**报修模块设计文档**

1. 报修模块概述

报修模块的功能主要包括：用户进行报修申请，管理员对报修申请进行审核和管理。

1. 流程图

（一）报修申请流程图

（二）报修申请审核流程图

用户

点击报修申请

填写报修表

提交

2.1同意报修申请流程图

管理员

点击报修审批

选择审核中的报修申请记录

点击同意申请

记录状态变为14（维修中）

2.2拒绝报修申请流程图

管理员

点击报修审批

选择审核中的报修申请记录

点击拒绝申请

记录状态变为12（申请失败）

2.3维修完成流程图

管理员

点击报修审批

选择维修中的报修申请记录

点击维修完成

记录状态变为15（维修完成）

1. 报修申请撤回流程图

用户

点击我的报修

点击撤回处于申请中的记录

确认

记录状态变为11（失效）

（四）报修管理流程图

4.1报修信息删除流程图

管理员

点击报修管理

选择报修记录

点击删除按钮

删除成功

4.2报修增加流程图

管理员

点击报修管理

点击增加按钮

填写报修信息

确认提交

增加成功

4.3报修信息导出流程图

管理员

点击导出按钮

选择相关信息

确认

导出成功

完毕

**三．表的创建**

1.报修申请表(maintainapply)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type |
| Id(编号) | p |  |  | Int（自增） |
| pro\_name  (项目名称) |  | Not null |  | varchar |
| bid  (基地名称) |  | Not null |  | varchar |
| username  (报修人姓名) |  | Not null |  | varchar |
| address  (维修位置) |  | Not null |  | varchar |
| college(开课学院) |  | Not null |  | varchar |
| reason  (报修原因) |  | Not null |  | varchar |
| file(材料路径) |  | null |  | varchar |
| money(预算经费) |  | null |  | varchar |
| status(状态值) |  | null | 13 | int |
| applytime(申报时间) |  | null |  | varchar |
| userid(申请人编号) |  | null |  | varchar |
| actualmoney(专业编号) |  | null | 0 | double |
| refuse(拒绝原因) |  | null |  | varchar |

1. 后台接口设计

本系统采用SSH框架实现，一共涉及四层分布：

controller层:控制层，主要实现页面的请求与响应,控制各个页面之间的跳转 ;

service层：业务逻辑层，主要实现代码内部逻辑功能；

dao层：数据库连接层，主要与数据库进行直接交互连接；

po层：实体层，每一个实体类对应数据库中的一张表。

下述为针对该模块所设计的action层，service层及对应dao层的相关接口：

Controller层: 1.报修申请控制：MaintainApplyController

2.我的报修控制：MymaintainController

3.报修申请审核控制：RepairApproveController

Service层： 1.报修申请逻辑：MaintainApplyService

2.我的报修逻辑：MymaintainService

3.报修申请审核逻辑：RepairApproveService

Dao层: 1.保修申请数据交互：MaintainApplyDao

2.我的报修数据交互：MymaintainDao

3.报修申请数据交互：RepairApproveDao

**Controller层：**

//报修申请控制层

@Controller("MaintainApplyController")

**public** **class** MaintainApplyController {

// 校内基地名查询(包括土地基地和校内基地)

@RequestMapping("/baseNeiName.do")

**public** String baseNeiName(HttpServletRequest request,

HttpServletResponse response);

// 提交维修申请

@RequestMapping("/insertmaintain.do")

**public** String insert\_maintain(HttpServletRequest request,

HttpServletResponse response);

// 获取维修信息

@RequestMapping("/query\_maintainapply.do")

**public** String query\_maintainapply(HttpServletRequest request,

HttpServletResponse response);

// 删除维修信息记录

@RequestMapping("/delmaintainapply.do")

**public** String delmaintainapply(HttpServletRequest request,

HttpServletResponse response);

// 增加维修信息记录

@RequestMapping("/addmaintainapply.do")

**public** String add\_maintainhistory(HttpServletRequest request,

HttpServletResponse response);

//获取年份

@RequestMapping("/getThoseYear.do")

**public** String getThoseYear(HttpServletRequest request,

HttpServletResponse response);

// 导出维修信息记录，参数为筛选条件，第一个基地名字，第二个为年份(如没有，则为-1)

@RequestMapping("/exportmaintainapply.do")

**public** String export\_maintainapply(HttpServletRequest request,

HttpServletResponse response);

}

//我的报修控制层

@Controller("MymaintainController")

**public** **class** MymaintainController {

// 获取该用户当年的报修申请记录

@RequestMapping("/Mymaintain.do")

**public** String Mymaintain(HttpServletRequest request,

HttpServletResponse response);

// 获取该用户所有的报修申请记录

@RequestMapping("/Mymaintain2.do")

**public** String Mymaintain2(HttpServletRequest request,

HttpServletResponse response);

// 根据状态值筛选，获取报修信息记录

@RequestMapping("/screen.do")

**public** String screen(HttpServletRequest request,

HttpServletResponse response);

// 撤回申请

@RequestMapping("/recallmymaint.do")

**public** String recallmymaint(HttpServletRequest request,

HttpServletResponse response);

}

//报修申请审核控制层

@Controller("repairApproveController")

**public** **class** RepairApproveController {

//获取审核中的报修申请记录

@RequestMapping("/getNoRepair.do")

**public** String getNoRepair(HttpServletRequest request,

HttpServletResponse response);

//获取维修中的报修申请记录

@RequestMapping("/getAgreeRepair.do")

**public** String getAgreeRepair(HttpServletRequest request,

HttpServletResponse response);

//根据刷选条件获取审核中的报修申请记录

@RequestMapping("/getChooseRepair1.do")

**public** String getChooseRepair1(HttpServletRequest request,

HttpServletResponse response);

//根据刷选条件获取维修中的报修申请记录

@RequestMapping("/getChooseRepair2.do")

**public** String getChooseRepair2(HttpServletRequest request,

HttpServletResponse response);

//获取基地名称和申报人集合

@RequestMapping("/getInfoApply.do")

**public** String getInfoApply(HttpServletRequest request,

HttpServletResponse response);

//拒绝报修申请

@RequestMapping("/refuseRepairApply.do")

**public** String refuseRepairApply(HttpServletRequest request,

HttpServletResponse response);

//同意报修申请报修申请

@RequestMapping("/agreeRepairApply.do")

**public** String agreeRepairApply(HttpServletRequest request,

HttpServletResponse response);

//维修完成

@RequestMapping("/repairFinish.do")

**public** String repairFinish(HttpServletRequest request,

HttpServletResponse response);

}

**Service层：**

//报修申请逻辑层

**public** **interface** MaintainApplyService {

/\*

参数说明：str，为报修申请信息的封装;infostr，字符串型，为要发送的信息

返回值： 无返回值

函数功能：增加报修申请信息记录，并发送信息给申请人

\*/

**public** **void** insert\_maintain(String str,String infostr);

/\*

参数说明：page,为当前页数;offsets,为每页的条数; order,排序列;

str1,为排序的顺序; str2,为模糊查询的值.

返回值： MaintainList,包括报修信息记录的集合和记录条数

函数功能：获取报修信息记录

\*/

**public** MaintainList query\_maintainapply(**int** offsets,**int** page,**int** order,String str1,String str2);

/\*

参数说明：str,为要删除的报修信息记录的编号的封装

返回值： 无返回值

函数功能：删除报修信息

\*/

**public** **void** delete\_maintainapply(String str);

/\*

参数说明：ma，为MaintainApply对象

返回值： 无返回值

函数功能：增加报修信息记录

\*/

**public** **void** insert\_maintainhistory(MaintainApply ma);

/\*

参数说明：bname,字符串型，为基地名称;years,int型，为年份

返回值： List<MaintainApplys>，为MaintainApplys对象的集合

函数功能：获取导出的报修信息数据

\*/

**public** List<MaintainApplys> export\_maintainapply(String bname,**int** years);

/\*

参数说明：无参数值

返回值： List<String>，为年份的集合

函数功能：获取报修信息记录存在的年份

\*/

**public** List<String> getThoseYear();

/\*

参数说明：无参数

返回值： List<Map<String,String>>，为基地名字的集合

函数功能：获取校内基地名称

\*/

**public** List<Map<String,String>> find\_basenamenei();

}

//我的报修逻辑层

**public** **interface** MymaintainService {

/\*

参数说明：year,整型，为年份;status,整型，为状态值;userid,字符串型，为用户编号; pageindex,为当前页数;size,为每页的条数; columnName,排序列名;

orderDir,为排序的顺序

返回值： MymaintainList,该对象包括报修申请数据记录+记录条数

函数功能： 获取该用户当年的报修申请记录

\*/

**public** MymaintainList Mymaintain(**int** pageindex,**int** size,String columnName,String orderDir,**int** year,**int** status,String userid);

/\*

参数说明：year,整型，为年份;status,整型，为状态值;userid,字符串型，为用户编号; pageindex,为当前页数;size,为每页的条数; columnName,排序列名;

orderDir,为排序的顺序

返回值：MymaintainList,该对象包括报修申请数据记录+记录条数

函数功能： 获取该用户所有的报修申请记录

\*/

**public** MymaintainList Mymaintain2(**int** pageindex,**int** size,**int** order, String orderDir,**int** year,**int** status,String userid);

/\*

参数说明：id,为报修申请记录的编号;infostr，字符串型，为要发送的信息

返回值： 无返回值

函数功能： 撤回报修申请

\*/

**public** **void** recallmymaint(String id,String infostr);

}

//保修申请审核逻辑

**public** **interface** RepairApproveService {

/\*

参数说明：baseid,字符串型，为基地名称; userid,字符串型,为用户名字

返回值： MaintainList,该对象包括报修申请数据记录+记录条数

函数功能： 获取报修申请数据

\*/

**public** MaintainList getRepairInfo(String baseid,String userid,**int** pageIndex,**int** size,**int** order,String orderDir,String searchValue,**int** status);

/\*

参数说明：refusestr，为保修申请记录的编号的封装;infostr，字符串型，为要发送的信息

返回值： 无返回值

函数功能： 拒绝保修申请

\*/

**public** **void** refuseRepairApply(String refusestr,String infostr);

/\*

参数说明：agreestr，为保修申请记录的编号的封装;infostr，字符串型，为要发送的信息

返回值： 无返回值

函数功能：同意保修申请

\*/

**public** **void** agreeRepairApply(String agreestr,String infostr);

/\*

参数说明：无参数值

返回值： List，为基地名字和申报部门的集合

函数功能：获取基地名字和申报部门的集合

\*/

**public** List getInfoApply();

/\*

参数说明：storestr，为保修申请记录的编号的封装;infostr，字符串型，为要发送的信息

返回值： 无返回值

函数功能：维修完成

\*/

**public** **void** finishRepairApply(String storestr,String infostr);

}

**Dao层：**

//保修申请的数据交互层

**public** **interface** MaintainApplyDao {

/\*

参数说明：str，为报修申请信息的封装

返回值： 无返回值

函数功能：插入报修申请信息记录

\*/

**public** **void** insert\_maintain(String str);

/\*

参数说明：page,为当前页数;offsets,为每页的条数; order,排序列;

str1,为排序的顺序; str2,为模糊查询的值.

返回值： MaintainList,包括报修信息记录的集合和记录条数

函数功能：获取报修信息记录

\*/

**public** MaintainList query\_maintainapply(**int** offsets,**int** page,String str,String str1,String str2);

/\*

参数说明：str,为要删除的报修信息记录的编号的封装

返回值： 无返回值

函数功能：删除报修信息

\*/

**public** **void** delete\_maintainapply(String str);

/\*

参数说明：ma，为MaintainApply对象

返回值： 无返回值

函数功能：增加报修信息记录

\*/

**public** **void** add\_maintain(MaintainApply ma);

/\*

参数说明：bname,字符串型，为基地名称;years,int型，为年份

返回值： List<MaintainApplys>，为MaintainApplys对象的集合

函数功能：获取导出的报修信息数据

\*/

**public** List<MaintainApplys> export\_maintainapply(String bname,**int** years);

**public** List<Map<String,String>> find\_basenameFinish();

/\*

参数说明：无参数值

返回值： List<String>，为年份的集合

函数功能：获取报修信息记录存在的年份

\*/

**public** List<String> getThoseYear();

/\*

参数说明：无参数

返回值： List<Map<String,String>>，为基地名字的集合

函数功能：获取校内基地名称

\*/

**public** List<Map<String,String>> find\_basenamenei();

}

//我的报修数据交互层

**public** **interface** MymaintainDao {

/\*

参数说明：year,整型，为年份;status,整型，为状态值;userid,字符串型，为用户编号; pageindex,为当前页数;size,为每页的条数; columnName,排序列名;

orderDir,为排序的顺序

返回值： MymaintainList,该对象包括报修申请数据记录+记录条数

函数功能： 获取该用户当年的报修申请记录

\*/

**public** MymaintainList Mymaintain(**int** pageindex, **int** size,

String columnName, String orderDir, **int** year, **int** status,

String userid);

/\*

参数说明：id,为报修申请记录的编号;infostr，字符串型，为要发送的信息

返回值： 无返回值

函数功能： 撤回报修申请

\*/

**public** **void** recallmymaint(String id);

/\*

参数说明：sql,字符串型，为封装好的sql语句

返回值： 无返回值消息记录

\*/

**public** **void** insertMessage(String sql);

}

//报修申请审核的数据交互层

**public** **interface** RepairApproveDao {

/\*

参数说明：baseid,字符串型，为基地名称; userid,字符串型,为用户名字

返回值： MaintainList,该对象包括报修申请数据记录+记录条数

函数功能： 获取报修申请数据

\*/

**public** MaintainList getRepairInfo(String baseid, String userid, **int** pageIndex, **int** size, String orderColumn, String orderDir,

String searchValue, **int** status);

/\*

参数说明：status,整型，为标志位

返回值： List<Map<String,String>>，为用户名字的集合

函数功能： 获取用户名字

\*/

**public** List<Map<String,String>> getUser(**int** status);

/\*

参数说明：status,整型，为标志位

返回值： List<Map<String,String>>，为基地名字的集合

函数功能： 获取基地名字

\*/

**public** List<Map<String,String>> getBase(**int** status);

/\*

参数说明：recordstr,字符串型,为报修申请记录的编号;status,整型，为状态值

返回值： 无返回值为status值

\*/

**public** **void** changeStatus(String recordstr,**int** status);

/\*

参数说明：refusestr,字符串型,为报修申请记录的编号

返回值： 无返回值

函数功能： 拒绝申请

\*/

**public** **void** refuseApply(String refusestr);

/\*

参数说明：storestr,字符串型,为报修申请记录的编号

返回值： 无返回值

函数功能： 维修完成

\*/

**public** **void** finish(String storestr);

}