height: 60px;">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-windows"

style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>计算机等级</span>

</a>

</p>

</div>

</div>

</div>

<div class="column">

<div class="columns is-gapless">

<div class="column">

<p class="is-info" style="line-height: 60px;">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-newspaper-o"

style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>普通话</span>

</a>

</p>

</div>

<div class="column">

<p class="is-info" style="line-height: 60px;">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-life-buoy"

style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>CET等级</span>

</a>

</p>

</div>

</div>

</div>

<div class="column">

<div class="columns is-gapless">

<div class="column">

<p class="is-info" style="line-height: 60px;">

<router-link to="/notice">

<i class="fa fa-facebook" style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>公告信息</span>

</router-link>

</p>

</div>

<div class="column">

<p class="is-info" style="line-height: 60px;">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-server"

style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>其他</span>

</a>

</p>

</div>

</div>

</div>

<div class="column">

<div class="columns is-gapless">

<div class="column">

<p class="is-info" style="line-height: 60px;">

<router-link to="/user/alter">

<i class="fa fa-unlock-alt"

style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>修改密码</span>

</router-link>

</p>

</div>

<div class="column">

<p class="is-info" style="line-height: 60px;">

<router-link to="/">

<i class="fa fa-user-circle-o" style="line-height: 60px;font-size: 20px;margin-right: 5px;"></i>

<span>退出登录</span>

</router-link></p>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="columns is-gapless">

<div

class="column is-12 is-primary has-text-centered"

style="padding: 0;background: white;height: 750px;">

<div class="columns is-gapless">

<div class="column is-2 newclass" style="overflow: auto;height: 750px;">

<!--菜单-->

<div class="menu" >

<menus label="菜单操作">

<div style="text-align: left;" >

<!--管理员--> <div v-show="show\_admin">

<menu-item icon="server">

<span>数据管理</span>

<menus slot="sub">

<div>

<div style="margin-left: 20px;">

<router-link to="/student">

<i class="fa fa-vcard"></i>

<span>学生管理</span>

</router-link>

</div>

<div style="margin-left: 20px;">

<router-link to="/houseman">

<i class="fa fa-vcard"></i>

<span>宿管管理</span>

</router-link>

</div>

<div style="margin-left: 20px;">

<router-link to="/updatman">

<i class="fa fa-vcard"></i>

<span>维修管理</span>

</router-link>

</div>

<div style="margin-left: 20px;">

<router-link to="/resource">

<i class="fa fa-vcard"></i>

<span>资源管理</span>

</router-link>

</div>

<div style="margin-left: 20px;">

<router-link to="/allottask">

<i class="fa fa-vcard"></i>

<span>维修任务分配</span>

</router-link>

</div>

<div style="margin-left: 20px;">

<router-link to="/power">

<i class="fa fa-vcard"></i>

<span>身份识别和权限管理</span>

</router-link>

</div>

<div style="margin-left: 20px;">

<router-link to="/reply">

<i class="fa fa-vcard"></i>

<el-badge :value="count" :max="99" class="item">

<span>投诉反馈</span>

</el-badge>

</router-link>

</div>

<div style="margin-left: 20px;">

<router-link to="/service">

<i class="fa fa-vcard"></i>

<span>服务评价统计</span>

</router-link>

</div>

</div>

</menus>

</menu-item>

</div>

<div class="divider"></div>

<menu-item icon="universal-access">

<span>个人操作</span>

<menus slot="sub">

<div>

<div style="margin-left: 20px;">

<router-link to="/selfmsg">

<i class="fa fa-edit"></i>

<span>修改信息</span>

</router-link>

</div>

<div v-show="show\_student" style="margin-left: 20px;">

<router-link to="/share">

<i class="fa fa-external-link"></i>

<span>寝室共享</span>

</router-link>

</div>

</div>

</menus>

</menu-item>

<div v-show="show\_studentq">

<div class="divider"></div>

<div style="margin: 0;">

<router-link to="/fix">

<i class="fa fa-gavel"></i>

<span>寝室报修</span>

</router-link>

</div>

</div>

<div v-show="show\_update">

<div class="divider"></div>

<div style="margin: 0;">

<router-link to="/task">

<i class="fa fa-map-signs"></i>

<el-badge :value="200" :max="99" class="item">

<span>查看维修任务</span>

</el-badge >

</router-link>

</div>

</div>

<div v-show="show\_studentq">

<div class="divider"></div>

<div style="margin: 0;">

<router-link to="/fix">

<i class="fa fa-meh-o"></i>

<span>维修反馈</span>

</router-link>

</div>

</div>

<div v-show="show\_house">

<div class="divider"></div>

<div style="margin: 0;">

<router-link to="/newmsg">

<i class="fa fa-comments-o"></i>

<span>消息发布</span>

</router-link>

</div>

</div>

<div>

<div class="divider"></div>

<div style="margin: 0;">

<router-link to="/notice">

<i class="fa fa-facebook"></i>

<span>公告信息</span>

</router-link>

</div>

</div>

<div v-show="show\_student">

<div class="divider"></div>

<div style="margin: 0;">

<router-link to="/assess">

<i class="fa fa-calendar-check-o"></i>

<span>维修评价</span>

</router-link>

</div>

</div>

<div v-show="show\_student">

<div class="divider"></div>

<div style="margin: 0;">

<router-link to="/complain">

<i class="fa fa-exclamation-circle"></i>

<span>投诉</span>

</router-link>

</div>

</div>

<div v-show="show\_student">

<div class="divider"></div>

<div style="margin: 0;" @click="badgemsg">

<router-link to="/msg">

<i class="fa fa-commenting"></i>

<el-badge :value="countmsg" :max="99" class="item" :hidden="hid">

<span>回馈消息</span>

</el-badge>

</router-link>

</div>

</div>

</div>

</menus>

</div>

</div>

<div class="column" style="height: 800px;overflow: auto;">

<div style="padding: 20px;margin-top: -10px;overflow: auto;">

<router-view/>

</div>

</div>

</div>

<div>

</div>

</div>

</div>

<div class="columns" style="">

<div class="column is-12 "

style="padding: 0;background: #232332;height: 200px;margin-top: 0;color: white;">

<p>欢迎使用本系统</p>

<div style="text-align: right;position：flex;bottom:-60px;z-index: 1000;">

<affix :offset="10" style="z-index: 1000;">

<button class="button is-info is-outlined"

@click="backLogin"

style="height: 40px;margin-right: 20px;">

<i class="fa fa-user-circle"></i>

<p>退出</p>

</button>

</affix>

<affix :offset="50">

<button class="button is-info is-outlined" style="height: 40px;margin-right: 20px;">

<i class="fa fa-send"></i>

<p>其他</p>

</button>

</affix>

</div>

</div>

</div>

</div>

</template>

<script>

const uid=''

const flag=''

import {delCookie} from '@/common/config/cookie'

export default {

data() {

return {

show\_admin:false,

show\_student:false,

show\_studentq:false,

show\_house:false,

show\_update:false,

hid:false,

user:[],

name:localStorage.getItem("username"),

uid:localStorage.getItem("uid"),

count:'',

countmsg:'',

};

},

created(){

this.$api.user.findtaskall().then(response=>{

for(let i=0;i<response.length;i++){

if(response[i].taskman==localStorage.getItem("username")){

this.count++

}

}

})

this.$api.user.findcomplainId(localStorage.getItem("uid")).then(response=>{

for(let i=0;i<response.length;i++){

this.countmsg++

}

});

},

mounted(){

this.flag = localStorage.getItem("flag")

if(this.flag==1){

this.show\_student=true

this.show\_studentq=true

}

if(this.flag==2){

this.show\_house=true

this.show\_studentq=true

}

if(this.flag==3){

this.show\_update=true

this.show\_studentq=false

}

if(this.flag==4){

this.show\_student=true

this.show\_admin=true

this.show\_house=true

this.show\_update=true

this.show\_studentq=true

}

},

methods: {

backLogin(){

this.$store.commit('changeLogin','0')

this.$router.push("/")

delCookie("user")

delCookie("password")

},

badgemsg(){

this.hid=true

}

}

}

</script>

<style>

body{

overflow-x: hidden;

}

body::-webkit-scrollbar {

display: none;

}

.newclass::-webkit-scrollbar {

display: none;

}

</style>

* + 1. Home

<template>

<div>

<el-card class="box-card">

<div slot="header" class="clearfix">

<i class="fa fa-fort-awesome"

style="line-height: 30px;font-size: 30px;margin-right: 10px;"></i>

<span style="font-size: 24px;">我们的校园</span>

</div>

<div>

<el-carousel :interval="4000" type="card" height="300px">

<div v-for="item in form">

<el-carousel-item>

<div><img :src="item.img" /></div>

<!--<h1>{{item.img}}</h1>-->

</el-carousel-item>

</div>

</el-carousel>

</div>

<div class="columns is-gapless">

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<a href="http://jwc.hbuas.edu.cn/content/index">

<i class="fa fa-fort-awesome"

style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>校园在线教务系统</span>

</a>

</el-card>

</div>

</div>

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-windows"

style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>计算机等级</span>

</a>

</el-card>

</div>

</div>

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-newspaper-o"

style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>普通话</span>

</a>

</el-card>

</div>

</div>

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-life-buoy"

style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>CET</span>

</a>

</el-card>

</div>

</div>

</div>

<div class="columns is-gapless">

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<router-link to="/notice">

<i class="fa fa-facebook" style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>公告信息</span>

</router-link>

</el-card>

</div>

</div>

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<a href="http://cjcx.neea.edu.cn/html1/folder/1508/206-1.htm?sid=300">

<i class="fa fa-server"

style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>其他</span>

</a>

</el-card>

</div>

</div>

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<router-link to="/user/alter">

<i class="fa fa-unlock-alt"

style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>修改密码</span>

</router-link>

</el-card>

</div>

</div>

<div class="column">

<div class="newcard">

<el-card shadow="hover">

<router-link to="/">

<i class="fa fa-user-circle-o" style="line-height: 20px;font-size: 30px;margin-right: 10px;"></i>

<span>退出登录</span>

</router-link>

</el-card>

</div>

</div>

</div>

</el-card >

</div>

</template>

<script>

export default {

data() {

return {

form:[

{img:'https://ss2.bdstatic.com/70cFvnSh\_Q1YnxGkpoWK1HF6hhy/it/u=1049238752,1099518948&fm=27&gp=0.jpg',id:0},

{img:"https://ss1.bdstatic.com/70cFvXSh\_Q1YnxGkpoWK1HF6hhy/it/u=3568325471,949016628&fm=27&gp=0.jpg",id:1},

{img:"https://ss1.bdstatic.com/70cFvXSh\_Q1YnxGkpoWK1HF6hhy/it/u=3561022367,1981579104&fm=27&gp=0.jpg",id:2},

{img:"https://ss1.bdstatic.com/70cFvXSh\_Q1YnxGkpoWK1HF6hhy/it/u=1967725943,463061552&fm=27&gp=0.jpg",id:3}

],

};

},

}

</script>

<style>

.newcard:hover{

cursor: pointer;

}

</style>

* + 1. Assess

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/complain" exact><i class="fa fa-tag"></i>消息发布</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div style="height: 600px;overflow: auto;">

<el-card class="box-card" style="text-align: left;">

<div slot="header" class="clearfix">

<span>投诉</span>

</div>

<div class="text item" style="width: 500px;">

<el-form status-icon label-width="100px" class="demo-ruleForm">

<el-form-item label="学号" prop="num">

<el-input type="text" v-model="form.num" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="姓名" prop="name">

<el-input type="text" v-model="form.name" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="投诉账号" prop="tnum">

<el-input type="text" v-model="form.tnum" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="投诉姓名" prop="tname">

<el-input type="text" v-model="form.tname" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="投诉内容描述" prop="desc">

<el-input

type="textarea"

:rows="5"

placeholder="请输入内容"

v-model="form.desc">

</el-input>

</el-form-item>

<el-form-item>

<el-button type="primary" @click="submitForm()">提交</el-button>

</el-form-item>

</el-form>

</div>

</el-card >

<el-card class="box-card" >

<div slot="header" class="clearfix" style="text-align: left;">

<span>可投诉维修人员信息</span>

</div>

<div style="width: 100%;text-align: left;height: 400px;overflow: auto;">

<el-table

:data="data"

style="width: 100%;text-align: left;"

max-height="600">

<el-table-column

prop="updatmanname"

label="姓名"

width="150">

</el-table-column>

<el-table-column

prop="updatmannum"

label="账号"

width="150">

</el-table-column>

<el-table-column

prop="updatmansex"

label="性别"

width="220">

</el-table-column>

<el-table-column

prop="updatmantel"

label="联系电话"

width="150">

</el-table-column>

<el-table-column

prop="updatmanqq"

label="QQ号码"

width="200">

</el-table-column>

<el-table-column

prop="updatmanpassword"

label="密码"

width="150">

</el-table-column>

</el-table>

</div>

</el-card>

</div>

</div>

</template>

<script>

const flag = 3

export default {

data() {

return {

form:[

{

num:'',

name:'',

tnum:'',

tname:'',

desc:''

}

],

data:[]

}

},

created(){

this.$api.admin.findbyuserall(flag).then(response=>{

for(let i=0;i<response.length;i++){

this.data.push(response[i])

}

});

},

methods: {

submitForm(){

this.$api.user.insertcom(

this.form.num,

this.form.name,

this.form.tnum,

this.form.tname,

this.form.desc

).then(response=>{

if(response){

this.$notify.open({

content: '投诉成功，管理员将会尽快给您回复',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '失败！！！！重新尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

})

}

}

}

</script>

<style>

</style>

* + 1. Complain

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/fix" exact><i class="fa fa-tag"></i>寝室报修</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div style="">

<el-card class="box-card" style="text-align: left;">

<div slot="header" class="clearfix">

<span>寝室报修</span>

</div>

<div style="width: 500px;">

<el-form status-icon ref="form" label-width="100px" class="demo-ruleForm">

<el-form-item label="学号：" prop="studentnum">

<el-input type="number" v-model="form.studentnum" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="发布时间" prop="fixtime">

<el-date-picker

v-model="form.fixtime"

type="date"

@change="getTime"

value-format="yyyy-MM-dd hh:mm:ss"

placeholder="选择日期">

</el-date-picker>

</el-form-item>

<el-form-item label="报修人：" prop="studentname">

<el-input type="text" v-model="form.studentname" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="报修图片：" style="text-align: left;">

<el-upload

class="upload-demo"

ref="upload"

:action="address"

:data="form"

multiple

:onError="uploadError"

:onSuccess="uploadSuccess"

:before-upload="beforeUpload"

:auto-upload="false"

list-type="picture">

<el-button slot="trigger" size="small" type="primary">选取图片</el-button>

</el-upload>

</el-form-item>

<el-form-item label="楼栋：" style="text-align: left;">

<el-input type="text" v-model="form.studentcode" auto-complete="off"></el-input>

</el-form-item >

<el-form-item label="寝室号：" style="text-align: left;">

<el-input type="text" v-model="form.studenthouse" auto-complete="off"></el-input>

</el-form-item >

<el-form-item label="损坏描述：" prop="desc">

<el-input type="textarea"

:rows="5"

v-model="form.fixdesc"></el-input>

</el-form-item>

<el-form-item style="text-align: left;">

<el-button type="primary" @click="submitUpload">提交报修</el-button>

</el-form-item>

</el-form>

</div>

</el-card>

</div>

</div>

</template>

<script>

import {api\_baseUrl} from '@/common/base/baseUrl/env'

//文件地址

const adr='insertfix'

export default {

data() {

return {

address:api\_baseUrl+adr,

form:{

studentnum:'',

studentname:'',

fixtime:'',

studentcode:'',

studenthouse:'',

fixdesc:'',

fixflag:'0'

},

}

},

methods: {

submitUpload() {

this.$refs.upload.submit();

},

//上传之前

beforeUpload(file){

console.log(file)

},

handleRemove(file, fileList) {

console.log(file, fileList);

},

handlePreview(file) {

console.log(file);

},

// 上传成功后的回调

uploadSuccess (response, file, fileList) {

if(response){

this.$notify.open({

content: '报修成功，请等待受理',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '报修失败！！！！重新尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

},

// 上传错误

uploadError (response, file, fileList) {

if(response){

this.$notify.open({

content: '报修失败！！！！重新尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

},

getTime(date){

this.form.time=date

}

}

}

</script>

<style>

</style>

* + 1. Fix

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/msg" exact><i class="fa fa-tag"></i>消息发布</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div style="">

<el-card class="box-card" style="text-align: left;">

<div slot="header" class="clearfix">

<span>

投诉消息

</span>

</div>

<div class="text item" v-for="item in form">

<el-collapse accordion>

<el-collapse-item :title="title">

<p style="margin-left: 20px;">{{item.feedbackdesc}}</p>

</el-collapse-item>

</el-collapse>

</div>

</el-card >

</div>

</div>

</template>

<script>

export default {

data() {

return {

form:[],

count:0,

title:'',

}

},

created(){

this.$api.user.findcomplainId(localStorage.getItem("uid")).then(response=>{

for(let i=0;i<response.length;i++){

this.form.push(response[i])

this.count++

this.title="你有"+this.count+"条回馈消息"

}

});

},

methods: {

submitForm(){

}

}

}

</script>

<style>

</style>

* + 1. NewMsg

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/newmsg" exact><i class="fa fa-tag"></i>消息发布</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div style="">

<el-card class="box-card" style="text-align: left;">

<div slot="header" class="clearfix">

<span>发布最新公告</span>

<!--<el-button style="float: right; padding: 3px 0" type="text">操作按钮</el-button>-->

</div>

<div class="text item" style="width: 500px;">

<el-form status-icon label-width="100px" class="demo-ruleForm">

<el-form-item label="账号" prop="num">

<el-input type="text" v-model="form.num" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="发布人" prop="name">

<el-input type="text" v-model="form.name" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="发布楼栋" prop="code">

<el-input type="text" v-model="form.code" auto-complete="off"></el-input>

</el-form-item>

<el-form-item label="发布时间" prop="time">

<el-date-picker

v-model="form.time"

type="date"

@change="getTime"

value-format="yyyy-MM-dd"

placeholder="选择日期">

</el-date-picker>

</el-form-item>

<el-form-item label="公告描述（注明：楼栋号）" prop="desc">

<el-input

type="textarea"

:rows="5"

placeholder="请输入描述内容（请注明：楼栋号）"

v-model="form.desc">

</el-input>

</el-form-item>

<el-form-item>

<el-button type="primary" @click="submitForm()">提交发布</el-button>

</el-form-item>

</el-form>

</div>

</el-card >

</div>

</div>

</template>

<script>

export default {

data() {

return {

form:[

{

name:'',

num:'',

code:'',

time:'',

desc:''}

]

}

},

methods: {

submitForm(){

this.$api.user.releasenotice(

this.form.num,

this.form.name,

this.form.desc,

this.form.code

).then(response=>{

if(response){

this.$notify.open({

content: '通知发布成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '失败！！！！重新尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

});

},

getTime(date){

this.form.time=date

}

}

}

</script>

<style>

</style>

* + 1. Notice

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/notice" exact><i class="fa fa-tag"></i>公告信息</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div style="">

<el-card class="box-card" style="text-align: left;max-height: 500px;overflow: auto;">

<div slot="header" class="clearfix">

<span>最新公告</span>

<!--<el-button style="float: right; padding: 3px 0" type="text">操作按钮</el-button>-->

</div>

<div v-for="o in data" class="text item" style="text-align: left;margin-bottom: 10px;">

<p>发布时间：{{o.noticetime}}</p>

<p>发布人：{{o.housemanname}}</p>

<p>发布楼栋：{{o.noticehose}}栋</p>

<p>公告/通知：{{o.noticedesc}}</p>

</div>

</el-card>

<el-card class="box-card" style='text-align: left;max-height: 500px;overflow: auto;'>

<div slot="header" class="clearfix">

<span>最近维修评价</span>

</div>

<div class="text item bcc">

<div v-for="item in items" style="margin-bottom: 20px;">

<el-collapse>

<el-collapse-item>

<template slot="title">

<i class="fa fa-exclamation-triangle" style="line-height: 48px;color: red;margin-left: 10px;"></i>

来至账号: {{item.studentnum}} 姓名：{{item.studentname}} 评价

</template>

<div><span>评价时间：{{item.assesstime}}</span></div>

<div><span>评价人：{{item.studentname}}</span></div>

<div><span>描述：{{item.assessdesc}}</span></div>

<div><span>报修星级评分</span> </div>

<el-rate

v-model="item.assesscode"

disabled

show-score

text-color="#ff9900"

score-template="{value}"

>

</el-rate>

</el-collapse-item>

</el-collapse>

</div>

</div>

</el-card>

</div>

</div>

</template>

<script>

import {formatDate} from '@/common/config/data'

export default {

data() {

return {

items:[],

data:[]

}

},

created(){

this.$api.user.findassessall().then(response=>{

for(let i =0;i<response.length;i++){

this.items.push(response[i])

this.items[i].assesstime=formatDate(new Date(this.items[i].assesstime), 'yyyy-MM-dd hh:mm:ss');

}

});

this.$api.user.findallnotice().then(response=>{

for(let i =0;i<response.length;i++){

this.data.push(response[i])

this.data[i].noticetime=formatDate(new Date(this.data[i].noticetime), 'yyyy-MM-dd hh:mm:ss');

}

});

},

methods: {

}

}

</script>

<style scoped>

.bcc{

max-height: 340px;overflow: auto;

}

.bcc::-webkit-scrollbar {

display: none;

}

</style>

* + 1. Selfmsg

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-universal-access"></i>个人操作</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/selfmsg" exact><i class="fa fa-tag"></i>修改信息</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div style="text-align: left;">

<el-card class="box-card" style="text-align: left;">

<div slot="header" class="clearfix">

<span>修改个人信息</span>

</div>

<div style="width: 500px;text-align: left;"class="bgc">

<div v-show="show\_student">

<label class="label" style="text-align: left;">账号：</label>

<p class="control">

<input class="input is-success is-medium" type="number" v-model="form.studentnum">

</p>

<label class="label" style="text-align: left;">姓名：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.studentname">

</p>

<label class="label" style="text-align: left;">性别：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.studentsex">

</p>

<label class="label" style="text-align: left;">电话：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="tel" v-model="form.studenttel">

</p>

<div>

<label class="label" style="text-align: left;">楼栋号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.studentcode">

</p>

</div>

<div>

<label class="label" style="text-align: left;">寝室号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.studenthouse">

</p>

</div>

<div>

<label class="label" style="text-align: left;">班级：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.studentclass">

</p>

</div>

<label class="label" style="text-align: left;">qq号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.studentqq">

</p>

</div>

<div v-show="show\_house">

<label class="label" style="text-align: left;">账号：</label>

<p class="control">

<input class="input is-success is-medium" type="number" v-model="form.housemannum">

</p>

<label class="label" style="text-align: left;">姓名：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.housemanname">

</p>

<label class="label" style="text-align: left;">性别：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.housemansex">

</p>

<label class="label" style="text-align: left;">电话：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="tel" v-model="form.housemantel">

</p>

<div>

<label class="label" style="text-align: left;">楼栋号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.housemancode">

</p>

</div>

<label class="label" style="text-align: left;">qq号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.housemanqq">

</p>

</div>

<div v-show="show\_update">

<label class="label" style="text-align: left;">账号：</label>

<p class="control">

<input class="input is-success is-medium" type="number" v-model="form.updatmannum">

</p>

<label class="label" style="text-align: left;">姓名：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.updatmanname">

</p>

<label class="label" style="text-align: left;">性别：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.updatmansex">

</p>

<label class="label" style="text-align: left;">电话：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="tel" v-model="form.updatmantel">

</p>

<label class="label" style="text-align: left;">qq号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.updatmanqq">

</p>

</div>

<div style="text-align: left;"><a

class="button is-primary"

@click="saveClick"

style="padding: 20px;">保存</a></div>

</div>

</el-card>

</div>

</div>

</template>

<script>

const flag=''

import {getCookie} from '@/common/config/cookie'

export default {

data() {

return {

show\_student:false,

show\_house:false,

show\_update:false,

form:{},

user:[],

data:[],

options: [{

value: 15,

label: 15

}],

};

},

created(){

this.$api.user.findbyuserid(localStorage.getItem("uid"),localStorage.getItem("uid"),localStorage.getItem("uid"),localStorage.getItem("flag")).then(response=>{

this.form = response

this.data = response

console.log(this.form)

});

},

mounted(){

this.flag = localStorage.getItem("flag")

if(this.flag==1){

this.show\_student=true

}

if(this.flag==2){

this.show\_house=true

}

if(this.flag==3){

this.show\_update=true

}

},

methods: {

saveClick(){

console.log(this.form)

if(this.flag==1){

this.$api.user.studentupdate(

this.flag,

this.form.studentnum,

this.form.studentname,

this.form.studenttel,

this.form.studentsex,

this.form.studenthouse,

this.form.studentcode,

this.form.studentclass,

this.form.studentqq,

this.form.studentpassword

).then(response=>{

if(response){

this.$notify.open({

content: '修改成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '修改失败',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

});

}

if(this.flag==2){

this.$api.user.studentupdate(

this.form.studentnum,

this.form.studentname,

this.form.studenttel,

this.form.studentcode,

this.form.studenthouse,

this.form.studentclass,

this.form.updatmanqq,

this.form.updatmanpassword,

localStorage.getItem("flag")

).then(response=>{

if(response){

this.$notify.open({

content: '修改成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '修改失败',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

});

}

if(this.flag==3){

this.$api.user.updatupdate(

this.flag,

this.form.updatmanname,

this.form.updatmannum,

this.form.updatmanpassword,

this.form.updatmanqq,

this.form.updatmansex,

this.form.updatmantel

).then(response=>{

if(response){

this.$notify.open({

content: '修改成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '修改失败',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

});

}

}

}

}

</script>

<style>

.bgc{

height: 600px;overflow: auto;

}

.bgc::-webkit-scrollbar {

display: none;

}

</style>

* + 1. Share

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/share" exact><i class="fa fa-universal-access"></i>个人操作</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/share" exact><i class="fa fa-tag"></i>寝室共享</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div>

<div style="font-size: 20px;margin-bottom: 20px; text-align: left;">

<el-card class="box-card">

<div slot="header" class="clearfix">

<span>寝室环境共享</span>

</div>

<div class="text item">

{{textdesc}}

</div>

</el-card>

</div>

<div>

<el-upload

class="upload-demo"

drag

ref="upload"

:action="address"

:data="upLoadData"

multiple

:onError="uploadError"

:onSuccess="uploadSuccess"

:before-upload="beforeUpload"

:auto-upload="false">

<i class="el-icon-upload"></i>

<div class="el-upload\_\_text">将文件拖到此处，或<em>点击上传</em></div>

</el-upload>

<div style="margin-top: 10px;margin-bottom: 10px;">

<el-form :inline="true" class="demo-form-inline">

<el-form-item label="文件类型">

<el-select v-model="upLoadData.sharetype" placeholder="请选择文件类型">

<el-option

v-for="item in options"

:key="item.value"

:label="item.label"

:value="item.value">

</el-option>

</el-select>

</el-form-item >

</el-form>

</div>

<el-button

style="margin-left: 10px;padding: 10px;"

size="small"

type="success"

@click="submitUpload"round>点击上传</el-button>

</div>

</div>

</div>

</template>

<script>

import {api\_baseUrl} from '@/common/base/baseUrl/env'

//文件地址

const adr='uploadfile'

const uid=localStorage.getItem("uid")

const name=localStorage.getItem("name")

export default {

data() {

return {

options: [{

value: '图片',

label: '图片'

}, {

value: '视频',

label: '视频'

}, {

value: '其他',

label: '其他'

}],

address:api\_baseUrl+adr,

upLoadData:{

studentnum:uid,

studentname:name,

sharetype:''

},

value: '',

textdesc:'上传寝室维修、建设环境效果，让更多人知道吧'

};

},

watch: {

value:{

handler(){

this.address=api\_baseUrl+adr

}

}

},

methods: {

//上传之前

beforeUpload(file){

},

submitUpload() {

this.$refs.upload.submit();

},

handleRemove(file, fileList) {

},

handlePreview(file) {

},

// 上传成功后的回调

uploadSuccess (response, file, fileList) {

if(response){

this.$notify.open({

content: '上传成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '失败！！！！重新尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

},

// 上传错误

uploadError (response, file, fileList) {

if(response){

this.$notify.open({

content: '上传失败！！！！重新尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

},

}

}

</script>

<style>

</style>

* + 1. Task

<template>

<div >

<div style="text-align: left;margin-bottom: 20px;">

<breadcrumb>

<breadcrumb-item>

<router-link to="/home" exact><i class="fa fa-home"></i>首页</router-link>

</breadcrumb-item>

<breadcrumb-item>

<router-link to="/task" exact><i class="fa fa-tag"></i>维修任务表</router-link>

</breadcrumb-item>

</breadcrumb>

</div>

<div style="">

<el-card class="box-card" style="text-align: left;">

<div style="">

<el-card class="box-card" style="text-align: left;">

<div slot="header" class="clearfix">

<span>

你的维修任务

</span>

</div>

<div style="">

<el-table

:data="data.slice((currentPage-1)\*pagesize,currentPage\*pagesize)"

style="width: 100%;text-align: left;"

max-height="600">

<el-table-column

prop="taskid"

label="任务编号"

width="150">

</el-table-column>

<el-table-column

prop="taskman"

label="维修处理人"

width="150">

</el-table-column>

<el-table-column

prop="taskaddress"

label="处理地址"

width="150">

</el-table-column>

<el-table-column

prop="taskcreatetime"

label="处理时间"

width="150">

</el-table-column>

<el-table-column

prop="taskdesc"

label="任务描述"

width="150">

</el-table-column>

<el-table-column

prop="taskflag"

label="任务状态"

width="150">

</el-table-column>

<el-table-column

prop="taskfirst"

label="任务优先级"

width="200">

</el-table-column>

<el-table-column

label="操作"

width="200">

<template slot-scope="scope">

<el-button

size="mini"

type="primary"

@click="handleDelete(scope.$index, scope.row,data)">任务维修完成</el-button>

</template>

</el-table-column>

</el-table>

</div>

<div>

<div style="margin-top: 10px;text-align: center;">

<el-pagination

background

@current-change="change"

@size-change="sizeChange"

layout="prev, pager, next"

:page-size="pagesize"

:total="total">

</el-pagination>

</div>

</div>

</el-card >

</div>

<div style="">

<el-card class="box-card" style="text-align: left;">

<div slot="header" class="clearfix">

<span>

所有维修任务

</span>

</div>

<div style="">

<el-table

:data="form.slice((currentPage-1)\*pagesize,currentPage\*pagesize)"

style="width: 100%;text-align: left;"

max-height="600">

<el-table-column

prop="taskid"

label="任务编号"

width="150">

</el-table-column>

<el-table-column

prop="taskman"

label="维修处理人"

width="150">

</el-table-column>

<el-table-column

prop="taskaddress"

label="处理地址"

width="150">

</el-table-column>

<el-table-column

prop="taskcreatetime"

label="处理时间"

width="150">

</el-table-column>

<el-table-column

prop="taskdesc"

label="任务描述"

width="150">

</el-table-column>

<el-table-column

prop="taskflag"

label="任务状态"

width="150">

</el-table-column>

<el-table-column

prop="taskfirst"

label="任务优先级"

width="200">

</el-table-column>

</el-table>

</div>

<div>

<div style="margin-top: 10px;text-align: center;">

<el-pagination

background

@current-change="change"

@size-change="sizeChange"

layout="prev, pager, next"

:page-size="pagesize"

:total="total">

</el-pagination>

</div>

</div>

</el-card >

</div>

</el-card >

</div>

</div>

</template>

<script>

import {formatDate} from '@/common/config/data'

const TASKFLAG=1

const FIXFLAG=2

export default {

data() {

return {

form:[],

data:[],

currentPage:1,

pagesize:5,

total:10,

}

},

created(){

this.$api.user.findtaskall().then(response=>{

this.form = response

for(let i=0;i<response.length;i++){

this.form[i].taskcreatetime=formatDate(new Date(this.form[i].taskcreatetime), 'yyyy-MM-dd hh:mm:ss');

}

for(let i=0;i<response.length;i++){

if(response[i].taskman==localStorage.getItem("name")){

this.data.push(response[i])

this.data[i].taskcreatetime=formatDate(new Date(this.data[i].taskcreatetime), 'yyyy-MM-dd hh:mm:ss');

}

}

})

},

methods: {

submitForm(){

},

change(value){

this.currentPage=value

},

sizeChange(value){

this.pagesize=value

},

handleDelete(index,row,rows){

this.$api.user.updatetaskstaut(row.taskid,TASKFLAG).then(response=>{

if(response){

this.$api.admin.updatefix(row.taskid,FIXFLAG).then(response=>{

this.$api.user.deletetask(row.taskid).then(response=>{

if(response){

this.$notify.open({

content: '任务处理成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

console.log(response,row.taskid)

rows.splice(index, 1)

}else{

this.$notify.open({

content: '请求失败！！稍后尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

})

})

}else{

this.$notify.open({

content: '请求失败！！稍后尝试',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

})

}

}

}

</script>

<style>

</style>

* + 1. Login

<template>

<div style="position: absolute;top: 0;left: 0;right: 0;bottom: 0;background: #232332;" >

<div style="width: 500px;background: white;border-radius: 10px;margin:200px auto;">

<div

style="height: 100px;background: #13A89E;border-radius: 10 0;color: white;font-size: 30px;text-align: center;line-height: 100px;">

<i class="fa fa-twitch" style="line-height: 100px;font-size: 40px;"></i>

寝室报修管理系统

</div>

<div style="padding: 40px;">

<label class="label" style="text-align: left;">账号：</label>

<p class="control has-icon">

<input

class="input is-medium is-success"

type="tel"

placeholder="请输入账号"

v-model="form.tel">

<i class="fa fa-user-circle-o"></i>

</p>

<label class="label" style="text-align: left;">密码：</label>

<p class="control has-icon">

<input

class="input is-rounded is-medium is-success"

type="password"

placeholder="请输入密码"

v-model="form.password">

<i class="fa fa-lock"></i>

</p>

<div>

<radio-group v-model="form.flag">

<radio val="1">学生</radio>

<radio val="2">宿管</radio>

<radio val="3">维修</radio>

<radio val="4">管理员</radio>

</radio-group>

</div>

<div class="columns" style="background: white;">

<div class="column" style="background: white;">

<p class="control notification" style="text-align: right;background: white;">

<button

class="button is-success is-large"

@click="loginClick">

登录

</button>

</p>

</div>

<div class="column">

<p class="control notification has-text-centered" style="text-align: left;background: white;">

<button class="button is-large" @click="registerClick">

注册

</button>

</p>

</div>

</div>

</div>

</div>

</div>

</template>

<script>

import {setCookie} from '@/common/config/cookie'

import {delCookie} from '@/common/config/cookie'

export default {

data() {

return {

form:{

tel:'',

password:'',

flag:''

},

user:[],

};

},

methods: {

loginClick(){

if(this.form.tel!=''&&this.form.password!=''&&this.form.flag!=''){

if(this.form.flag==1){

let that = this

this.$api.user.userLogin(this.form.tel,this.form.password,this.form.flag).then(response=>{

console.log(response);

if(that.form.password==response.studentpassword){

this.user.push(response)

let uid = localStorage.setItem("uid", that.form.tel)

let flag = localStorage.setItem("flag", that.form.flag)

let name = localStorage.setItem("name", response.studentname)

that.$store.commit('changeLogin','100')

setCookie("user",that.form.tel,2)

setCookie("password",that.form.password,2)

that.$notify.open({

content: '登录成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

this.$router.push("/home")

} else{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

})

.catch(err=>{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

})

}

if(this.form.flag==2){

let that = this

this.$api.user.userLogin(this.form.tel,this.form.password,this.form.flag).then(response=>{

console.log(response);

this.user.push(response)

if(that.form.password==response.housemanpassword){

console.log(response);

let uid = localStorage.setItem("uid", that.form.tel)

let flag = localStorage.setItem("flag", that.form.flag)

let name = localStorage.setItem("name", response.housemanname)

that.$store.commit('changeLogin','100')

that.$notify.open({

content: '登录成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

this.$router.push("/home")

}else{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

})

.catch(err=>{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

})

}

if(this.form.flag==3){

let that = this

this.$api.user.userLogin(this.form.tel,this.form.password,this.form.flag).then(response=>{

console.log(response);

if(that.form.password==response.updatmanpassword){

console.log(response);

this.user.push(response)

that.$store.commit('changeLogin','100')

let uid = localStorage.setItem("uid", that.form.tel)

let flag = localStorage.setItem("flag", that.form.flag)

let name = localStorage.setItem("name", response.updatmanname)

that.$notify.open({

content: '登录成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

this.$router.push("/home")

} else{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

})

.catch(err=>{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

})

}

if(this.form.flag==4){

let that = this

this.$api.user.userLogin(this.form.tel,this.form.password,this.form.flag).then(response=>{

if(that.form.password==response.superpwd){

this.user.push(response)

if(response!=null){

this.$store.commit('changeLogin','100')

let uid = localStorage.setItem("uid", that.form.tel)

let flag = localStorage.setItem("flag", that.form.flag)

let name = localStorage.setItem("name", response.supername)

setCookie("user",that.form.tel,2)

setCookie("password",that.form.password,2)

that.$notify.open({

content: '登录成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

that.$router.push("/home")

}else{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

}else{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

})

.catch(err=>{

this.$notify.open({

content: '登录失败，请输入正确密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

})

}

}else{

this.$notify.open({

content: '登录失败，请输入正确账号密码',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

},

registerClick(){

this.$router.push("/user/register")

}

}

}

</script>

<style scoped>

.bg{

background-position: center;

}

body{

overflow: hidden;

}

</style>

* + 1. Register

<template>

<div style="position: absolute;top: 0;left: 0;right: 0;bottom: 0;background: #232332;overflow: auto;" >

<div style="width: 500px;background: white;border-radius: 10px;margin:200px auto;">

<div

style="height: 100px;background: #13A89E;border-radius: 10 0;color: white;font-size: 30px;text-align: center;line-height: 100px;">

用户注册

</div>

<div style="padding: 40px;">

<div>

<radio-group v-model="flag">

<radio val="1">学生</radio>

<radio val="2">宿管</radio>

<radio val="3">维修</radio>

</radio-group>

</div>

<label class="label" style="text-align: left;">账号：</label>

<p class="control">

<input class="input is-success is-medium" type="number" v-model="form.number">

</p>

<label class="label" style="text-align: left;">姓名：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.name">

</p>

<label class="label" style="text-align: left;">性别：</label>

<p class="control has-icon has-icon-right">

<input class="input is-success is-medium" type="text" v-model="form.sex">

</p>

<label class="label" style="text-align: left;">电话：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="tel" v-model="form.tel">

</p>

<div v-show="show\_update">

<label class="label" style="text-align: left;">年龄：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="number" v-model="form.age">

</p>

</div>

<div v-show="show\_house">

<label class="label" style="text-align: left;">楼栋号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="number" v-model="form.value1">

</p>

</div>

<div v-show="show\_student">

<label class="label" style="text-align: left;">楼栋号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.value1">

</p>

</div>

<div v-show="show\_student">

<label class="label" style="text-align: left;">寝室号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.value2">

</p>

</div>

<div v-show="show\_student">

<label class="label" style="text-align: left;">班级：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.classname">

</p>

</div>

<label class="label" style="text-align: left;">qq号：</label>

<p class="control has-icon has-icon-right ">

<input class="input is-success is-medium" type="text" v-model="form.qq">

</p>

<label class="label" style="text-align: left;">密码：</label>

<p class="control has-icon">

<input

class="input is-rounded is-medium is-success"

type="password"

placeholder="请输入密码"

v-model="form.password">

<i class="fa fa-lock"></i>

</p>

<label class="label" style="text-align: left;">确认密码：</label>

<p class="control has-icon">

<input

class="input is-rounded is-medium is-success"

type="password"

placeholder="请输入密码"

v-model="form.newpassword">

<i class="fa fa-lock"></i>

<div class="columns" style="background: white;">

<div class="column" style="background: white;">

<p class="control notification" style="text-align: right;background: white;">

<button

class="button is-success is-large"

@click="registerClick">

注册

</button>

</p>

</div>

<div class="column">

<p class="control notification has-text-centered" style="text-align: left;background: white;">

<button class="button is-large" @click="backLogin">

返回登录

</button>

</p>

</div>

</div>

</div>

</div>

</div>

</template>

<script>

export default {

data() {

return {

show\_student:true,

show\_house:false,

show\_update:false,

options: [{

value: 15,

label: 15

}],

form:{

number:'',

name:'',

sex:'',

tel:'',

value1: '',

value2: '',

classname:'',

qq:'',

password:'',

newpassword:'',

},

flag:'1',

};

},

watch: {

flag:{

handler(){

let flags = this.flag

this.flagChange(flags)

console.log(this.flag)

}

}

},

mounted(){

},

methods: {

registerClick(){

if(this.form.number!=''

&&this.form.name!=''

&&this.form.sex!=''

&&this.form.tel!=''

&&this.form.qq!=''

&&this.form.password!=''

&&this.form.newpassword!=''){

if(this.form.password==this.form.newpassword){

if(this.flag==1){

this.$api.user.studentRegistre(

this.form.number,

this.form.name,

this.form.tel,

this.form.sex,

this.form.value1,

this.form.value2,

this.form.classname,

this.form.qq,

this.form.password,

this.flag

).then(response=>{

console.log(response);

if(response){

this.$notify.open({

content: '学生注册成功',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '注册失败',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

});

}

if(this.flag==2){

this.$api.user.houseRegistre(

this.form.number,

this.form.name,

this.form.tel,

this.form.sex,

this.form.value1,

this.form.qq,

this.form.password,

this.flag

).then(response=>{

console.log(response);

if(response){

this.$notify.open({

content: '宿管员注册成功！等待管理员确认',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '注册失败',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

});

}

if(this.flag==3){

this.$api.user.updatRegistre(

this.form.number,

this.form.name,

this.form.tel,

this.form.sex,

this.form.qq,

this.form.password,

this.flag

).then(response=>{

console.log(response);

if(response){

this.$notify.open({

content: '维修员注册成功！等待管理员确认',

icon: 'smile-o',

placement: 'top-center',

transition: 'bounce',

type: 'success',

});

}else{

this.$notify.open({

content: '注册失败',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

});

}

}else{

this.$notify.open({

content: '密码不一致',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

this.form.newpassword=''

}

}else{

this.$notify.open({

content: '信息不能为空',

placement: 'top-center',

transition: 'bounce',

type: 'danger',

});

}

},

backLogin(){

this.$router.push("/")

},

flagChange(flags){

if(flags==1){

this.show\_student=true

this.show\_house=false

this.show\_update=false

}

if(flags==2){

this.show\_house=true

this.show\_update=true

this.show\_student=false

}

if(flags==3){

this.show\_update=true

this.show\_student=false

this.show\_house=false

}

}

}

}

</script>

<style scoped>

.bg{

background-position: center;

}

body{

overflow: hidden;

}

</style>

## 页面路由

export default[

{

path: '/home',

name: 'Home',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/index')), 'home'),

children: [ //这里就是二级路由的配置

{

path: '/home',

name: 'Home',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/home')), 'Home'),

},

{

path: '/selfmsg',

name: 'Selfmsg',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/selfmsg')), 'Selfmsg'),

},

{

path: '/share',

name: 'Share',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/share')), 'Share'),

},

{

path: '/fix',

name: 'Fix',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/fix')), 'Fix'),

},

{

path: '/assess',

name: 'Assess',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/assess')), 'Assess'),

},

{

path: '/notice',

name: 'Notice',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/notice')), 'Notice'),

},

//管理员

{

path: '/student',

name: 'Student',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/student')), 'Student'),

},

{

path: '/houseman',

name: 'Houseman',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/houseman')), 'Houseman'),

},

{

path: '/updatman',

name: 'Updatman',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/updatman')), 'Updatman'),

},

{

path: '/newmsg',

name: 'Newmsg',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/newmsg')), 'Newmsg'),

},

{

path: '/complain',

name: 'Complain',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/complain')), 'Complain'),

},

{

path: '/msg',

name: 'Msg',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/msg')), 'Msg'),

},

{

path: '/task',

name: 'Task',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/main/task')), 'Task'),

},

{

path: '/house',

name: 'House',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/house')), 'House'),

},

{

path: '/allottask',

name: 'Allottask',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/allottask')), 'Allottask'),

},

{

path: '/power',

name: 'Power',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/power')), 'Power'),

},

{

path: '/reply',

name: 'Reply',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/reply')), 'Reply'),

},

{

path: '/service',

name: 'Service',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/service')), 'Service'),

},

{

path: '/resource',

name: 'Resource',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/admin/resource')), 'Resource'),

},

{

path: '/user/alter',

name: 'Alter',

meta:{auth:true},

component: r => require.ensure([], () => r(require('@/pages/user/alter')), 'Alter')

}

]

},

]

export default [

{

path: '/',

name: 'Login',

component: r => require.ensure([], () => r(require('@/pages/user/index')), 'Login'),

},

{

path: '/user/register',

name: 'Register',

component: r => require.ensure([], () => r(require('@/pages/user/register')), 'Register')

},

]

## 页面接口信息

import {fetch} from '@/common/config/fetch'

//admin申请表

/\*

\*

\*

\* \*/

export function findall(){

return fetch('/findall','GET')

}

/\*\*

申请同意：

请求地址：/agree

请求参数:

申请信息id :applyid

返回值：成功返回true，失败返回false;

\*/

export function agree(applyid){

return fetch('/agree',{applyid:applyid},'GET')

}

/\*\*

根据id删除申请信息:

请求地址：/deletebyid

请求参数：

申请信息id :applyid

返回值:

成功返回true，失败返回false

\*/

export function deletebyid(applyid){

return fetch('/deletebyid',{applyid:applyid},'GET')

}

/\*\*

修改申请信息

请求地址：/updateApply

请求参数:

申请信息类

返回值：成功返回true，失败返回false;

\*/

export function updateApply(applyid,applydesc,createtime,flag,statu){

return fetch('/updateApply',{applyid:applyid,applydesc:applydesc,createtime:createtime,flag:flag,statu:statu},'GET')

}

/\*\*

查询所有用户信息：

请求地址：/findbyuserall

请求参数：

flag为1时查询学生

flag为2时查询宿管员

flag为3时查询维修员

返回值：学生，宿管员，维修员实体类集合

\*/

export function findbyuserall(flag){

return fetch('/findbyuserall',{flag:flag},'GET')

}

/\*\*

修改：

请求地址：/update

请求参数：

如果标志为1，参数为学生信息的所有字段。

如果标志为2，参数为宿管员信息的所有字段。

如果标志为3，参数为管理员信息的所有字段。

返回值：

成功返回true,失败返回false

\*/

export function update(flag,

studentclass,

studentcode,

studenthouse,

studentname,

studentnum,

studentpassword,

studentqq,

studentsex,

studenttel){

return fetch('/update',{

flag:flag,

studentclass:studentclass,

studentcode:studentcode,

studenthouse:studenthouse,

studentname:studentname,

studentnum:studentnum,

studentpassword:studentpassword,

studentqq:studentqq,

studentsex:studentsex,

studenttel:studenttel

},'GET')

}

/\*\*

删除：

请求地址：/delete

如果标志为1，参数为学生账号。

如果标志为2，参数为宿管员账号。

如果标志为3，参数为管理员账号。

返回值：

成功返回true,失败返回false

\*/

export function deleteMsg(flag,studentnum,housemannum,updatmannum){

return fetch('/delete',{flag:flag,studentnum:studentnum,housemannum:housemannum,updatmannum:updatmannum},'GET')

}

//宿管接口

/\*\*

修改：

请求地址：/update

请求参数：

如果标志为1，参数为学生信息的所有字段。

如果标志为2，参数为宿管员信息的所有字段。

如果标志为3，参数为管理员信息的所有字段。

返回值：

成功返回true,失败返回false

\*/

export function updateHouse(flag,

housemanname,

housemannum,

housemancode,

housemanpassword,

housemanqq,

housemansex,

housemantel){

return fetch('/update',{

flag:flag,

housemanname:housemanname,

housemannum:housemannum,

housemancode:housemancode,

housemanpassword:housemanpassword,

housemanqq:housemanqq,

housemansex:housemansex,

housemantel:housemantel

},'GET')

}

//维修接口

/\*\*

修改：

请求地址：/update

请求参数：

如果标志为1，参数为学生信息的所有字段。

如果标志为2，参数为宿管员信息的所有字段。

如果标志为3，参数为管理员信息的所有字段。

返回值：

成功返回true,失败返回false

\*/

export function updateUpdate(flag,

updatmanname,

updatmannum,

updatmanqq,

updatmanpassword,

updatmansex,

updatmantel){

return fetch('/update',{

flag:flag,

updatmanname:updatmanname,

updatmannum:updatmannum,

updatmanqq:updatmanqq,

updatmanpassword:updatmanpassword,

updatmansex:updatmansex,

updatmantel:updatmantel

},'GET')

}

/\*\*

查看未处理的报修任务：（这是管理员查看，不是维修人员的那个任务，那是另一个表）

请求地址：/findfixbystaut

请求参数：无

返回值：维修表类集合

\*/

export function findfixbystaut(){

return fetch('/findfixbystaut','GET')

}

/\*\*

查询所有维修任务：

请求地址：/findtaskall

请求参数：无

返回值：维修任务集合类，失败返回空

\*/

export function findtaskall(){

return fetch('/findtaskall','GET')

}

/\*\*

改变报修任务状态：

请求地址：/updatefix

请求参数：

报修任务id：fixid

保修任务状态：fixflag(0是未受理，1是已受理，2是已处理)

返回值：成功返回true,失败返回false

\*/

export function updatefix(fixid,fixflag){

return fetch('/updatefix',{fixid:fixid,fixflag:fixflag},'GET')

}

/\*\*

删除报修任务：

请求地址：/deletefix

请求参数：

维修任务id：fixid

返回值：成功返回true,失败返回false

\*/

export function deletefix(fixid){

return fetch('/deletefix',{fixid:fixid},'GET')

}

/\*\*

查看资源：

请求地址：/seefile

请求参数：无

返回值：资源类集合

\*/

export function seefile(){

return fetch('/seefile','GET')

}/\*\*

删除资源：

请求地址：/deletefile

请求参数：

资源id：shareid

返回值：成功返回true,失败返回false

\*/

export function deletefile(shareid){

return fetch('/deletefile',{shareid:shareid},'GET')

}

/\*\*

维修任务分配：

请求地址：/addtask

请求参数：

任务描述：taskdesc（也就是报修描述里得内容）

任务处理人：taskman（通过查询维修人来选择得其中一个）

任务地点：taskaddress（将报修得寝室号和楼栋号合起来）

任务处理时间：tasktime

任务优先级：taskfirst（0表示预约，1表示紧急报修任务）

返回值：成功返回true,失败返回false;

\*/

export function addtask(taskdesc,taskman,taskaddress,tasktime,taskfirst){

return fetch('/addtask',{

taskdesc:taskdesc,

taskman:taskman,

taskaddress:taskaddress,

tasktime:tasktime,

taskfirst:taskfirst

},'GET')

}

//投诉

/\*\*

查询所有投诉信息：

请求地址：/finfcomall

请求参数：无

返回值：投诉信息类集合

\*/

export function finfcomall(){

return fetch('/finfcomall','GET')

}

/\*\*

删除投诉信息：

请求地址：/deletecom

请求参数：

complaindescid：投诉信息id

返回值：成功返回true,失败返回false;

\*/

export function deletecom(complaindescid){

return fetch('/deletecom',{complaindescid:complaindescid},'GET')

}

/\*\*

添加投诉反馈信息：

请求地址：/insertfeedback

请求参数：

投诉id：complaindescid

反馈描述：feedbackdesc

反馈人账号：number

反馈人姓名：username

返回值：成功返回true,失败返回false;

\*/

export function insertfeedback(complaindescid,feedbackdesc,number,username){

return fetch('/insertfeedback',{

complaindescid:complaindescid,

feedbackdesc:feedbackdesc,

number:number,

username:username

},'GET')

}

import {fetch} from '@/common/config/fetch'

//user登录接口

/\*

\*

\*

\* \*/

export function userLogin(number,password,flag){

return fetch('/login',{number:number,password:password,flag:flag},'GET')

}

/\*\*

\* 学生注册

\* number:'',

name:'',

sex:'',

tel:'',

value1: '',

value2: '',

classname:'',

qq:'',

password:'',

newpassword:'',

\*/

export function studentRegistre(studentnum,studentname,studenttel,studentsex,studentcode,studenthouse,studentclass,studentqq,studentpassword,flag){

return fetch('/registre',{

studentnum:studentnum,

studentname:studentname,

studenttel:studenttel,

studentsex:studentsex,

studentcode:studentcode,

studenthouse:studenthouse,

studentclass:studentclass,

studentqq:studentqq,

studentpassword:studentpassword,

flag:flag},'GET')

}

/\*\*

\* 宿管注册

\* number:'',

name:'',

sex:'',

tel:'',

value1: '',

value2: '',

classname:'',

qq:'',

password:'',

newpassword:'',

\*/

export function houseRegistre(housemannum,housemanname,housemantel,housemansex,housemancode,housemanqq,housemanpassword,flag){

return fetch('/registre',{

housemannum:housemannum,

housemanname:housemanname,

housemantel:housemantel,

housemansex:housemansex,

housemancode:housemancode,

housemanqq:housemanqq,

housemanpassword:housemanpassword,

flag:flag},'GET')

}

/\*\*

\* 维修注册

\* number:'',

name:'',

sex:'',

tel:'',

value1: '',

value2: '',

classname:'',

qq:'',

password:'',

newpassword:'',

\*/

export function updatRegistre(updatmannum,updatmanname,updatmantel,updatmansex,updatmanqq,updatmanpassword,flag){

return fetch('/registre',{

updatmannum:updatmannum,

updatmanname:updatmanname,

updatmantel:updatmantel,

updatmansex:updatmansex,

updatmanqq:updatmanqq,

updatmanpassword:updatmanpassword,

flag:flag},'GET')

}

/\*\*

修改：

请求地址：/update

请求参数：

如果标志为1，参数为学生信息的所有字段。

如果标志为2，参数为宿管员信息的所有字段。

如果标志为3，参数为管理员信息的所有字段。

返回值：

成功返回true,失败返回false

\*/

export function studentupdate(

flag,

studentnum,

studentname,

studenttel,

studentsex,

studenthouse,

studentcode,

studentclass,

studentqq,

studentpassword){

return fetch('/update',{

flag:flag,

studentnum:studentnum,

studentname:studentname,

studenttel:studenttel,

studentsex:studentsex,

studenthouse:studenthouse,

studentcode:studentcode,

studentclass:studentclass,

studentqq:studentqq,

studentpassword:studentpassword},'GET')

}

export function updatupdate(

flag,

updatmanname,

updatmannum,

updatmanpassword,

updatmanqq,

updatmansex,

updatmantel){

return fetch('/update',{

flag:flag,

updatmanname:updatmanname,

updatmannum:updatmannum,

updatmanpassword:updatmanpassword,

updatmanqq:updatmanqq,

updatmansex:updatmansex,

updatmantel:updatmantel},'GET')

}

/\*\*

\* 查询用户数据信息

\* \*/

export function findbyuserid(updatmannum,studentnum,housemannum,flag){

return fetch('/findbyuserid',{updatmannum:updatmannum,studentnum:studentnum,housemannum:housemannum,flag:flag},'GET')

}

/\*\*

资源上传：

请求地址:/uploadfile

请求参数：

学生学号：studentnum

学生姓名：studentname

文件类型：sharetype

返回值：成功返回true,失败返回false

\*/

export function uploadfile(studentnum,studentname,sharetype){

return fetch('/uploadfile',{studentnum:studentnum,studentname:studentname,sharetype:sharetype},'GET')

}

/\*\*

添加报修任务：

请求地址：/insertfix

请求参数：

维修表的全部字段（除fixid和fiximage以外）

返回值：成功返回true,失败返回false

\*/

export function insertfix(

studentnum,

studentname,

fixtime,

studentcode,

studenthouse,

fixdesc,

fixflag){

return fetch('/insertfix',{

studentnum:studentnum,

studentname:studentname,

fixtime:fixtime,

studentcode:studentcode,

studenthouse:studenthouse,

fixdesc:fixdesc,

fixflag:fixflag

},'GET')

}

/\*\*

查询所有维修任务：

请求地址：/findtaskall

请求参数：无

返回值：维修任务集合类，失败返回空

\*/

export function findtaskall(){

return fetch('/findtaskall','GET')

}

/\*\*

修改维修任务状态：（此接口用于维修员，当维修员完成任务后就要将此任务标记为已处理）

请求地址：/updatetaskstaut

请求参数：

任务状态 ：taskflag（0表示未处理，1表示已处理）

维修任务id：taskid

返回值：成功返回true,失败返回false;

\*/

export function updatetaskstaut(taskid,taskflag){

return fetch('/updatetaskstaut',{

taskid:taskid,

taskflag:taskflag

},'GET')

}

/\*\*

根据维修任务id删除维修任务：

请求地址：/deletetask

请求参数：

维修任务id：taskid

返回值：成功返回true,失败返回false;

\*/

export function deletetask(taskid){

return fetch('/deletetask',{

taskid:taskid

},'GET')

}

//投诉模块

/\*\*

添加投诉信息：

请求地址：/insertcom

请求参数：

studentnum：学号

number：被投诉人账号

username：背投诉人姓名

studentname：名字

complaindesc：投诉描述

返回值：成功返回true,失败返回false;

\*/

export function insertcom(studentnum,studentname,number,username,complaindesc){

return fetch('/insertcom',{

studentnum:studentnum,

studentname:studentname,

number:number,

username:username,

complaindesc:complaindesc

},'GET')

}

//评价模块

/\*\*

添加评价：

请求地址：/insertassess

请求参数：

studentnum：学号

studentname：名字

assessdesc：评价描述

assesscode：评价星级

taskid：评价任务id

返回值：成功返回true,失败返回false;

\*/

export function insertassess(studentnum,studentname,taskid,assesscode,assessdesc){

return fetch('/insertassess',{

studentnum:studentnum,

studentname:studentname,

taskid:taskid,

assesscode:assesscode,

assessdesc:assessdesc

},'GET')

}

/\*\*

查询所有评价：

请求地址：/findassessall

请求参数：无

返回值：评价信息类集合

\*/

export function findassessall(){

return fetch('/findassessall','GET')

}

//通知

/\*\*

发布通知：

请求地址：/releasenotice

请求参数：

通知实体类（除id，发布时间）

返回值：成功返回true,失败返回false;

\*/

export function releasenotice(

housemannum,

housemanname,

noticedesc,

noticehose

){

return fetch('/releasenotice',{

housemannum:housemannum,

housemanname:housemanname,

noticedesc:noticedesc,

noticehose:noticehose

},'GET')

}

/\*\*

查询所有通知：

请求地址：/findallnotice

请求参数：无

返回值：成功返回通知类集合，失败返回null

\*/

export function findallnotice(){

return fetch('/findallnotice','GET')

}

/\*\*

根据投诉id查询投诉反馈信息：

请求地址：/findcomplainId

请求参数：

投诉id：complaindescid(改为通过账号获取number)

返回值：投诉表信息集合

\*/

export function findcomplainId(complaindescid){

return fetch('/findcomplainId',{complaindescid:complaindescid},'GET')

}