**实习模块后台分析文档**

1. 需求说明

此模块主要涉及三个功能。第一，管理员导入实习计划安排表（excel文档）；第二，用户完善本院的实习计划与安排；第三，管理员检测实习计划是否填写完成并对数据进行相关的增删查改的工作。下面我将对此进行详细的阐述与分析。

1. 流程图
2. 管理员导入数据

管理员

实习安排表

实习维护

进入

导入

当年的实 习安排数据

产生

1. 管理员导出+增加+删除+修改记录

管理员

进入

实习维护

显示

删除

修改

增加

导出

当年的实习计划记录

单条实习安排记录

需完善或填写错误的记录

按照需求所需的实习安排

多条无用的实习安排数据

选择

学年学期

3.检测数据完整性

管理员

点击

检测数据完整性按钮

显示

用户尚未完善的记录

点击

提醒按钮

发送

消息（提醒用户）

4.用户完善实习安排计划

对应的记录

修改完成的记录

用户

进入

实习申请

显示

用户所属院的所有记录

点击

每条记录

完善

未完成的数据输入

保存

删除

1. 系统提醒

当年实习计划数据

判断

是否进入起始周

是

发送信息提醒用户

完毕

是否填写完毕周

否

否

是

三．表的创建

1.课程安排表(courseArrange)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type |
| Id(编号) | p |  |  | Int（自增） |
| cid(课程代码) |  | Not null |  | varchar |
| count(人数) |  | null |  | int |
| selectedCount(已选人数) |  | null |  | int |
| composition(教学班组成) |  | null |  | varchar |
| college(开课学院) |  | Not null |  | varchar |
| coursename(课程名称) |  | Not null |  | varchar |
| weekClassify(周学时) |  | null |  | float |
| credit(学分) |  | null |  | float |
| courseNature(课程性质) |  | null |  | varchar |
| courseCategory(课程类别) |  | null |  | varchar |
| checkMethod(考核方式) |  | null |  | varchar |
| mid(专业编号) | F | null | 参照“专业表”的主键 | varchar |
| siteRepuire(场地要求) |  | null |  | varchar |
| tid(教师职工号) | F | null | 参照教师表的主键 | varchar |
| tname(教师姓名) |  | null |  | varchar |
| technicalTitle (职称) |  | null |  | varchar |
| Semester（学期) |  | null |  | varchar |
| Week(起止周) |  | null |  | varchar |
| Status(修改状态标识) |  | null | 0 | int |

注：Status(修改状态标识)，用此字段来标识用户是否已经完善了实习计划，若status为0，标识用户尚未完善实习计划；若status为1，标识用户已经完善了实习计划---此字段用于检测数据的完成度

2.班级安排—课程表（classArrage\_course）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type |
| id | p | Not null | 自增 | int |
| week(周次) |  | null |  | int |
| starttime(开始时间) |  | null |  | varchar |
| endtime(结束时间) |  | null |  | varchar |
| content（实习内容） |  | null |  | varchar |
| source(实习来源） |  | null |  | varchar |
| site(实习地点) |  | null |  | varchar |
| category(实习类别) |  | null |  | varchar |
| form(实习形式) |  | null |  | varchar |
| telephone(联系电话) |  | null |  | varchar |
| aim(实习目的) | F | Not null | 参照“专业培训表”的主键 | varchar |
| expense(经费) |  | null |  | varchar |
| cid | F | Not null | 参照“课程表“的主键 | int |
| guideTeacher(指导教师) |  | null |  | varchar |
| assistant(实验员) |  | null |  | varchar |
| remark(备注) |  | null |  | varchar |

3.专业培训表(major\_aim)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Key Type | Null/Unique | Default | Data Type |
| Id | p | Not null |  | Int(自增) |
| majorid（专业编号） | F | Not null | 参照“专业表“的主键 | varchar |
| aim(培训目的) |  | Not 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.实习申请控制：PlanController

2.实习计划管理控制：PlanMaintainController

Service层： 1.实习申请逻辑：PlanService

2.实习计划管理逻辑：PlanMaintainService

Dao层: 1.实习申请数据交互：PlanDao

2.实习计划管理数据交互：PlanMaintainDao

**Controller层：**

//实习申请的控制层

@Controller("planController")

**public** **class** PlanController **implements** ServletContextAware{

//检测学年学期和数据条数

@RequestMapping("/Checkinfo.do")

**public** String Checkinfo(HttpServletRequest request,

HttpServletResponse response);

// 显示用户所属学院的所有实习计划

@RequestMapping("/displayThisCollegePlan.do")

**public** String displayThisCollegePlan(HttpServletRequest request,

HttpServletResponse response);

// 获取某门课程的班级安排

@RequestMapping("/getplandata.do")

**public** String getplandata(HttpServletRequest request,

HttpServletResponse response);

// 保存对实习申请班级安排的修改

@RequestMapping("/savePlanModify.do")

**public** String savePlanModify(HttpServletRequest request,

HttpServletResponse response);

// 功能:1.获取专业所对应的培训目的(通过参数课程代码)2.获取基地类型

@RequestMapping("/getPlanAim.do")

**public** String getPlanAim(HttpServletRequest request,

HttpServletResponse response);

//1.获取专业所对应的培训目的(通过参数课程代码)2.获取基地类型3.通过基地类型名称获取基地名字

@RequestMapping("/getBasenameOfType.do")

**public** String getBasenameOfType(HttpServletRequest request,

HttpServletResponse response);

//根据基地类型获取基地名字

@RequestMapping("/getBasenameOneOfType.do")

**public** String getBasenameOneOfType(HttpServletRequest request,

HttpServletResponse response);

// 获取所有学院

@RequestMapping("/getCollege.do")

**public** String getCollege(HttpServletRequest request,

HttpServletResponse response);

// 获取学院所对应的教师

@RequestMapping("/getCollege\_Teacher.do")

**public** String getCollege\_Teacher(HttpServletRequest request,

HttpServletResponse response);

// 删除班级安排记录(单条)

@RequestMapping("/deleteClassRecord.do")

**public** String deleteClassRecord(HttpServletRequest request,

HttpServletResponse response);

}

//实习计划管理的控制层

@Controller("planMaintainController")

**public** **class** PlanMaintainController **implements** ServletContextAware{

// 显示实习计划数据

@RequestMapping("/displayPlanInfo.do")

**public** String displayPlanInfo(HttpServletRequest request,

HttpServletResponse response);

// 导出实习计划安排表

@RequestMapping("/exportPlanInfo.do")

**public** String exportPlanInfo(HttpServletRequest request,

HttpServletResponse response);

// 判断是否存在该教师

@RequestMapping("/checkIsThisUser.do")

**public** String checkIsThisUser(HttpServletRequest request,

HttpServletResponse response);

// 判断是否存在该专业编号

@RequestMapping("/checkIsThisMid.do")

**public** String checkIsThisMid(HttpServletRequest request,

HttpServletResponse response);

// 判断是否存在该课程代码

@RequestMapping("/checkIsThisCid.do")

**public** String checkIsThisCid(HttpServletRequest request,

HttpServletResponse response);

// 增加单条实习计划数据

@RequestMapping("/addOnePlanInfo.do")

**public** String addOnePlanInfo(HttpServletRequest request,

HttpServletResponse response);

// 获取所有学院,学期学年

@RequestMapping("/getPlanMaintainInfo.do")

**public** String getPlanMaintainInfo(HttpServletRequest request,

HttpServletResponse response);

// 获取开课学院

@RequestMapping("/getReadyCollege.do")

**public** String getReadyCollege(HttpServletRequest request,

HttpServletResponse response);

// 获取学年

@RequestMapping("/getReadyYear.do")

**public** String getReadyYear(HttpServletRequest request,

HttpServletResponse response);

// 获取学期

@RequestMapping("/getReadySem.do")

**public** String getReadySem(HttpServletRequest request,

HttpServletResponse response);

// 导入实习计划数据

@RequestMapping("/importPlanInfo.do")

**public** String importPlanInfo(HttpServletRequest request,

HttpServletResponse response,ModelMap map);

// 检测数据完整性

@RequestMapping("/checkIsSave.do")

**public** String checkIsSave(HttpServletRequest request,

HttpServletResponse response);

// 删除实习计划数据

@RequestMapping("/deletePlanInfo.do")

**public** String deletePlanInfo(HttpServletRequest request,

HttpServletResponse response);

// 提醒教师完善实习计划

@RequestMapping("/callAttention.do")

**public** String callAttention(HttpServletRequest request,

HttpServletResponse response);

// 修改课程安排表

@RequestMapping("/alterOneRecord.do")

**public** String alterRecord(HttpServletRequest request,

HttpServletResponse response);

}

**Service层：**

//实习申请的逻辑层

**public** **interface** PlanService {

/\*

1.参数：userid,字符串型,代表用户的编号; pageindex,为当前页数;size,为每页的条数; order,排序列; orderDir,为排序的顺序; searchValue,为模糊查询的值.

2.返回值： PlanList,需要显示给用户的数据对象

3.函数功能：根据用户编号获取该用户所在学院的实习计划

\*/

**public** PlanList getThisCollegePlan( String userid,

**int** pageindex, **int** size, **int** order, String orderDir,

String searchValue,String semester);

/\*

1.参数：id,整型,代表班级安排记录的主键值

2.返回值： 无返回值

3.函数功能：删除单条班级安排记录

\*/

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

/\*

1.参数：id,整型，为课程表id plandata,字符串型，为插入 班级安排\_课程表 的多条记录所构成的字符串

2.返回值： 无返回值

3.函数功能：完善实习计划

\*/

**public** **void** savePlanModify(**int** id, String plandata);

/\*

1.参数：majorid,字符串型，代表专业代码

2.返回值： List<major\_aim>,存放专业培训表多条记录的对象集合

3.函数功能：根据专业编号获取该专业的培训目的

\*/

**public** List<Majoraim> getPlanAim(String cid);

/\*

1.参数：collagename,字符串型，代表学院名字

2.返回值： List<UserInfo>,教师的字符串集合

3.函数功能：根据学院编号获取该学院的老师

\*/

**public** List<UserInfo> getCollege\_Teacher(String collagename);

/\*

1.参数：id,整型， 代表课程表id

2.返回值： List<Classarragecourse>,申请表里的数据集合

3.函数功能：根据课程id获取申请表的数据集合

\*/

**public** List<Classcourse> plandata(**int** id);

/\*

1.参数：plandata，为实习申请信息

2.返回值： 无返回值

3.函数功能：修改课程安排表

\*/

**public** **void** alterRecord(String plandata);

/\*

1.参数：userid,用户id;semester,学年学期

2.返回值： int型，记录数

3.函数功能：获得该年该用户所处学院的实习记录条数

\*/

**public** **int** checkinfo(String userid,String semester);

/\*

1.参数：typename,字符串型，为基地类型

2.返回值： List<String>，为基地名字的集合

3.函数功能：根据基地类型名获得基地名称

\*/

**public** List<String> getProperBase(String typename);

}

//实习计划管理的逻辑层

**public** **interface** PlanMaintainService {

/\*

1.参数：semester,字符串型,代表某一学年的某一学期，例：2016-2017-1; pageindex,为当前页数;

size,为每页的条数; order,排序列; orderDir,为排序的顺序; searchValue,为模糊查询的值.

2.返回值：PlanList,需要显示给用户的数据对象

3.函数功能：获取当年所有的实习计划

\*/

**public** PlanList getPlanInfo(String semester, **int** pageindex, **int** size,

**int** order, String orderDir, String searchValue);

/\*

1.参数：str,字符串型，为需要插入课程安排表的记录字符串

2.返回值： 无返回值

3.函数功能：增加一条实习计划

\*/

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

/\*

1.参数：semester,字符串型,代表某一学年的某一学期，例：2016-2017-1.

2.返回值：List<AllPlan>，为导出表所需要的所有记录的对象集合

3.函数功能：获取导出表所需的记录的对象集合

\*/

**public** List<AllPlan> getPlanTable(String year, **int** semester, String college);

/\*

1.参数：semester,字符串型,代表某一学年,例：2016-2017; status,整型，修改位，0代表未完善，1代表已完善

2.返回值：List<AllPlan>，为导出表所需要的所有记录的对象集合

3.函数功能：获取未完善的记录

\*/

**public** PlanList checkIsSave(String semester, **int** status, **int** pageindex,

**int** size, **int** order, String orderDir, String searchValue);

/\*

1.参数：recordstr,字符串型，为多个记录编号整合所构成的字符串

2.返回值：无返回值

3.函数功能：删除实习计划记录

\*/

**public** **void** deletePlanInfo(String recordstr);

/\*

1.参数：recordstr,字符串型，为多个记录编号整合所构成的字符串

2.返回值：无返回值

3.函数功能：提醒用户完善实习计划

\*/

**public** **void** callAttention(String recordstr);

/\*

1.参数：无参

2.返回值：List<String>

3.函数功能：获取学年学期

\*/

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

/\*

1.参数：id,整型，为用户的id

2.返回值：boolean

3.函数功能：判断是否存在该教师职工号

\*/

**public** **boolean** checkIsUser(String id);

/\*

1.参数：aid,整型，为专业的编号

2.返回值：boolean

3.函数功能：判断是否存在该专业编号

\*/

**public** **boolean** checkIsMid(String aid);

/\*

1.参数：semester,为学年学期，cid,整型，为课程代码

2.返回值：int,0代表没有该课程代码,1代表存在该课程代码

3.函数功能：判断是否该学期是否存在该课程代码

\*/

**public** **int** checkIsCid(String semester, String cid);

/\*

1.参数：无参

2.返回值：List<String>

3.函数功能：获取实习计划的学院

\*/

**public** List<String> getPlanCollege(String semester);

/\*

1.参数：semester,字符串，为学年学期;str,字符串，为课程表记录的插入语句

2.返回值：无参

3.函数功能：删除原来学年学期的记录

\*/

**public** **void** delete\_0(String semester);

/\*

1.参数：str,字符串型，为实习计划记录的封装

2.返回值：无返回值

3.函数功能： 插入实习计划

\*/

**public** **void** add\_0(String str);

/\*

1.参数：学年

2.返回值：List<String>，为学期的集合

3.函数功能： 根据学年获取学期

\*/

**public** List<String> getSem(String semester);

/\*

1.参数：year,字符串型，为学年;semester,字符串型，为学期

2.返回值：List<String>，为学院的集合

3.函数功能： 根据学年学期获取学院

\*/

**public** List<String> getCollegehh(String year, **int** semester);

/\*

1.参数：semester,字符串型，为学期学年;startTime,字符串型,为该学期的第一周时间

2.返回值：无返回值

3.函数功能： 插入该学年学期的第一周时间

\*/

**public** **void** addStartDate(String semester, String startTime);

/\*

1.参数：无参数

2.返回值：无返回值

3.函数功能： 获取各学年学期的开始时间

\*/

**public** List<StartDate> getStartDate();

/\*

1.参数：id,整型，为实习申请记录的编号;plandata,字符串型,为实习申请记录修改后的信息

2.返回值：无返回值

3.函数功能： 修改实习申请计划

\*/

**public** **void** alterRecord(**int** id,String plandata);

/\*

1.参数：year,字符串型,为学年;semester,字符串型，为学期;college,字符串型，为学院

2.返回值：List<PracticeCollection>，为PracticeCollection对象的集合

3.函数功能： 获取导出的信息

\*/

**public** List<PracticeCollection> getPlanTable\_0(String year, **int** semester,

String college);

}

**Dao层：**

//实习申请的数据交互层

**public** **interface** PlanDao {

/\*

1.参数：userid,字符串型,代表用户的编号; pageindex,为当前页数;size,为每页的条数; order,排序列;

orderDir,为排序的顺序; searchValue,为模糊查询的值.

2.返回值： PlanList,需要显示给用户的数据对象

3.函数功能：根据用户编号获取该用户所在学院的实习计划

\*/

**public** PlanList getThisCollegePlan(String userid, **int** pageindex, **int** size,

String columnName, String orderDir, String searchValue,String semester);

/\*

1.参数：str1,整型，为课程表id; str3,字符串型，为插入 班级安排\_课程表 的多条记录所构成的字符串

2.返回值： 无返回值

3.函数功能：保存实习计划

\*/

**public** **void** updatePlan(**int** str1, String str3);

/\*

1.参数：id,整型,代表班级安排记录的主键值

2.返回值： 无返回值

3.函数功能：删除单条班级安排记录

\*/

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

/\*

1.参数：majorid,字符串型，代表专业代码

2.返回值： List<major\_aim>,存放专业培训表多条记录的对象集合

3.函数功能：从专业培训表中获取特定专业的多个培训目的

\*/

**public** List<Majoraim> getPlanAim(String majorid);

/\*

1.参数：collagename,字符串型，代表学院名字

2.返回值： List<UserInfo>,教师的字符串集合

3.函数功能：根据学院编号获取该学院的老师

\*/

**public** List<UserInfo> getCollege\_Teacher(String collagename);

/\*

\* 1.参数：id,整型 ,为实习申请记录的编号

\* 2.返回值： List<Classcourse>,班级安排的数据集合

\* 3.函数功能：根据实习申请记录的编号获取班级安排数据集合

\*/

**public** List<Classcourse> plandata(**int** id);

/\*

1.参数：plandata，为实习申请信息

2.返回值： 无返回值

3.函数功能：修改课程安排表

\*/

**public** **void** alterRecord(String plandata);

/\*

1.参数：userid,用户id;semester,学年学期

2.返回值： int型，记录数

3.函数功能：获得该年该用户所处学院的实习记录条数

\*/

**public** **int** checkinfo(String userid,String semester);

/\*

1.参数：typename,字符串型，为基地类型

2.返回值： List<String>，为基地名字的集合

3.函数功能：根据基地类型名获得基地名称

\*/

**public** List<String> getProperBase(String typename);

}

//实习计划管理的数据交互层

**public** **interface** PlanMaintainDao {

/\*

1.参数：semester,字符串型,代表某一学年的某一学期，例：2016-2017-1; pageindex,为当前页数; size,为每页的条数;

order,排序列; orderDir,为排序的顺序; searchValue,为模糊查询的值.

2.返回值： PlanList,需要显示给用户的数据对象

3.函数功能：获取当年所有的实习计划

\*/

**public** PlanList getPlanInfo(String semester,**int** pageindex,**int** size,String order,String orderDir,String searchValue);

/\*

1.参数：str,字符串型，为需要插入课程安排表的记录字符串

2.返回值： 无返回值

3.函数功能：向课程安排表中插入一条实习计划数据

\*/

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

/\*

1.参数：semester,字符串型,代表某一学年的某一学期，例：2016-2017-1.

2.返回值： List<AllPlan>，为导出表所需要的所有记录的对象集合

3.函数功能：获取导出表所需的记录的对象集合

\*/

**public** List<AllPlan> getPlanTable(String year,**int** semester,String college);

/\*

1.参数：semester,字符串型,代表某一学年,例：2016-2017; status,整型，修改位，0代表未完善，1代表已完善

2.返回值： List<AllPlan>，为导出表所需要的所有记录的对象集合

3.函数功能：获取未完善的记录

\*/

**public** PlanList checkIsSave(String semester,**int** status,**int** pageindex,**int** size,String order,String orderDir,String searchValue);

/\*

1.参数：recordstr,字符串型，为多个记录编号整合所构成的字符串

2.返回值：无返回值

3.函数功能：删除实习计划记录

\*/

**public** **void** deletePlanInfo(String recordstr);

/\*

1.参数：recordstr,字符串型，为课程安排表多个记录编号整合所构成的字符串

2.返回值：无返回值

3.函数功能：向特定的记录所对应的教师发送提醒的信息

\*/

**public** **void** callAttention(String recordstr);

/\*

1.参数：无参

2.返回值：List<String>

3.函数功能：获取学年学期

\*/

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

/\*

1.参数：id,整型，为用户的id

2.返回值：boolean

3.函数功能：判断是否存在该教师职工号

\*/

**public** **boolean** checkIsUser(String id);

/\*

1.参数：aid,整型，为专业的编号

2.返回值：boolean

3.函数功能：判断是否存在该专业编号

\*/

**public** **boolean** checkIsMid(String aid);

/\*

1.参数：semester,为学年学期，cid,整型，为课程代码

2.返回值：int,0代表没有该课程代码,1代表存在该课程代码

3.函数功能：判断是否该学期是否存在该课程代码

\*/

**public** **int** checkIsCid(String semester,String cid);

/\*

1.参数：无参

2.返回值：List<String>

3.函数功能：获取实习计划的学院

\*/

**public** List<String> getPlanCollege(String semester);

/\*

1.参数：semester,字符串，为学年学期;str,字符串，为课程表记录的插入语句

2.返回值：无参

3.函数功能：删除原来学年学期的记录

\*/

**public** **void** delete\_0(String semester);

/\*

1.参数：str,字符串型，为实习计划记录的封装

2.返回值：无返回值

3.函数功能： 插入实习计划

\*/

**public** **void** add\_0(String str);

/\*

1.参数：学年

2.返回值：List<String>，为学期的集合

3.函数功能： 根据学年获取学期

\*/

**public** List<String> getSem(String semester);

/\*

1.参数：year,字符串型，为学年;semester,字符串型，为学期

2.返回值：List<String>，为学院的集合

3.函数功能： 根据学年学期获取学院

\*/

**public** List<String> getCollegehh(String year,**int** semester);

/\*

1.参数：semester,字符串型，为学期学年;startTime,字符串型,为该学期的第一周时间

2.返回值：无返回值

3.函数功能： 插入该学年学期的第一周时间

\*/

**public** **void** addStartDate(String semester,String startTime );

/\*

1.参数：无参数

2.返回值：无返回值

3.函数功能： 获取各学年学期的开始时间

\*/

**public** List<StartDate> getStartDate();

/\*

1.参数：plandata，为实习申请信息

2.返回值： 无返回值

3.函数功能：修改课程安排表

\*/

**public** **void** alterRecord(**int** id,String plandata);

/\*

1.参数：year,字符串型,为学年;semester,字符串型，为学期;college,字符串型，为学院

2.返回值：List<PracticeCollection>，为PracticeCollection对象的集合

3.函数功能： 获取导出的信息

\*/

**public** List<PracticeCollection> getPlanTable\_0(String year, **int** semester,

String college);

}