# 软件工程综合实训

# 《学生日常党团活动管理系统》

# 详细设计说明书

组 号： 7

班 级： 软件三班

组 长： 杨凡

小组成员： 孙浩南

指导教师： 张德珍，张秀国

日 期： 2020.12.05

目录

[1. 引言](#_Toc5566)

[2. 程序系统的结构](#_Toc31603)

[3. 程序1（查看通知）设计说明](#_Toc15834)

[4. 程序2（查看活动）设计说明](#_Toc10631)

[5. 程序3（报名）设计说明](#_Toc25993)

[6. 程序4（签到）设计说明](#_Toc11604)

[7. 程序5（消息）设计说明](#_Toc14249)

[8. 程序6（留言）设计说明](#_Toc13306)

[9. 程序7（登陆注册）设计说明](#_Toc27591)

[10. 程序8（申请活动）设计说明](#_Toc529)

[11. 程序9（统计活动）设计说明](#_Toc31315)

[12. 程序10（统计学生&活动）设计说明](#_Toc17304)

[13. 程序11（审核活动）设计说明](#_Toc26461)

[14. 程序12（发布活动）设计说明](#_Toc31811)

[15. 程序13（发布通知）设计说明](#_Toc28)

# 引言

## 编写目的

此说明书在概要设计的基础上，对学生党团活动管理系统各个模块，程序分别进行了实现层面上的要求和说明。在以下的详细设计报告中将对在本阶段中队系统所做的所有详细设计进行说明。

主要工作包括:根据软件需求说明书所描述的数据，功能，运行，性能，需求，并依照概要设计说明书所确定的处理流程、总体设计、软件系统的结构设计，逐个模块的程序描述（包括各模块的功能、性能、输入、输出、算法、程序逻辑、接口等)。

## 背景

1. 待开发的软件系统的名称是“学生日常党团活动管理系统”
2. 本项目由XXX提出，由XXX开发，面向的用户是XX大学的学院党团活动管理人员和学生。实现该软件的计算机网络采用XX大学的校园局域网。
3. 该活动管理系统的部分学生信息应该由校教务系统提供。

# 程序系统的结构

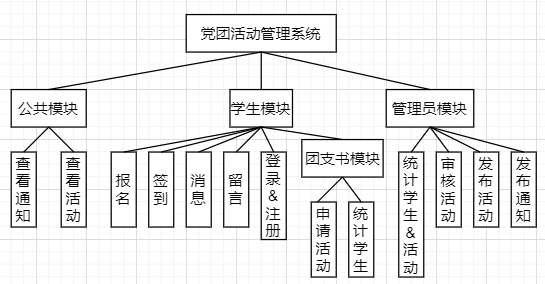


图1:系统功能结构图

# 程序设计说明

注释设计:

a.在实体类的每一条属性加上注释说明

b.MVC每一层的方法加上注释说明

# 程序1（查看通知）设计说明

## 程序描述

从数据库中读取出所有学院发布的所有公共通知以供所有访问网站的用户浏览查看。因为采用了分页查询，所以需要通过程序实现对pageBean对象的创建并返回前端页面显示。采用关键字搜索，需要做模糊处理。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 搜索关键字，分页信息 | 进行带关键字的模糊分页查询，并将总记录数，总页码，每页的数据集合，当前页码，每页显示的数据数存入PageBean对象并返回前端 | 通知的PageBean对象 |

## 输人项

给出对每一个输入项的特性，包括名称、标识、数据的类型和格式、数据值的有效范围、输入的方式。 数量和频度、输入媒体、输入数据的来源和安全保密条件等等。

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 关键字 | searchString | String | Varchar(30) |
| 总页码 | totalPage | Int | Int(20) |
| 当前页码 | currentPage | Int | Int(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 分页对象 | NoticePageBean | PageBean | 无 |

## 算法

分页查询时采用的limit后跟的两个数，一个时开始的位置start，一个是pageSize，其中，开始的位置start的计算方法是：(currentPage-1)\*pageSize;

还需要求总页码totalPage，此时需要的计算方法是：

totalPage = (totalCount % pageSize ) == 0 ? totalCount/pageSize : (totalCount/pageSize)+1;

算法就是用总记录数除以每页的记录数得到余数，如果余数是0，总页码数就是之前的商，如果余数不是0，总页码数就是之前的商+1

## 流程逻辑

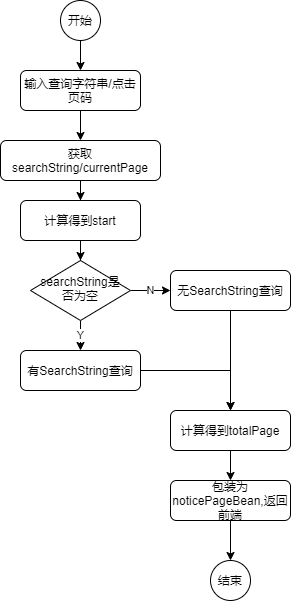


图2：查看通知流程

## 接口

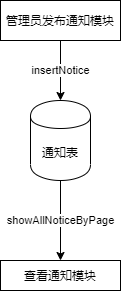


图3：产看通知接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序2（查看活动）设计说明

## 程序描述

在网站首页分页显示所有学院所有管理员发布的所有活动，要通过关键字searchString，学院名collegeCollege，班级名\_class来查询。因为采用了分页查询，所以需要通过程序实现对pageBean对象的创建并返回前端页面显示。采用关键字搜索，需要做模糊处理。学院名和班级名需要精确查询。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 关键字，学院名，班级名，分页信息 | 进行带关键字的模糊分页查询，带学院名，班级名的精确查询，并将总记录数，总页码，每页的数据集合，当前页码，每页显示的数据数存入PageBean对象并返回前端 | 活动的PageBean对象 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 关键字 | searchString | String | Varchar(30) |
| 总页码 | totalPage | Int | Int(20) |
| 当前页码 | currentPage | Int | Int(20) |
| 学院名 | collegeName | String | Varchar(30) |
| 班级名 | \_class | Stirng | Varchar(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 分页对象 | ActivityPageBean | PageBean | 无 |

## 算法

分页查询时采用的limit后跟的两个数，一个是开始的位置start，一个是pageSize，其中，开始的位置start的计算方法是：(currentPage-1)\*pageSize;

还需要求总页码totalPage，此时需要的计算方法是：

totalPage = (totalCount % pageSize ) == 0 ? totalCount/pageSize : (totalCount/pageSize)+1;

算法就是用总记录数除以每页的记录数得到余数，如果余数是0，总页码数就是之前的商，如果余数不是0，总页码数就是之前的商+1

## 流程逻辑

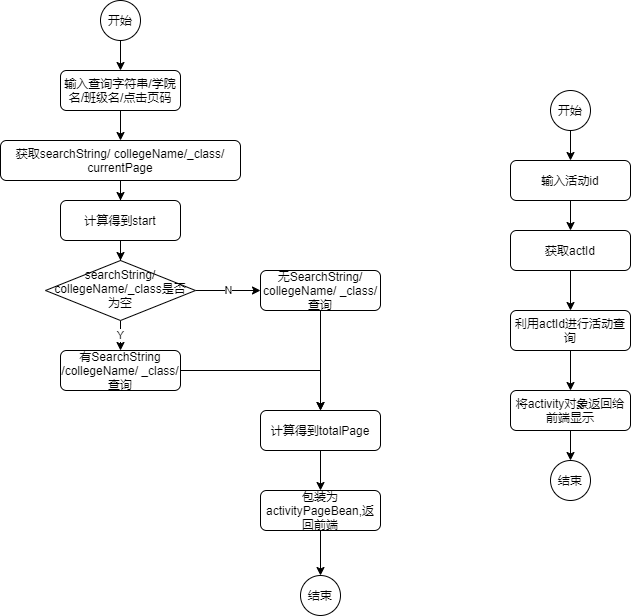


图4：查看活动流程

## 接口

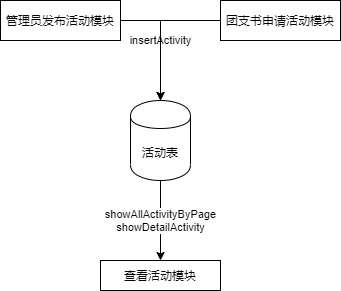


图5：查看活动接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序3（报名）设计说明

## 程序描述

该功能是建立在查看活动的基础上的。学生登陆后报名，将活动id和session中的学生信息一起提交给后台，关联student，activity，stu\_act三表一起，写入报名信息

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 学生id，活动id | 输入学生id，活动id，设置报名状态为已报名，是否签名设置为否，将所有信息写入stu\_act，并根据活动id将活动中的报名人数加一 | 报名是否成功的信息 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 学生id | StuId | int | Int(20) |
| 活动id | ActId | int | Int(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 报名是否成功 | Msg | String | 无 |

## 算法

报名时，需要做各种判断来判断。该学生是否能进行报名：

* 需要登录后才能报名
* 自己的身份：班级，学院，党员，团员
* 活动的类型：班级党团，班级党，学院党团，学院党
* 根据学生的身份和所要报名的活动的类型来判断是否能成功报名。

## 流程逻辑

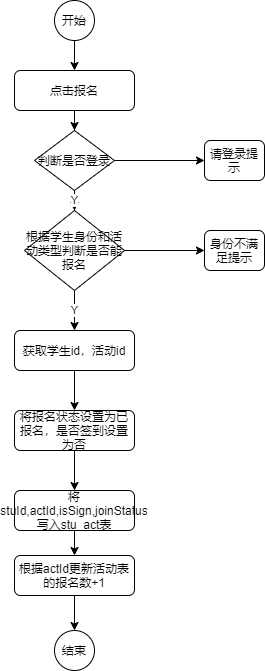


图6：报名流程

## 接口

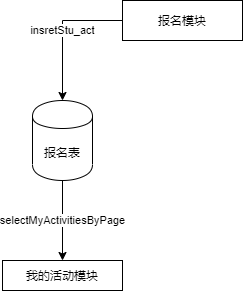


图7:报名接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序4（签到）设计说明

## 程序描述

在我的活动详情界面点击签到，将报名id传入后端，根据报名id来更新报名表中的isSign字段为已签到。前台的显示需要根据该字段判断签到字段是否需要显示。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 报名id | 根据报名id去更新报名表的是否签到标志，并根据报名表中的活动id更行活动的签到数+1 | 签到是否成功 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 学生id | StuId | int | Int(20) |
| 活动id | ActId | int | Int(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 报名是否成功 | Msg | String | 无 |

## 算法

无

## 流程逻辑

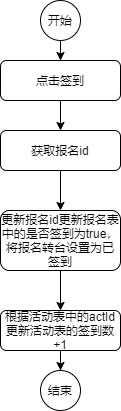


图8：签到流程

## 接口

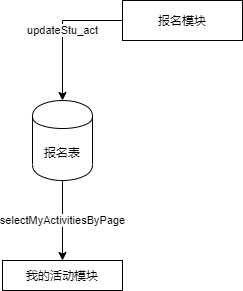


图9：签到接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序5（消息）设计说明

## 程序描述

当学院管理员发送了某个通知，需要通知改学院的所有学生。当活动逾期还没有签到时，需要给对应的同学发送一个消息。可通过消息id查询某个消息，更新消息为已读，删除消息；可根据关键字和页码来分页显示消息查看。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 消息id | 根据消息id可以查看消息 | 消息对象 |
| 消息id | 更新消息为已读，删除消息 | 是否已读，是否删除成功 |
| 搜索关键字，分页信息 | 进行带关键字的模糊分页查询，并将总记录数，总页码，每页的数据集合，当前页码，每页显示的数据数存入PageBean对象并返回前端 | 消息的PageBean对象 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 消息id | MesId | int | Int(20) |
| 关键字 | searchString | String | Varchar(30) |
| 总页码 | totalPage | Int | Int(20) |
| 当前页码 | currentPage | Int | Int(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 消息对象 | notice | Notice | 无 |
| 消息的pageBean对象 | NoticePageBean | PageBean | 无 |
| 回馈消息 | msg | String | 无 |

## 算法

分页查询时采用的limit后跟的两个数，一个时开始的位置start，一个是pageSize，其中，开始的位置start的计算方法是：(currentPage-1)\*pageSize;

还需要求总页码totalPage，此时需要的计算方法是：

totalPage = (totalCount % pageSize ) == 0 ? totalCount/pageSize : (totalCount/pageSize)+1;

算法就是用总记录数除以每页的记录数得到余数，如果余数是0，总页码数就是之前的商，如果余数不是0，总页码数就是之前的商+1

## 流程逻辑

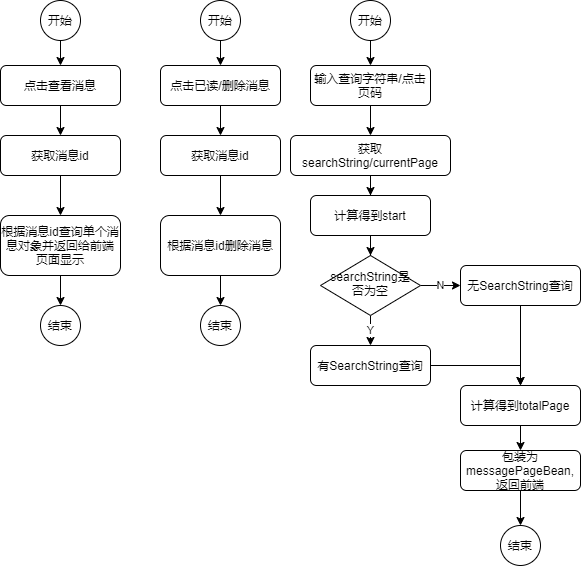


图10：消息流程

## 接口

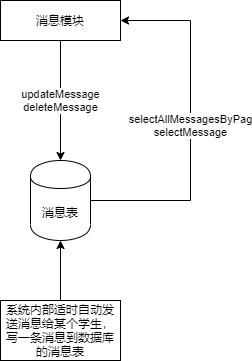


图11:消息接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

说明在本程序的设计中尚未解决而设计者认为在软件完成之前应解决的问题。

# 程序6（留言）设计说明

## 程序描述

先填写留言内容，然后点击提交，会获取奥活动的id，然后再后台自动获取当前系统的时间，然后将留言内容和留言时间一起写入数据库的报名表中。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 留言内容，报名id | 根据报名id来更新报名表，将当前时间，留言内容更新到报名表 | 留言是否成功 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 留言内容 | wordsContent | String | Varchar(20000) |
| 报名id | Stu\_actId | Int | Int(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 回馈消息 | msg | String | 无 |

## 算法

无

## 流程逻辑

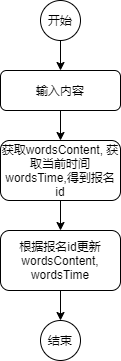


图12：留言流程

## 接口

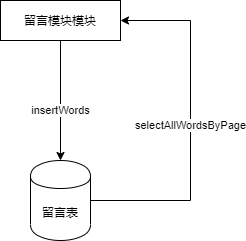


图13：留言接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序7（登陆注册）设计说明

## 程序描述

学生注册时选择是党员还是团员，选择是否是班级团支书，选择学院，班级，设置密码，将这些信息写入学生表中，根据自增长的id加上一个基数得到学号（也就是登录账号，学生在我的信息界面可查看）

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 账号，密码 | 将账号和密码作为条件去查询学生表，根据返回结构是否为空来判断你是否登录成功 | 登录是否成功 |
| 学院，班级，身份信息，密码，验证码 | 先判断验证码是否合规将这些信息写入学生表，再根据自动生成的自增长的id得到学号并更新学生表 | 注册是否成功 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 账号 | StuNum | String | Varchar(30) |
| 密码 | Password | String | Varchar(30) |
| 学院名 | collegeName | String | Varchar(30) |
| 班级名 | \_class | String | Varchar(30) |
| 团员/党员 | Identity | Int | Int(20) |
| 是否团支书 | isTuan | Int | Int(20) |
| 验证码 | checkCode | Stirng | 4位字符 |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 反馈信息 | Msg | String | 无 |

## 算法

注册时，首先在前端判断各个字段的格式是否符合规范，在都符合规范的情况下，将数据传到后端，在后端首先判断该班级是否已有团支书，然后判断验证码是否合规，如果合规就进行数据库写入。

登录时，首先在前端判断各个字段的格式是否符合规范，在都符合规范的情况下，将数据传到后端，如果选择的时团支书。然后判断验证码是否合规，如果合规再根据学号的密码进行学生表的查询。

## 流程逻辑

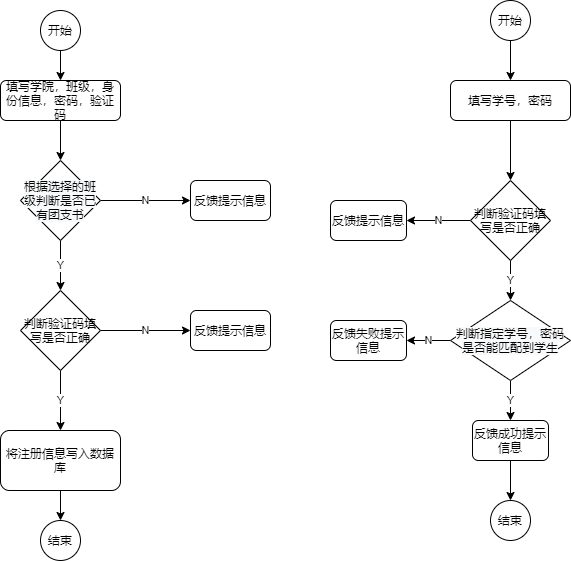


图14：登陆注册流程

## 接口

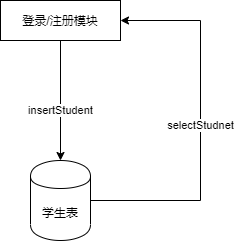


图15：登陆注册接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

本模块的学生注册并不能注册人的身份都是学生，这会造成系统的身份验证出现很大的难度。实际在现实中，学生信息需要从学校的教务系统获取。

# 程序8（申请活动）设计说明

## 程序描述

团支书在活动申请界面填写活动的信息（班级名，活动标题，活动类型，活动报名得分，活动开始日期时间，活动结束时间日期，活动地点，报名人数上限，报名截止日期时间，活动内容）发送到后台，写入数据库

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 活动各种信息 | 将前端所有提交数据写入数据库的活动表 | 申请是否成功 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 活动标题 | actTitle | String | Varchar(30) |
| 活动类型 | Category | int | Int(20) |
| 活动报名得分 | getPoints | int | Int(20) |
| 活动开始时间 | actBeginTime | String | Varchar(30) |
| 活动结束时间 | actEndTime | String | Varchar(30) |
| 活动地点 | actPlace | String | Varchar(100) |
| 报名人数上限 | numLevel | int | Int(20) |
| 报名截止时间 | enrollEndTime | String | Varchar(30) |
| 活动内容 | actContent | String | Varchar(20000) |
| 班级名 | \_calss | String | Varchar(30) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 反馈信息 | Msg | String | 无 |

## 算法

无

## 流程逻辑

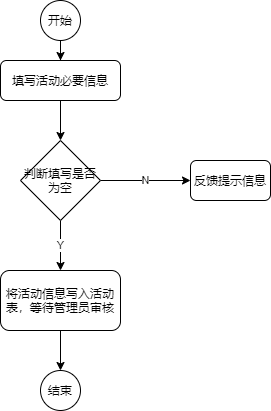


图16：申请活动流程

## 接口

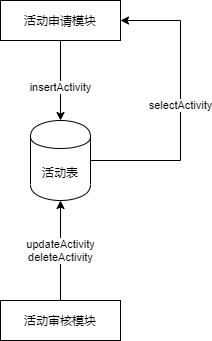


图17：申请活动接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序9（统计学生）设计说明

## 程序描述

在团支书界面能分页显示本班级的所有学生以及一些必要信息

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 分页信息 | 将总记录数，总页码，每页的数据集合，当前页码，每页显示的数据数存入PageBean对象并返回前端 | 通知的PageBean对象 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 总页码 | totalPage | Int | Int(20) |
| 当前页码 | currentPage | Int | Int(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 分页对象 | StudnetPageBean | PageBean | 无 |

## 算法

分页查询时采用的limit后跟的两个数，一个是开始的位置start，一个是pageSize，其中，开始的位置start的计算方法是：(currentPage-1)\*pageSize;

还需要求总页码totalPage，此时需要的计算方法是：

totalPage = (totalCount % pageSize ) == 0 ? totalCount/pageSize : (totalCount/pageSize)+1;

算法就是用总记录数除以每页的记录数得到余数，如果余数是0，总页码数就是之前的商，如果余数不是0，总页码数就是之前的商+1

## 流程逻辑

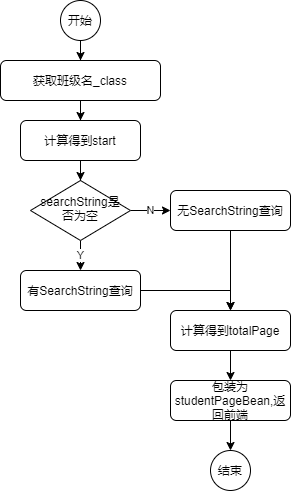


图18：统计学生流程

## 接口

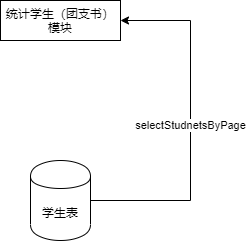


图19：统计学生接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序10（统计学生&活动）设计说明

## 程序描述

在学院管理员界面能分班分页显示的所有学生以及一些必要信息，以及显示所有自己发布的和已经审核通过的活动。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 班级关键字，分页信息 | 进行带关键字的精确分页查询，并将总记录数，总页码，每页的数据集合，当前页码，每页显示的数据数存入PageBean对象并返回前端 | 通知的PageBean对象 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 班级关键字 | \_class | String | Varchar(30) |
| 总页码 | totalPage | Int | Int(20) |
| 当前页码 | currentPage | Int | Int(20) |

。

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 分页对象 | StudnetPageBean  ActivityPageBean | PageBean | 无 |

## 算法

分页查询时采用的limit后跟的两个数，一个是开始的位置start，一个是pageSize，其中，开始的位置start的计算方法是：(currentPage-1)\*pageSize;

还需要求总页码totalPage，此时需要的计算方法是：

totalPage = (totalCount % pageSize ) == 0 ? totalCount/pageSize : (totalCount/pageSize)+1;

算法就是用总记录数除以每页的记录数得到余数，如果余数是0，总页码数就是之前的商，如果余数不是0，总页码数就是之前的商+1

## 流程逻辑

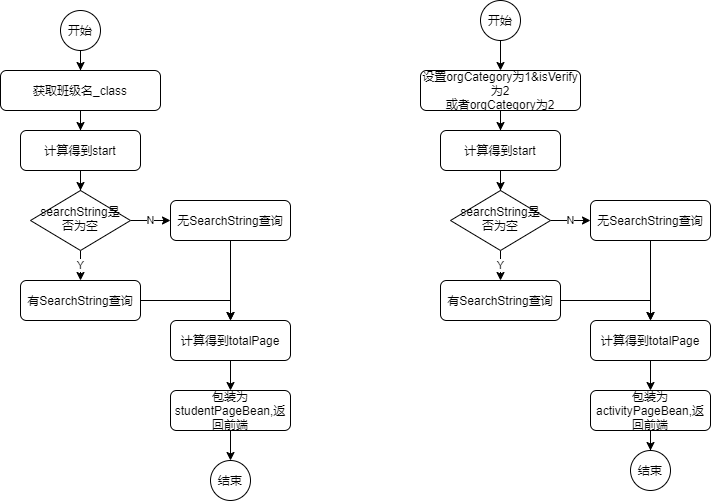


图20:统计学生&活动流程

## 接口

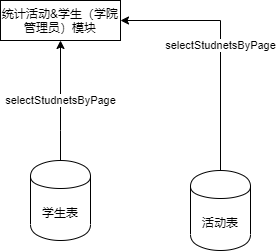


图21：统计学生&活动接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序11（审核活动）设计说明

## 程序描述

学院管理员审核本学院本机提交的活动申请，点击审核通过会更新活动的审核状态为1，点击审核不通过会直接在数据库中删除活动并给该活动所属的团支书发一个消息。并且审核失败的时候需要发送一封信息给团支书

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 审核方案（Y/N）  活动id | 按照审核的方案来进行相关的活动表的操作（更新活动表&删除活动表） | 审核结果反馈信息 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 活动id | actId | Int | Int(20) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 审核结果反馈信息 | Msg | String | 无 |

## 算法

无

## 流程逻辑

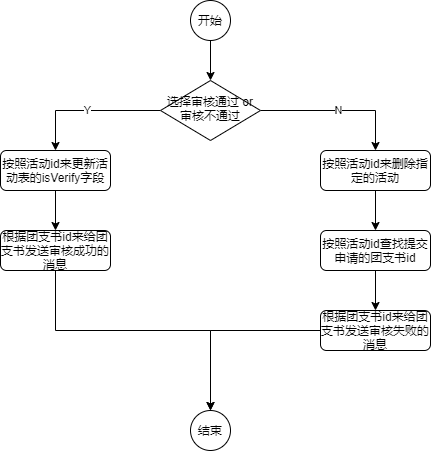


图22：审核活动流程

## 接口

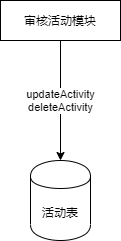


图23：审核活动接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序12（发布活动）设计说明

## 程序描述

学院管理员填写活动的信息（班级名，活动标题，活动类型，活动报名得分，活动开始日期时间，活动结束时间日期，活动地点，报名人数上限，报名截止日期时间，活动内容）发送到后台，写入数据库。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 活动各种信息 | 将前端所有提交数据写入数据库的活动表 | 申请是否成功 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 活动标题 | actTitle | String | Varchar(30) |
| 活动类型 | Category | int | Int(20) |
| 活动报名得分 | getPoints | int | Int(20) |
| 活动开始时间 | actBeginTime | String | Varchar(30) |
| 活动结束时间 | actEndTime | String | Varchar(30) |
| 活动地点 | actPlace | String | Varchar(100) |
| 报名人数上限 | numLevel | int | Int(20) |
| 报名截止时间 | enrollEndTime | String | Varchar(30) |
| 活动内容 | actContent | String | Varchar(20000) |
| 管理员id | ManagerId | Int | int(30) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 反馈信息 | Msg | String | 无 |

## 算法

无

## 流程逻辑

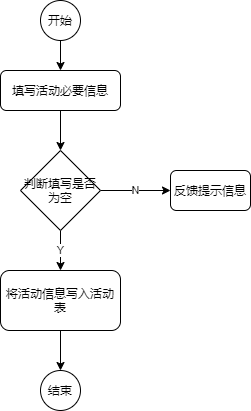


图24：发布活动流程

## 接口

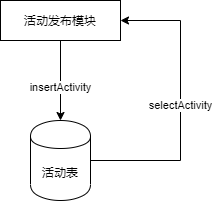


图25：发布活动接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无

# 程序13（发布通知）设计说明

## 程序描述

学院管理员发布关于本学院党团活动的相关通知，填写通知标题，通知内容，获取通知时间写入数据库，并通知发布消息给该学院的所有学生。

## 功能

|  |  |  |
| --- | --- | --- |
| 输入 | 处理 | 输出 |
| 通知标题，通知内容，通知时间，管理员id | 从前台获取通知标题，通知内容，在后台获取通知时间，连同学院管理员id，一起写入数据库的通知表 | 发布通知反馈信息 |

## 输人项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 通知标题 | NoticeTitle | String | Varchar(1000) |
| 通知内容 | noticeContent | String | Varchar(20000) |
| 通知时间 | noticeTime | String | Varchar(30) |
| 管理员id | managerId | String | Varchar(30) |

## 输出项

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 数据范围 |
| 反馈信息 | Msg | String | 无 |

## 算法

无

## 流程逻辑

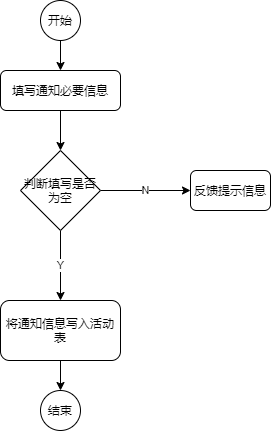


图26：发布通知流程

## 接口

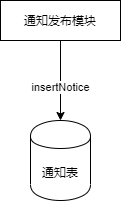


图27：发布互动接口

## 存储分配

程序仅涉及若干局部变量，没有特殊的存储要求。

## 限制条件

无

## 尚未解决的问题

无