**土地租赁模块设计文档**

1. 土地租赁概述

土地租赁模块主要用于教师对校内6大基地中的土地进行租赁。我们负责将繁绕复杂的工序简单化，合理化，多样化。首先，用户可以通过两种方式查看相应土地的信息，如果是足不出户的用户可以选择登录网站来了解每一块土地的信息，如果是亲临现场考察的用户可以扫描土地上所标识的二维码进行查看。对于最为重要的土地租赁流程，用户按照网站提示进行租赁申请，管理员进行审核，我们将管理员的管理与教师用户的申请进行合理的划分，使双方可以进行有效的，紧密的，实时的交互。

1. 流程图

（一）总业务流程图

2

1

管理员

设置土地布局

用户

土地租赁

缴费

提交信息

审核土地申请

反馈

线下

缴费成功

租赁成功

3

4

5

6

7

8

设置土地布局

1. 分流程图

1.土地布局流程图：

2. 土地租赁流程图：

管理员

设置土地的大小

点击土地布局设置

拖动土地改变位置

点击提交

设置完成

点击插入土地

点击删除土地

土地申请

（点击主页土地租赁）

出现6大基地列表

点击选择的基地

出现该基地下所有土地布局

点击土地

出现土地信息

点击租赁

出现土地信息，个人信息，租赁时间

提交

3.用户申请交互流程图：

显示用户土地申请

用户

申请中状态

交费中状态

点击撤销申请

点击撤销

申请状态变为失效

申请状态变为未交费

点击查看申请详情

显示该申请信息

点击我的申请

4.土地申请审核流程图：

管理员

点击土地申请审核

显示所有申请记录

输入条件查询

显示符合条件的土地申请记录

点击同意租赁

教师缴费

缴费完成

审核通过

线下

1

下图对应上图中的标记1处，为土地申请记录各种状态转换的逻辑。为了流程的简单易理解，在此处我将三用户提交的租赁申请设定为A，B，C，申请的六种状态分别为：1.申请中 2.交费中 3.锁定 4.未交费 5.申请失败 6.申请成功

至交费期限

用户告知管理员放弃交费

A，B，C三用户同一租赁申请

管理员同意A的申请

A状态为交费中

B，C状态为锁定

B

A是否交费

管理员同意B的申请

A状态为未交费

B，C状态为申请中

B，C状态为申请中

管理员点击A的申请成功

A状态为申请成功

B，C状态为申请失败

A提前缴费完成

A未交费

管理员点击A的拒绝交费

管理员

查询

5.用户手机操作流程图：

用户

扫描二维码

查看土地信息

6.土地使用信息流程图：

管理员

点击土地使用管理

导出土地使用信息

输入条件查询

显示符合条件的土地使用信息

删除选定的记录

1. 工作计划制定：

点击

管理员

工作计划制定

设置申报时间，租赁时间等

开始租赁

结束租赁

1. 表的创建

1.基地信息表baseInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| bid  （基地编号） | P | Not null |  | int | 1 |
| bname  （基地名称） |  | Not null |  | varchar | “长安基地” |
| btype  （基地类型） |  | Not null |  | int | 1 |

注：基地信息表中的btype字段一共可取三个值：1.代表土地 2.代表校内

3.代表校外

2.土地信息表landInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| Lid  （土地编号） | P | Not null |  | varchar |  |
| lname  (土地名称) |  | Not null |  | Varchar | “耕园果蔬基地” |
| plantingContent  （推荐种植内容） |  | null |  | varchar | “玉米” |
| landArea  (土地面积/亩) |  | Not null |  | Int | 8 |
| Major\_oriented  (面向专业) |  | null |  | Varchar(200) | “动物科学，动物科学教育” |
| Afford  (可承担人数/人) |  | null |  | int | 70 |
| buildingArea  (建筑面积) |  | null |  | int | 70 |
| bid  (基地编号) | F | Not null |  | int | 参照基地信息表的主键 |

3.土地布局表(landLayout)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| id  (序号) | p | Not null |  | int | 1 |
| x\_axis  (x坐标) |  | Not null |  | int | 100 |
| y\_axis  (y坐标) |  | Not null |  | int | 100 |
| width  (宽度/px) |  | Not null |  | int | 20 |
| height  （高度/px） |  | Not null |  | int | 40 |
| lid  (土地编号) | F | Not null |  | varchar | 参照土地信息表的主键 |
| bid  (基地编号) | F | Not null |  | int | 参照基地信息表的主键 |

4.土地申请表(landApply)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| la\_id  （序号） | P | Not null |  | int | 1 |
| Lid  （土地编号） |  | Not null |  | varchar |  |
| 起始日期  (startTime) |  | Not null |  | date |  |
| 结束日期  (endTime) |  | Not null |  | date |  |
| applicantId  （申请人） | F | Not null |  | varchar | 参照教师表的主键 |
| Planting  (种植内容) |  | null |  | varchar |  |
| Status  （申请状态） |  | Not null |  | int | 2 |
| startPayTime  （开始交费时间） |  | null |  | date | 2017-11-21 |
| applyDept (申报部门) |  | Not null |  | int |  |

注意：土地申请表中的status字段一共有七种值供选择：1.交费中 2.申请中

3.未交费 4.锁定 5.竞争失败 6.申请成功 7.失效（用户自行取消申请）

5.土地使用表(landRentInfo)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| lr\_id  （序号） | P | Not null |  | int | 1 |
| Lid  （土地编号） |  | Not null |  | varchar |  |
| startTime  (起始日期) |  | Not null |  | date |  |
| endTime  (结束日期) |  | Not null |  | date |  |
| userId  （使用人） | F | Not null |  | varchar | 参照教师表中的主键 |
| Planting  (种植内容) |  | null |  | varchar |  |
| rentMoney  （经费） |  | Not null |  | int | 1000 |
| chargeDate  (收费日期) |  | Not null |  | date | 2017-11-20 |

1. 部门表(applydept)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| aid  (部门编号) | P | Not null |  | int | 自增 |
| Dept (部门名称) |  | Not null |  | varchar | “信科院” |
| type  (部门类型) |  | Not null |  | int | 1或2 |

注：部门表中的type，为部门类型，值为1时，表示学院，值为2时，表示行政部

1. 专业表(major)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| Mid  (专业编号) | P | Not null |  | int | 自增 |
| Mname  (专业名称) |  | Not null |  | varchar | “计算机” |
| aid  (部门编号) | F | Not null |  | Int | 参照学院表中的主键 |

8.人员信息表userInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| Id（序号） | P | Not null |  | int | 自增 |
| Name（姓名） |  | Not null |  | varchar | ”张三” |
| Sex(性别) |  | Not null |  | varchar | “男” |
| Birthdate  （出生日期) |  | null |  | date | 1989-01-12 |
| ID\_number(身份证号) |  | Not null |  | varchar | “430381989011×××” |
| Telephone  (联系电话) |  | null |  | varchar | “183××××××××” |
| Usertype  （人员类别） |  | null |  | varchar | 在职人员 |
| Dept  （部门） |  | null |  | varchar | “教务处” |
| College  (所属学院) |  | null |  | varchar | “信科院” |
| userright  (用户权限) |  | Not null |  | int | 参照权限表中的主键 |
| formerUnit  (原工作单位) |  | null |  | varchar | “湖南农业大学” |
| hukou  (户口性质) |  | null |  | varchar | “农业户口” |
| arrivetime  （来校工作时间） |  | null |  | date | “2016-09-12” |
| Worktime  (参加工作时间) |  | null |  | date | “2016-09-12” |
| contactForm  (签订合同形式) |  | null |  | varchcar | “劳务协议” |
| workingForm  (用工形式) |  | null |  | varchar | “计划内” |
| startContactTime  （合同开始时间） |  | null |  | date | 2011-09-01 |
| endContactTime  （合同结束时间） |  | null |  | date | 2011-09-01 |

9.admin(用户权限表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| id  (序号) | P | Not null |  | int | 自增 |
| Name  （管理类型） |  | Not null |  | varchar | “普通教师” |
| Upow（管理权限值） |  | Not null |  | bigint |  |

1. adminfunction页面权限值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| qid  (序号) | P | Not null |  | int | 自增 |
| pname  （页面名称） |  | Not null |  | varchar | “租赁审批” |
| value（页面权限值） |  | Not null |  | bigint |  |

1. notification(公告表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| id  (序号) | P | Not null |  | int | 自增 |
| title（标题） |  | Not null |  | varchar |  |
| message（公告内容） |  | Not null |  | varchar |  |

1. message(消息表)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type | Sample Data |
| id  (序号) | P | Not null |  | int | 自增 |
| title（标题） |  | Not null |  | varchar |  |
| content（消息内容） |  | Not null |  | varchar |  |
| time(发布时间) |  | Not null |  | varchar | “2017-03-12” |
| isRead(是否已读) |  | Not null | 0 | int | 0为未读，1为已读 |
| Userid(用户编号) |  | Not null |  | varchar |  |

1. 代码设计

本系统采用SSH框架实现，一共涉及四层分布：

action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.申请控制：LandApplyController

2.审核控制：CheckController

3.土地租赁管理控制:LandRentController

4.用户信息管理:PersonalManageController

5.公告和信息管理控制：NotificationController

Service层： 1.申请逻辑：LandApplyService

2.审核逻辑：CheckService

3.土地租赁管理逻辑:LandRentService

4.用户信息管理逻辑:UserInfoService

5. 公告和信息管理逻辑：NotificationService

Dao层: 1.土地信息数据交互：LandInfoDao

2.土地申请数据交互：LandApplyDao

3.审核数据交互:CheckViewDao

3.土地使用数据交互：LandRentInfoDao

4.用户数据交互：UserInfoDao

5.公告和信息管理数据交互：NotificationDao

Action层:

//申请模块的控制层

@Controller("landApplyController")

**public** **class** LandApplyController {

// 基地查询

@RequestMapping("/baseInfo.do")

**public** String selectBase(HttpServletRequest request, ModelMap map,

HttpServletResponse response) {

**return null;**

}

// 基地+部门查询

@RequestMapping("/getBase\_deptInfo.do")

**public** String selectBase\_deptInfo(HttpServletRequest request,ModelMap map){

**return** **null**;

}

//土地布局查询(根据基地编号获取土地布局)

@RequestMapping("/getDifferLayout.do")

**public** String getDifferLayout(HttpServletRequest request,ModelMap map)

{

**return** **null**;

}

//修改及保存土地布局信息

@RequestMapping("/updateLayout\_Info.do")

**public** String updateLayout\_Info(HttpServletRequest request,ModelMap map)

{

**return** **null**;

}

// 租赁申请时，获取土地布局+土地基本信息+土地现租赁情况+土地租赁历史

@RequestMapping("/getRentCollection.do")

**public** String getRentCollection(HttpServletRequest request,ModelMap map){

**return** **null**;

}

// 租赁申请时，提交租赁申请

@RequestMapping("/submitLandApply.do")

**public** String submitLandApply(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//获取租赁开始，结束时间，申请开始，结束时间，交费限制天数

@RequestMapping("/getLandApplyDate.do")

**public** String getLandApplyDate(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//修改或更新租赁开始时间

@RequestMapping("/updateLandApplyDate.do")

**public** String updateLandApplyDate(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//结束土地申请和租赁

@RequestMapping("/endLandApply.do")

**public** String endLandApply(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//获取用户当年的土地租赁申请

@RequestMapping("/myRentFont.do")

**public** String myRentFont(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//撤销租赁申请

@RequestMapping("/myFameCancel1.do")

**public** String myFameCancel1(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

}

//审核模块的controller类

**public** **class** CheckController {

// 查询所有未审核申请记录

@RequestMapping("/checkApplyRecord.do")

**public** String checkApply(HttpServletRequest request,

HttpServletResponse response, ModelMap map)

{

**return** **null**;

}

// 查询所有逾期记录

@RequestMapping("/overdue.do")

**public** String overdue(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

// 查询所有交费中的记录

@RequestMapping("/agApply.do")

**public** String agApply(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

// 逾期恢复

@RequestMapping("/overduerecovery.do")

**public** String overduerecovery(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//同意申请

@RequestMapping("/agreeApply.do")

**public** String agreeApply(HttpServletRequest request,ModelMap map)

{

**return** **null**;

}

// 拒绝申请

@RequestMapping("/refuseApply.do")

**public** String refuseApply(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

// 确认交费

@RequestMapping("/confirm.do")

**public** String confirm(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

// 取消交费

@RequestMapping("/cancel.do")

**public** String cancel(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

// 土地租赁详情查看

@RequestMapping("/detail.do")

**public** String Rentdetail(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

// 根据条件刷选记录

@RequestMapping("/Selet.do")

**public** String Select(HttpServletRequest request,

HttpServletResponse response){

**return** **null**; }

//使用模块的控制层

**public** **class** LandRentController {

// 土地租赁记录

@RequestMapping("/landRentInfo.do")

**public** String selectBase(HttpServletRequest request,

HttpServletResponse response) {

**return** **null**;

}

//根据刷选条件选

@RequestMapping("/submitChoose.do")

**public** String submitChoose(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//获取部门，种植内容

@RequestMapping("/getExistRentInfo.do")

**public** String getExistRentInfo(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//删除记录

@RequestMapping("/deleteLandRentInfo.do")

**public** String deleteLandRentInfo(HttpServletRequest request,

HttpServletResponse response) {

**return** **null**;

}

//导出土地租赁信息

@RequestMapping("/exportLandRentInfo.do")

**public** String exportLandRentInfo(HttpServletRequest request,

HttpServletResponse response, ModelMap map){

**return** **null**;

}

//修改土地租赁信息

@RequestMapping("/landManageUpdate.do")

**public** String landManageUpdate(HttpServletRequest request,

HttpServletResponse response) {

**return** **null**;

}

}

//用户管理的控制层

@Controller("personalManageController")

**public** **class** PersonalManageController {

// 获取用户管理人员信息

@RequestMapping("/manger.do")

**public** String manger(HttpServletRequest request,

HttpServletResponse response) {

**return** **null**;

}

// 人员信息详情查看

@RequestMapping("/Mangerdetail.do")

**public** String Mangerdetail(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

// 删除人员信息

@RequestMapping("/deInfo.do")

**public** String deleteInfo(HttpServletRequest request,

HttpServletResponse response, ModelMap map) {

**return** **null**;

}

//修改人员信息

@RequestMapping("/upInfo.do")

**public** String upInfo(HttpServletRequest request,HttpServletResponse response){

**return** **null**;

}

// 增加人员信息

@RequestMapping("/addInfo.do")

**public** String addInfo(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

// 获取身份属性和部门

@RequestMapping("/getAllInfo.do")

**public** String getAllInfo(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

// 导出功能

@RequestMapping("/exportPersonInfo.do")

**public** String exportPersonInfo(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;}

// 导入功能

@RequestMapping("/inputExcel.do")

**public** String inputExcel(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

// 筛选userInfo中的部门

@RequestMapping("/Checkdept.do")

**public** String Checkdept(HttpServletRequest request,

HttpServletResponse response){

**return** **null**;

}

}

//发布通告和信息控制层

@Controller("NotificationController")

**public** **class** NotificationController {

// 通知发布页面获取所有部门

@RequestMapping("jsp/notification.do")

**public** String notification(HttpServletRequest request, ModelMap map, HttpServletResponse response)

**return** **null**;

}

// 发布通知

@RequestMapping("jsp/saveNotification.do")

**public** String saveNotification(HttpServletRequest request, ModelMap map,HttpServletResponse response) {

**return** **null**;

}

//获取通知详情

@RequestMapping("jsp/newlist.do")

**public** String newlist(ModelMap map, HttpServletResponse response,HttpServletRequest request){

**return** **null**;

}

//消息发布

@RequestMapping("jsp/saveMessage.do")

**public** String saveMessage(HttpServletRequest request, ModelMap map,

HttpServletResponse response) {

**return** **null**;

}

//获取消息列表详情

@RequestMapping("jsp/getMessage.do")

**public** String getMessage(ModelMap map, HttpServletRequest request,

HttpServletResponse response) {

**return** **null**;

}

//获取未读消息数

@RequestMapping("jsp/getNoReadMessageCount.do")

**public** String getNoReadMessageCount(ModelMap map,

HttpServletRequest request, HttpServletResponse response) {

**return** **null**;

}

//设置消息为”已读”状态，返回“未读消息数”

@RequestMapping("jsp/setReadMessage.do")

**public** String setReadMessage(ModelMap map, HttpServletRequest request,HttpServletResponse response) {

**return** **null**;

}

//通知详情页面

@RequestMapping("jsp/newdetail.do")

**public** String newdetail(ModelMap map, HttpServletRequest request,

HttpServletResponse response) {

**return** **null**;

}

}

Service层：  
//土地申请表的业务逻辑层接口

**public** **interface** LandApplyService {

/\*

参数说明：无参数值

返回值： List<BaseInfo>,为基地信息BaseInfo对象的集合

函数功能：查询六大基地的信息

\*/

**public** List<BaseInfo> getBaseInfos();

/\*

参数说明：bid,整型，表示基地编号

返回值： List<Layout\_InfoView>,为土地信息Layout\_InfoView对象的集合

函数功能：查询土地布局信息

\*/

**public** List<Layout\_InfoView> getDifferLayout(**int** bid);

/\*

参数说明：无参数值

返回值： Startplan对象的集合

函数功能：获得申报的开始和结束时间

\*/

**public** List<Startplan> getLandApplyDate();

/\*

参数说明：Startplan对象

返回值： 无返回值

函数功能：更新申报的开始和结束时间，租赁的开始和结束时间，交费限制天数

\*/

**public** **void** updateLandApplyDate(Startplan sp);

/\*

参数说明：id,字符串型

返回值： 无返回值

函数功能：开启工作计划

\*/

**public** Startplan getStartPlan(String id)

/\*

参数说明：无参数值

返回值： 无返回值

函数功能：结束工作计划

\*/

**public** **void** endStartPlan();

/\*

参数说明：bid,基地编号；

返回值：List<RentCollection>，RentCollection对象的集合；

函数功能：获取土地布局，土地信息，租赁历史

\*/

**public** List<RentCollection> getRentCollection(**int** bid);

/\*

参数说明：userid,字符串型，为用户编号；lidList，土地编号的集合;str, 为用户的申请数据;info\_str，为发送的消息；

返回值：int型，值为0，代表一个用户多次申请了同一块土地；值为1，代表正常

函数功能：同意用户的租赁申请

\*/

**public** **int** submitLandApply(String userid,String lidList,String str,String info\_str);

/\*

参数说明：landinfoStr,字符串型，为土地具体信息；layoutStr，为土地布局的信息；

返回值：无返回值

函数功能：更新土地布局

\*/

**public** **void** updateLayInfo(String landinfoStr,String layoutStr);

/\*

参数说明：bid，为基地编号； path，项目的路径

返回值：无返回值

函数功能：删除该基地下的所有土地

\*/

**public** **void** delLayout\_info(**int** bid,String path);

}

//审核模块业务逻辑层接口

**public** **interface** checkService {

/\*

参数说明：id,为状态值;pageindex,为页码;size,为每页条数;order,排序的列; orderDir,为升序还是降序

返回值： CheckList,为土地申请视图CheckView对象的集合

函数功能：查询所有土地申请记录(获得的记录按申请时间，默认为倒序)

\*/

**public** CheckList getLandApply(**int** id,**int** pageindex,**int** size,**int** order,String orderDir);

/\*

参数说明：landstr，土地信息的封装；recordstr,土地编号的封装;infostr,为信息的封装

返回值： int型，0为执行不成功，1为执行成功

函数功能： 同意申请

\*/

**public** **int** agreeApply(String landstr,String recordstr,String infostr);

/\*

参数说明：recordstr,土地编号的封装;infostr,为信息的封装

返回值： 无返回值

函数功能： 拒绝申请

\*/

**public** **void** refuseapply(String recordStr,String infoStr);

/\*

参数说明：recordstr,土地编号的封装;infostr,为信息的封装

返回值： 无返回值

函数功能： 逾期恢复

\*/

**public** **void** overduerecovery(String recordStr,String infoStr);

/\*

参数说明：la\_id,序号

返回值： List<CheckView>，CheckView对象的集合；

函数功能： 获得每块土地的信息

\*/

**public** List<CheckView> Rentdetail(**int** la\_id);

/\*

参数说明：flag,为标志值;pageindex,为页码;size,为每页条数;order,排序的列; orderDir,为升序还是降序 ,

返回值： CheckList,为土地申请视图CheckView对象的集合

函数功能：根据条件获取土地申请记录(获得的记录按申请时间，默认为倒序)

\*/

**public** CheckList getInfo(**int** flag,**int** pageindex,**int** size,String basename,String username,String usercollage,**int** order,String orderDir);

/\*

参数说明：无参数

返回值： List<UserInfo>，UserInfo对象的集合；

函数功能： 获得申请人

\*/

**public** List<UserInfo> getappliInfos();

/\*

参数说明：landstr，土地编号的集合;recordstr,为记录主键的集合; infostr,为消息的集合

返回值：无返回值；

函数功能： 取消交费

\*/

**public** **void** cancelPayFor(String landstr, String recordstr, String infostr);

/\*

参数说明：landstr，土地编号的集合;recordstr,为记录主键的集合; infostr,为消息的集合

返回值：无返回值；

函数功能：确认交费

\*/

**public** **void** confirmPayFor(String landstr, String recordstr, String infostr);

}

//土地使用信息表的业务逻辑层接口

**public** **interface** LandRentService {

/\*

参数说明：bname,为基地名;deptName,部门名称;plantingContent,种植内容; page,为页码;length,为每页条数

返回值：RentList，为土地使用对象LandRentInfo的集合

函数功能：查询所有土地使用情况

\*/

**public** RentList getLandRentInfos(String bname,String deptName,String plantingContent,**int** page,**int** length);

/\*

参数说明：check,土地信息记录的编号数组

返回值：无返回值

函数功能：删除土地信息

\*/

**public** **void** deleteRentInfo(String[] check);

/\*

参数说明：dept,申报部门;planCareer,从事内容;expense,费用;startTime,开始时间;endTime,结束时间;

返回值： 无返回值

函数功能：修改土地信息

\*/

**public** **void** landManageUpdate(**int** dept,String planCareer,**int** expense,String startTime,String endTime,**int** lr\_id);

}

//用户信息表的逻辑层接口

**public** **interface** UserInfoService {

/\* 参数： pageindex,为页码;size,为每页条数 ;searchValue,为模糊查询值

返回值： MangerList，包括了用户信息记录和记录条数

函数功能：获取用户信息记录

\*/

**public** MangerList manger(**int** pageindex,**int** size,String searchValue);

/\* 参数： id,为用户编号

返回值： List<Manger>，Manger对象的集合

函数功能：获取用户信息详情

\*/

**public** List<Manger> Mangerdetail(String id);

/\* 参数： str,为多个用户编号的封装

返回值：无返回值

函数功能：删除多个用户信息记录

\*/

**public** **void** deleteInfo(String str);

/\*

参数说明：id,为用户编号;name,为用户名称;sex,为用户性别;birthdate,为用户生日日期;category,用工类型...

返回值： 无返回值

函数功能：修改用户信息

\*/

**public** **void** upInfo(String id,String name,String sex,String birthdate,String category,String attritube,String dept,String telephone,String idcard,String password);

/\*

参数说明：无参数

返回值： List<ApplyDept>，为ApplyDept对象的集合

函数功能：获取用户信息所存在部门

\*/

**public** List<ApplyDept> getDepts();

/\*

参数说明：无参数

返回值： List<Admin>，为Admin对象的集合

函数功能：获取Admin中人员属性集合

\*/

**public** List<Admin> getAttritube();

/\*

参数说明：各参数分别为用户表中的字段

返回值： int型，成功返回1

函数功能：增加用户信息

\*/

**public** **int** addInfo(String id,String name,String sex,String birthdate,String category,String attritube,String dept,String telephone,String idcard,String password);

/\*

参数说明：dept,为部门

返回值： List<Manger>，Manger对象的集合

函数功能：获得导出的用户信息记录

\*/

**public** List<Manger> exportPersonInfo(String dept);

}

//发布公告和消息的逻辑层

**public** **interface** NotificationService {

/\*

参数说明：insertSql,字符串,为要插入的sql字符串

返回值： 无返回值

函数功能：发布公告

\*/

**public** **void** setNotification(String insertSql);

/\*

参数说明：title，为标题;content,为消息的内容;department,为标志值

返回值： 无返回值

函数功能：发布消息

\*/

**public** **void** addMessage(String title,String content,String department);

/\*

参数说明：currentPage，当前页;itemsPerPage,为每页的条数

返回值： List,为公告记录的集合

函数功能：获得当前页的公告记录

\*/

**public** List getNotificationInfo(String currentPage,String itemsPerPage);

/\*

参数说明：userid,为用户编号;currentPage，当前页;itemsPerPage,为每页的条数

返回值： List,为公告记录的集合

函数功能：获得某用户当前页的消息记录

\*/

**public** List getMessageInfos(String userid,String currentPage,String itemsPerPage);

/\*

参数说明：userid,为用户编号;

返回值： List,为消息记录的集合

函数功能：获得某用户前五条消息

\*/

**public** List<Message> getMessageTop5Infos(String userid);

/\*

参数说明：无参数值

返回值： List,为公告记录的集合

函数功能：获得前五条公告

\*/

**public** List<Notification> getNotificationTop5Infos();

/\*

参数说明：userid,为用户编号;

返回值： int型

函数功能：获得某用户未读消息的条数

\*/

**public** **int** getNoreadMessageCount(String userid);

/\*

参数说明：id,为消息记录编号;

返回值： 无返回值

函数功能：将某条信息设置为已读

\*/

**public** **void** setReadMessage(**int** id);

/\*

参数说明：id,为公告记录编号;

返回值： Notification,为公告信息对象

函数功能：获得公告详情

\*/

**public** Notification getNotification(String id);

}

Dao层:  
//基地信息表的数据库连接层接口

**public** **interface** BaseInfoDao {

/\*

参数说明：baseType,整型，表示基地类型

返回值： List<BaseInfo>,为基地信息BaseInfo对象的集合

函数功能：根据基地类型获得相应基地的信息

\*/

**public** List<BaseInfo> getBaseInfos(**int** baseType);

}

//土地布局管理表的数据库连接层接口

**public** **interface** LandInfoDao {

/\*

参数说明：lid,整型，表示土地编号

返回值： List<LandInfo>,为土地管理LandInfo对象的集合

函数功能：根据土地编号获取该土地布局管理信息

\*/

**public** List<LandInfo> getLandInfo(String lid);

/\*

参数说明：bid,整型，表示基地编号

返回值： List<LandInfo>,为土地管理LandInfo对象的集合

函数功能：根据基地编号获得该基地中的所有土地布局管理信息

\*/

**public** List<LandInfo> getLandInfos(**int** bid);

/\*

参数说明：bid,整型，表示基地编号

返回值： 无参数值

函数功能：根据基地编号删除该基地下的所有土地布局

\*/

**public** **void** delLayout\_info(**int** bid);

/\*

参数说明：landinfoStr,土地信息封装信息;layoutStr,土地布局封装信息

返回值： 无参数值

函数功能：向数据库插入土地信息记录和土地布局记录

\*/

**public** **void** doLayout\_info(String landinfoStr,String layoutStr);

}

//土地申请表的数据库连接层接口

**public** **interface** LandApplyDao {

/\*

/\*

参数说明：无参

返回值： List<LandApply>,为土地申请LandApply对象的集合

函数功能：查询所有土地申请记录

\*/

**public** List<LandApply> getLandApplys();

/\*

参数说明：sp, 为Startplan对象(工作计划对象)

返回值： 无返回值

函数功能：修改工作计划

\*/

**public** **void** updateLandApplyDate(Startplan sp);

/\*

参数说明：无参数值

返回值： 无返回值

函数功能：结束工作计划

\*/

**public** **void** endAllRent();

/\*

参数说明：id,为工作计划表的主键

返回值： 为Startplan对象(工作计划对象)

函数功能：获取工作计划详情

\*/

**public** Startplan getStartPlan(String id);

/\*

参数说明：bid,整型，为基地编号

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

函数功能： 根据基地编号bid获取土地布局+土地信息+土地租赁历史

\*/

**public** List<RentCollection> getRentCollection(**int** bid);

/\*

参数说明：semester,字符串型，为学年学期

返回值： long[],长整形的数值集合;

函数功能： 获取当前阶段的维修申请，租赁申请，实习申请

\*/

**public** **long**[] getRepairAndPracCount(String semester);

/\*

参数说明：bid,整型，为基地编号

返回值： List<RentAdd>，为RentAdd对象的集合

函数功能： 获得土地租赁历史

\*/

**public** List<RentAdd> getRentAdd(**int** bid);

/\*

参数说明：userid,字符串型，为用户编号;lidList，土地编号集合;str,为申报的信息集合

返回值： int型，申报成功为1，同一个人申请同一块土地多次，为0

函数功能：租赁申报， 申报成功为1，申报失败为0

\*/

**public** **int** submitApply(String userid,String lidList,String str);

}

//审核模块的数据库交互层

**public** **interface** CheckViewDao {

/\*

参数说明：id,为状态值;basename,为基地名称;username,为用户名;usercollage,为部门;

pageindex,为页码;size,为每页条数;columnName,为排序的列数;orderDir,为升序还是倒序

返回值： CheckList，该对象包含了土地租赁申请记录和记录条数

函数功能：根据基地名称，用户名，部门三个条件刷选申请记录状态为id的记录

\*/

**public** CheckList getLandApply(**int** id, **int** pageindex, **int** size,

String basename, String username, String usercollage,

String columnName, String orderDir);

/\*

参数说明：la\_id,为土地租赁申请记录编号

返回值： List<CheckView>，为土地租赁记录

函数功能：获得特定的土地租赁记录信息

\*/

**public** List<CheckView> detail(**int** la\_id);

/\*

参数说明：recordStr,为土地租赁申请记录编号封装后的字符串,status,为记录所要改成的状态值

返回值： 无返回值

函数功能：更改特定记录的状态值

\*/

**public** **void** updateStatus(String recordStr, **int** status);

/\*

参数说明：sql,为插入的消息的sql语句

返回值： 无返回值

函数功能：向消息表中插入消息

\*/

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

/\*

参数说明：landstr,为土地编号;status1为转变后的状态值，status2为转变前的状态值

返回值： 无返回值

函数功能：将特定的土地由状态status1改成status2

\*/

**public** **void** changeSolid(String landstr, **int** status1, **int** status2);

/\*

参数说明：recordStr,为土地租赁记录的编号;status,为要改成的状态值

返回值： 无返回值

函数功能：确认交费，将记录的状态改为申请成功，并将记录插入土地租赁历史表中

\*/

**public** **void** payForSuccess(String recordStr, **int** status);

/\*

参数说明：landstr,为土地编号的封装

返回值： 无返回值

函数功能：同意申请，将审核状态的变为锁定状态，同时给锁定状态的发送通知

\*/

**public** **void** lockInfo(String landstr);

/\*

参数说明：landstr,为土地编号的封装

返回值： 无返回值

函数功能：取消交费，将锁定状态的变为审核状态，发送通知

\*/

**public** **void** releaseInfo(String landstr);

/\*

参数说明：landstr,为土地编号的封装

返回值： 无返回值

函数功能：确认交费，将锁定状态的变为同类竞争，发送通知

\*/

**public** **void** confirmInfo(String landstr);

/\*

参数说明：recordStr,为土地租赁申请编号的封装;status，为要改成的状态值

返回值： int型

函数功能：同意申请，将审核中的记录变为交费中，判断土地编号是否有相同的，返回flag值

\*/

**public** **int** agreeInfo(String recordStr, **int** status);

/\*

参数说明：无参数值

返回值： List<Map<String, String>>

函数功能：获取租赁审批未审核的申请人

\*/

**public** List<Map<String, String>> getCheckApplicant();

/\*

参数说明：无参数值

返回值： List<Map<String, String>>

函数功能：获取租赁审批未交费的申请人

\*/

**public** List<Map<String, String>> getPayApplicant();

/\*

参数说明：无参数值

返回值： List<Map<String, String>>

函数功能：获取租赁审批逾期的申请人

\*/

**public** List<Map<String, String>> getoverdueApplicant();

/\*

参数说明：无参数值

返回值： List<Map<String, String>>

函数功能：获取租赁审批未审核的部门

\*/

**public** List<Map<String, String>> getCheckDept();

/\*

参数说明：无参数值

返回值： List<Map<String, String>>

函数功能：获取租赁审批未交费的部门

\*/

**public** List<Map<String, String>> getPayDept();

/\*

参数说明：无参数值

返回值： List<Map<String, String>>

函数功能：获取租赁审批逾期的部门

\*/

**public** List<Map<String, String>> getoverdueDept();

/\*

参数说明：recordStr,为土地租赁申请记录的编号

返回值： 无返回值

函数功能：逾期恢复

\*/

**public** **void** overduerecovery(String recordStr);

}

//土地使用表的数据库连接层接口

**public** **interface** LandRentInfoDao {

/\*

参数说明：userId,字符串型，为土地的使用人编号

返回值： List<LandRentInfo>，为土地使用对象LandRentInfo的集合

函数功能：根据土地使用人编号查询土地使用记录

\*/

**public** List<LandRentInfo> getLandRentInfo(String userId);

/\*

参数说明：bname,字符串型，基地名称;dept,为部门名称;planting,为种植内容;page,为页码;length,为每页条数

返回值： RentList，存放记录和记录条数的对象

函数功能：根据条件刷选并获得土地租赁信息

\*/

**public** RentList getRentMaintain(String bname,String dept,String planting,**int** page,**int** length);

/\*

参数说明：str,为土地信息记录的编号集合

返回值： 无返回值

函数功能：删除多条土地租赁信息记录

\*/

**public** **void** deleteRentInfo(String str);

/\*

参数说明：lr\_id,为

返回值： 无返回值

函数功能：删除多条土地租赁信息记录

\*/

**public** List<RentMaintain> getSingleRentInfo(String lr\_id,String dept );

/\*

参数说明：无参数

返回值： List<ApplyDept>，为ApplyDept对象的集合

函数功能：获得土地信息记录中存在的部门

\*/

**public** List<ApplyDept> getExistDept();

/\*

参数说明：无参数

返回值： List<String>,String的集合

函数功能：获得土地信息记录中存在的种植内容

\*/

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

}

//用户信息表的数据库连接层接口

**public** **interface** UserInfoDao {

/\* 参数： pageindex,为页码;size,为每页条数 ;searchValue,为模糊查询值

返回值： MangerList，包括了用户信息记录和记录条数

函数功能：获取用户信息记录

\*/

**public** MangerList manger(**int** pageindex, **int** size,String searchValue);

/\* 参数： id,为用户编号

返回值： List<Manger>，Manger对象的集合

函数功能：获取用户信息详情

\*/

**public** List<Manger> Mangerdetail(String id) ;

/\* 参数： str,为多个用户编号的封装

返回值：无返回值

函数功能：删除多个用户信息记录

\*/

**public** **void** deleteInfo(String str) ;

/\*

参数说明：id,为用户编号;name,为用户名称;sex,为用户性别;birthdate,为用户生日日期;category,用工类型...

返回值： 无返回值

函数功能：修改用户信息

\*/

**public** **void** upInfo(String id, String name, String sex, String birthdate, String category, String attritube,String dept, String telephone, String idcard,String password);

/\*

参数说明：无参数

返回值： List<ApplyDept>，为ApplyDept对象的集合

函数功能：获取用户信息所存在部门

\*/

**public** List<ApplyDept> getDepts();

/\*

参数说明：无参数

返回值： List<Admin>，为Admin对象的集合

函数功能：获取Admin中人员属性集合

\*/

**public** List<Admin> getAttritube();

/\*

参数说明：各参数分别为用户表中的字段

返回值： int型，成功返回1

函数功能：增加用户信息

\*/

**public** **int** addInfo(String id, String name, String sex, String birthdate, String category, String attritube, String dept, String telephone, String idcard, String password);

/\*

参数说明：dept,为部门

返回值： List<Manger>，Manger对象的集合

函数功能：获得导出的用户信息记录

\*/

**public** List<Manger> exportPersonInfo(String dept);

}

//发布公告和消息的数据交互层

**public** **interface** NotificationDao {

/\*

参数说明：insertSql,字符串,为要插入的sql字符串

返回值： 无返回值

函数功能：发布公告

\*/

**public** **void** setNotification(String insertSql);

/\*

参数说明：currentPage，当前页;itemsPerPage,为每页的条数

返回值： List,为公告记录的集合

函数功能：获得当前页的公告记录

\*/

**public** List getNotificationInfo(String currentPage,String itemsPerPage);

/\*

参数说明：title，为标题;content,为消息的内容;department,为标志值

返回值： 无返回值

函数功能：发布消息

\*/

**public** **void** addMessage(String title,String content,String department);

/\*

参数说明：userid,为用户编号;currentPage，当前页;itemsPerPage,为每页的条数

返回值： List,为公告记录的集合

函数功能：获得某用户当前页的消息记录

\*/

**public** List getMessageInfos(String userid,String currentPage,String itemsPerPage);

/\*

参数说明：userid,为用户编号;

返回值： List,为消息记录的集合

函数功能：获得某用户前五条消息

\*/

**public** List<Message> getMessageTop5Infos(String userid);

/\*

参数说明：无参数值

返回值： List,为公告记录的集合

函数功能：获得前五条公告

\*/

**public** List<Notification> getNotificationTop5Infos();

/\*

参数说明：userid,为用户编号;

返回值： int型

函数功能：获得某用户未读消息的条数

\*/

**public** **int** getNoreadMessageCount(String userid);

/\*

参数说明：id,为消息记录编号;

返回值： 无返回值

函数功能：将某条信息设置为已读

\*/

**public** **void** setReadMessage(**int** id);

/\*

参数说明：id,为公告记录编号;

返回值： Notification,为公告信息对象

函数功能：获得公告详情

\*/

**public** Notification getNotification(String id);

}