四、概要设计说明书

[1．引言 2](#_Toc506972944)

[1.1编写目的 2](#_Toc506972945)

[1.2项目背景 2](#_Toc506972946)

[1.3定义 2](#_Toc506972947)

[1.4参考资料 2](#_Toc506972948)

[2．任务概述 2](#_Toc506972949)

[2.1目标 2](#_Toc506972950)

[2.2运行环境 2](#_Toc506972951)

[2.3需求概述 2](#_Toc506972952)

[2.4条件与限制 2](#_Toc506972953)

[3．总体设计 2](#_Toc506972954)

[3.1处理流程 2](#_Toc506972955)

[3.2总体结构和模块外部设计 2](#_Toc506972956)

[3.3功能分配 2](#_Toc506972957)

[4．接口设计 2](#_Toc506972958)

[4.1外部接口 2](#_Toc506972959)

[4.2内部接口 2](#_Toc506972960)

[5．数据结构设计 2](#_Toc506972961)

[5.1逻辑结构设计 2](#_Toc506972962)

[5.2物理结构设计 2](#_Toc506972963)

[5.3数据结构与程序的关系 2](#_Toc506972964)

[6．运行设计 2](#_Toc506972965)

[6.1运行模块的组合 2](#_Toc506972966)

[6.2运行控制 2](#_Toc506972967)

[6.3运行时间 2](#_Toc506972968)

[7．出错处理设计 2](#_Toc506972969)

[7.1出错输出信息 2](#_Toc506972970)

[7.2出错处理对策 2](#_Toc506972971)

[8.安全保密设计 2](#_Toc506972972)

[9.维护设计 2](#_Toc506972973)

# 1．引言

## 1.1编写目的

本阶段已在系统的需求分析的基础上，对软件学院实训过程项目与任务跟踪系统做概要设计。主要解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明。

在下一阶段的详细设计中，程序设计员可参考此概要设计报告，在概要设计对机票预定系统所做的模块结构设计的基础上，对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书，以便于了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

## 1.2项目背景

该项目是基于四川大学软件学院暑期实训的背景产生的。实训工程师和学员需要一个通用的网络平台来实现相关信息的通知和实训作业（项目文档）的提交，其实现的主要目的有：

1. 便于实训工程师对于整体教学进度的把控，以及对于学员项目进展情况的实时了解
2. 便于同一项目组的同学共同维护和推进项目
3. 用于展示教学计划，项目相关文档等，在整个实训过程中具有公示栏作用。

此外，在实训结束以后，该系统还具有：

1. 通过每一小组的文档得分，可以计算学员的实训分数
2. 教学计划制定者可以通过此系统了解教学计划的合理性
3. 实训策划者可以通过此系统了解学员的学习情况，实训效果

等意义。

## 1.3定义

## 1.4参考资料

[1] Software Engineering A Practitioners Approach 8th Edition

[2] 软件构架实践【第三版】

[3] 数据库设计Step by Step (DBFocus博客文集)

# 2．任务概述

## 2.1目标

实现一个高可用性、高易用性、高可扩展性的系统，并能够完整地实现需求文档中所描述的各类需求。在实际应用中能够真正帮助实训工程师和学员管理班级、跟进教学计划、管理项目和文档。

## 2.2运行环境

操作系统：Microsoft Windows 7,10; Mac OS X  
前端浏览器：Google Chrome浏览器  
后台支持环境：Java 1.8+  
数 据 库：MySQL 8.0+

## 2.3需求概述

本系统面向超级管理员（学校与公司管理人员），实训工程师和学员这三类直接用户，功能需求主要分为项目库管理，文档库管理，学员、班级管理，工程师核心，学员核心这几部分。超级管理员需要实现对项目库的管理和登录学员、工程师信息、创建班级等功能，工程师主要需要实现对班级的管理以及对班内项目开展情况的跟进，而学生主要需要实现对文档库的维护功能。此外，本系统还需要兼顾性能需求和运行需求等其他非功能需求。

## 2.4条件与限制

* 方便操作，操作流程合理

尽量从用户角度出发，以方便使用本产品。

* 支持没有计算机使用经验、计算机使用经验较少及有较多计算机使用经验的用户均能方便地使用本软件。容错能力
* 系统具有一定的容错和抗干扰能力，在非硬件故障或非通讯故障时，软件能够保证正常运行，并有足够的提示信息帮助用户有效正确地完成任务。
* 用户可自定义

为了满足业务的不断变化，一些重要的参数应该可以灵活设置。

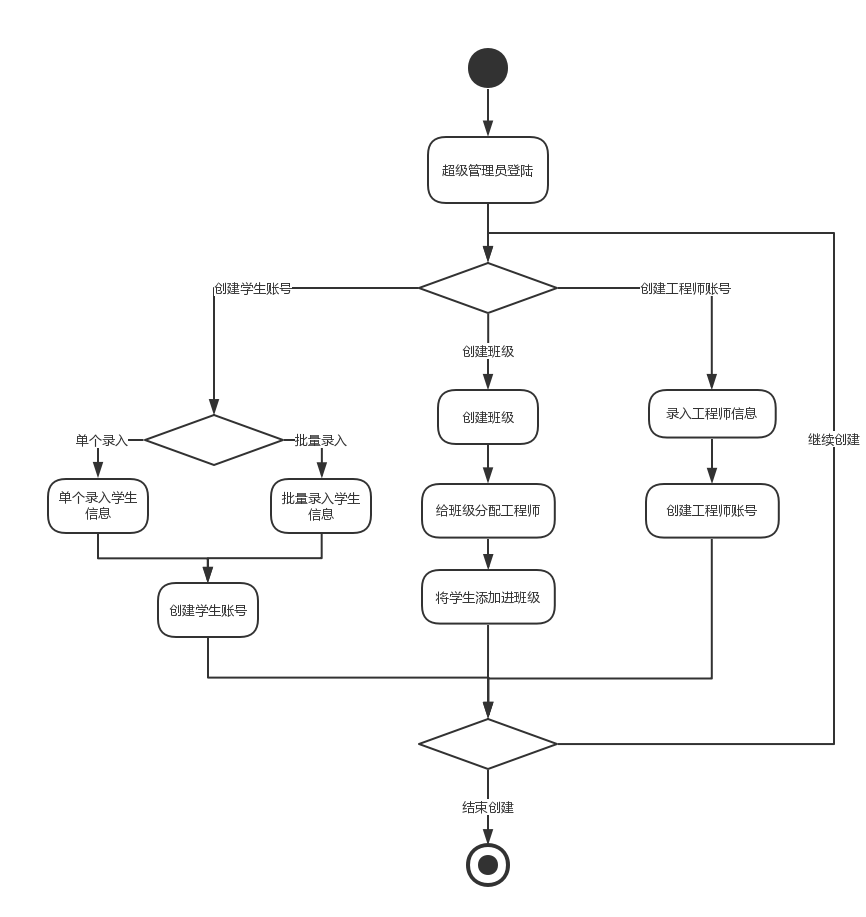
联机帮助与操作指南

# 3．总体设计

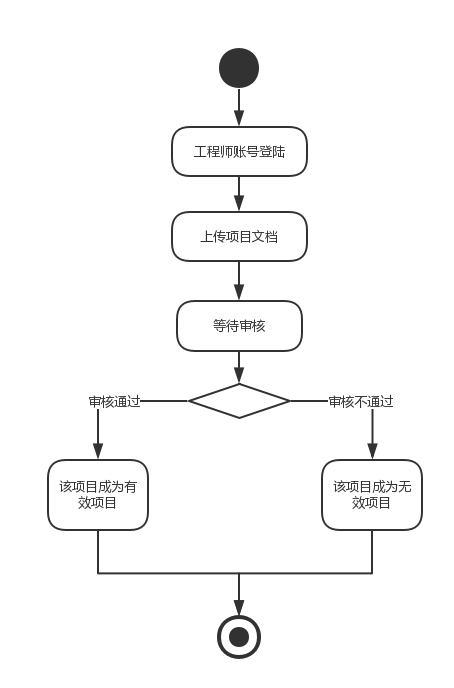
## 3.1处理流程

下面用UML活动图描述程序主要的流程，由于程序功能较多，所以省略部分简单流程。

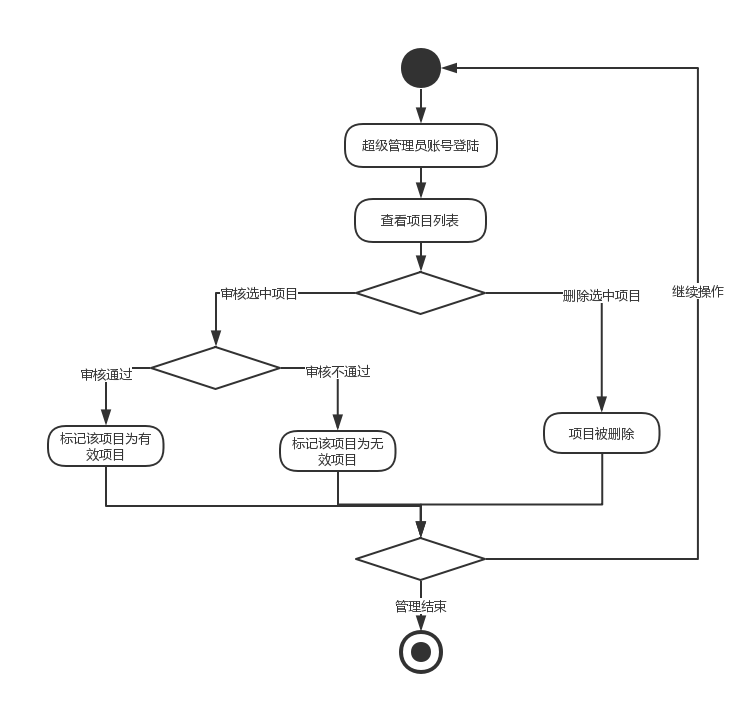
（1）超级管理员创建账号与班级流程



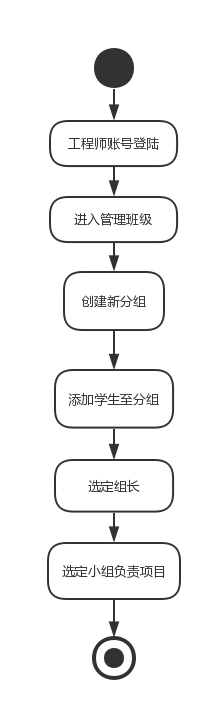
（2）项目提出与审核流程



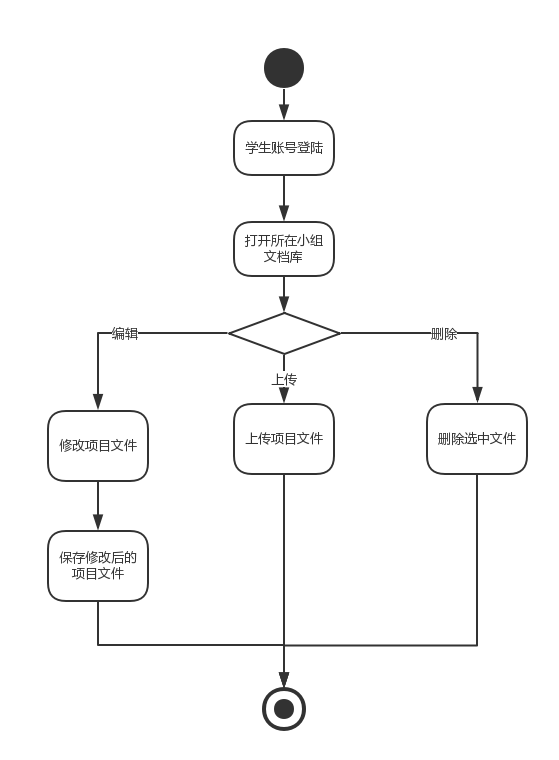
（3）项目管理流程



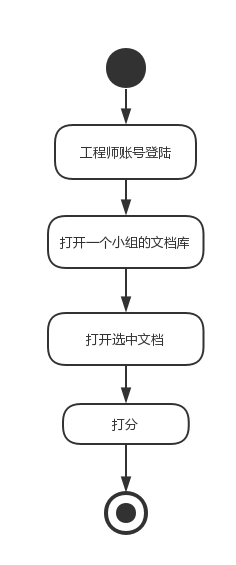
（4）分组流程



（5）文档提交和编辑流程



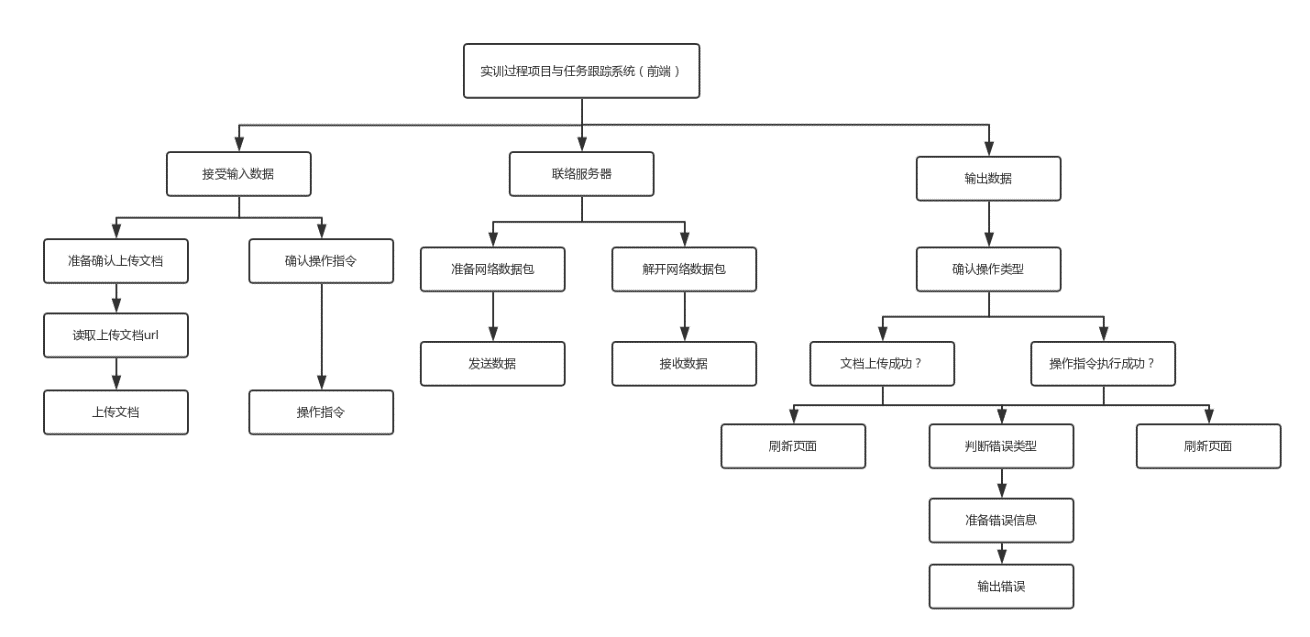
（6）评分流程



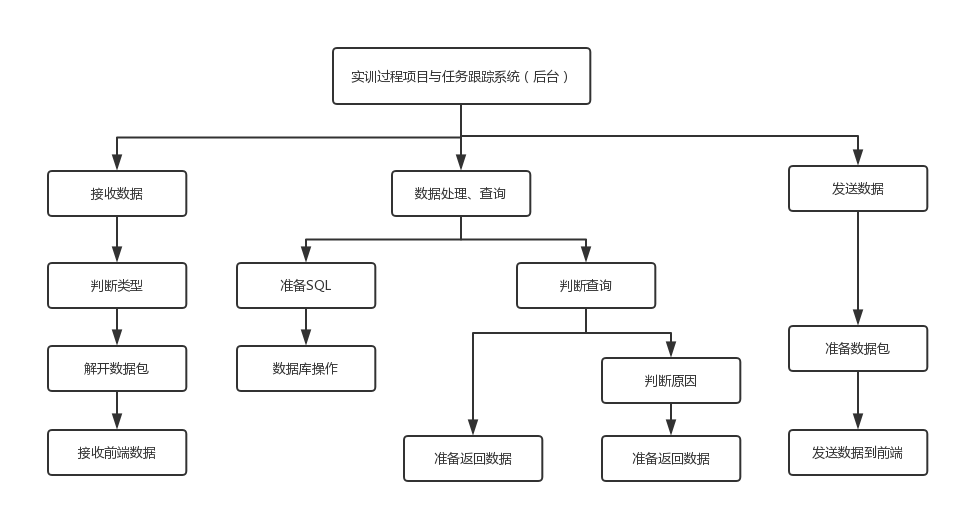
## 3.2总体结构和模块外部设计

程序总体可分为前端和后台两部分，下面以结构图来描述实训过程项目与任务跟踪系统的软件总体结构。框内注明了模块的名字；方框之间的箭头表示模块的调用关系。

**3.2.1 前端**



**3.2.2 后台**



## 3.3功能分配

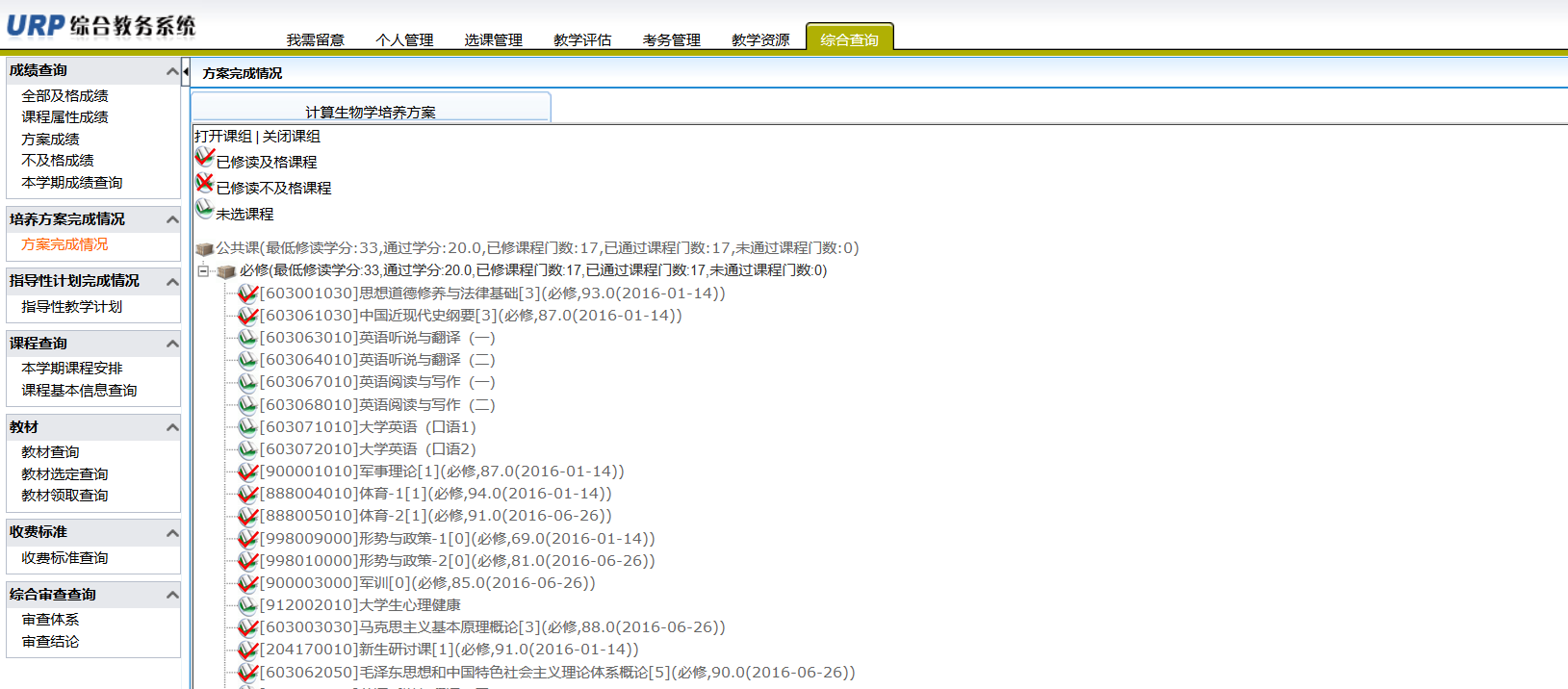
各项模块的功能可参照3.1 中的说明。前端程序主要有三大块：接收数据、网络通信及输出部分。后台程序主要也是由三大功能：接收网络数据、数据库操作及发送网络数据部分。后台程序需与已建立的SQL SERVER 数据库互连，其接口将于下面部分阐述。

# 4．接口设计

## 4.1外部接口

4.1.1 用户界面

在用户界面部分，根据需求分析的结果，用户需要一个用户友善界面。在界面设计上，应做到简单明了，易于操作，并且要注意到界面的布局，应突出的显示重要以及出错信息。外观上也要做到合理化，考虑到作为用户的管理员、工程师和学员都有项目管理/文档管理的需求，且大家多对github的项目管理风格较熟悉，应尽量向这一方向靠拢。另，菜单界面应效仿URP综合教务系统的界面设计（如下图），在简洁大方的同时，可以展示足够的信息量。



总的来说，系统的用户界面应作到可靠性、简单性、易学习和使用。

4.1.2 软件接口

服务器程序可使用Java提供的对 MySQL SERVER 的接口，进行对数据库的所有访问。

服务器程序上可使用MySQL SERVER 的对数据库的备分命令，以做到对数据的保存。

在网络软件接口方面，使用一种无差错的传输协议，采用滑动窗口方式对数据进行网络传输及接收。

4.1.3 硬件接口

在输入方面，对于键盘、鼠标的输入，可用Java的标准输入/输出，对输入进行处理。

在输出方面，打印机的连接及使用，也可用Java的标准输入/输出对其进行处理。在网络传输部分.

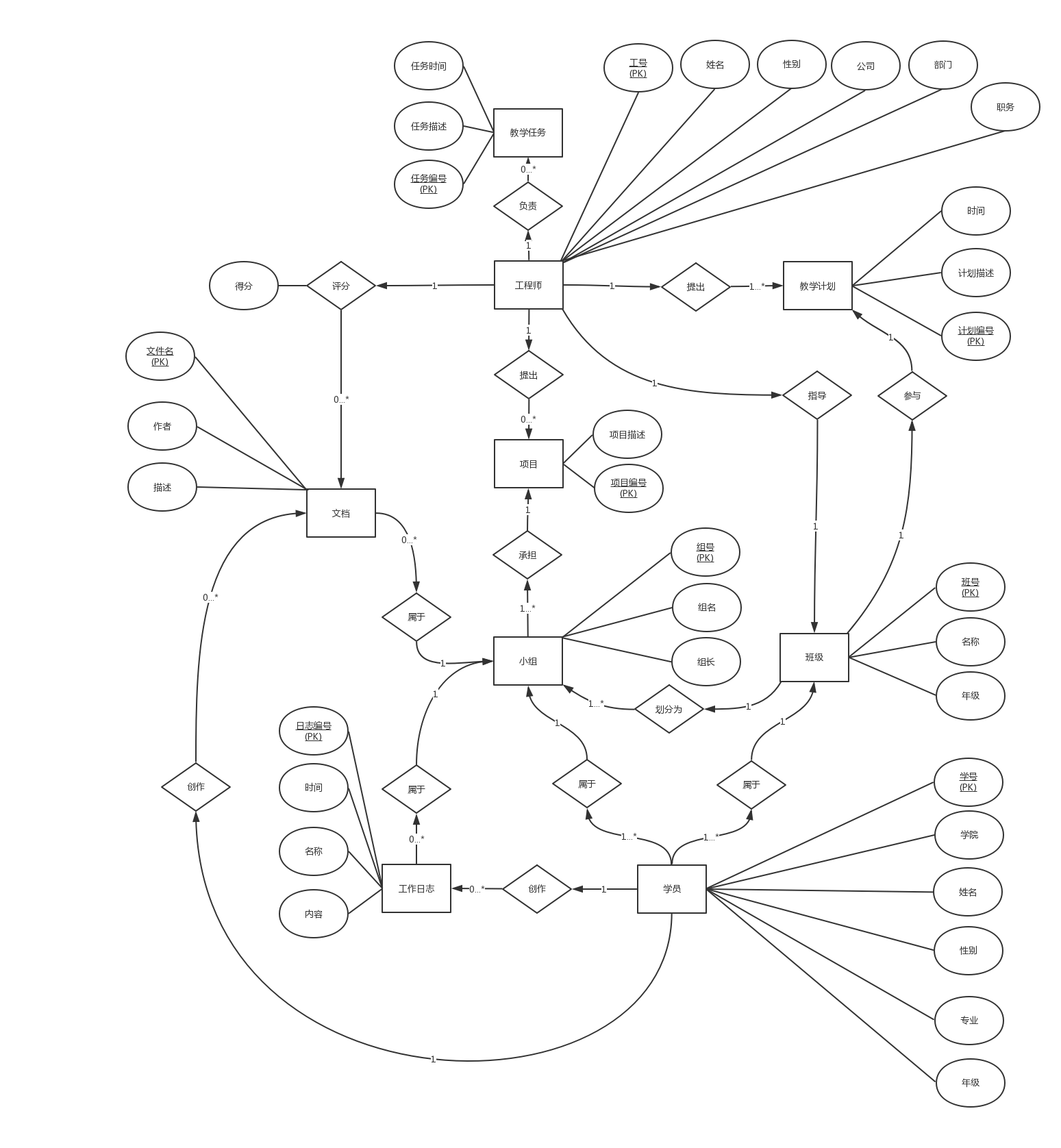
## 4.2内部接口

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传递。具体参数的结构将在下面数据结构设计的内容中说明。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。

# 5．数据结构设计

## 5.1逻辑结构设计

根据需求分析，绘制出的E-R图如下：



根据此E-R图导出的一系列需要数据库系统维护的数据表如下（下划线加粗为主键，仅下划线为外键）：

1. 工程师信息表：

工程师 (**工号**, 姓名, 性别, 公司, 部门, 职务)

工程师信息表包含了管理系统中所有工程师的详细信息，包括其工号及姓名等个人信息。其中，以每位工程师的工号作为主键。当超级管理员创建/删除工程师、更改工程师信息时需要对本表进行修改。

1. 项目信息表：

项目 (**项目编号**, 项目名称, 项目描述, 通过, 工号)

项目信息表包含了管理系统中由工程师创建的所有的项目的信息，包括项目名称、项目描述以及项目是否审核通过。其中，项目编号为主键，创建该项目的工程师的工号为外键。当工程师创建新的项目以及更改项目信息时，需要对本表进行修改。

1. 班级信息表：

班级 (**班级编号**, 班级名称, 年级, 工号, 教学计划编号)

班级信息表包括了超级管理员分班之后各班级的信息，包括班级的名称以及年级等。其中，班级编号为主键，负责该项目的工程师的工号以及这一班级实施的教学计划的编号为两外键。当超级管理员进行分班操作以及分配负责该班级的工程师以及适用的教学计划时，需要对本表进行修改。

1. 班级分组信息表

小组 (**小组编号**, 小组名称, 小组描述, 学号, 班级编号)

班级分组信息表包括了工程师对自己负责的班级进行分组之后的信息，包括各小组的名称以及小组描述。其中，小组编号为主键，该组组长的学号以及该组所对应的班级的编号为两外键。当工程师对负责的班级进行分组操作时，需要对本表进行修改。

1. 学员信息表

学员 (**学号**, 姓名, 性别, 年级, 学院, 专业, 班级编号, 小组编号)

学员信息表包括了各学员的信息如姓名等个人信息。其中，学员的学号为主键，该学员所在的班级编号以及小组编号为两外键。当超级管理员进行学员信息的导入等操作时，需要对本表进行修改。

1. 文档信息表

文档 (**文件名**, 小组编号, 学号, 描述, 评分)

文档信息表包括了在项目进行过程中各小组成员（学员）上传的文档的信息。其中，以文件名为主键，上传该文件的学员所在的小组的编号以及此学员的学号为两外键。工程师对文档的打分信息也保存在这一表中。当学员上传文档以及工程师对文档内容进行打分时，需要对本表进行修改。

1. 教学任务信息表

教学任务 (**工号, 教学任务编号,** 任务名称, 任务描述, 制定时间)

教学任务信息表包括了各工程师需要完成的教学任务的信息，如任务名称、任务描述和制定时间等。其中，对应工程师的工号以及教学任务编号共同组成该表的主键，其中工号也是外键。当超级管理员为工程师制定任务时，需要对本表进行修改。

1. 教学计划信息表

教学计划 (**工号, 计划编号,** 时间, 计划名称, 计划描述)

教学计划信息表包括了各工程师制定的教学计划的信息，如计划名称、计划描述和时间等。其中，对应工程师的工号以及教学计划编号共同组成该表的主键，其中工号也是外键。当工程师添加/修改/删除教学计划时，需要对本表进行修改。

1. 工作日志信息表

工作日志 (**日志编号**, 时间戳, 日志名称, 日志内容, 小组编号, 学号)

工作日志信息表中包括各小组在项目进行过程中添加的工作日志的信息，如日志编号、添加的时间、日志的名称和内容等。其中，日志编号为主键，该日志对应的小组的编号以及创建该日志的学员的学号为两外键。当学员添加日志时，会对本表进行修改。

## 5.2物理结构设计

在5.1节中叙述的数据库中各数据表的模式的物理描述如下：

1. 工程师 - Engineer
   1. **工号**, int
   2. 姓名, String
   3. 性别, Enum(“男”,“女”)
   4. 公司, String
   5. 部门, String
   6. 职务, String
2. 项目 - Project
   1. **项目编号**, int
   2. 项目名称, String
   3. 项目描述, String
   4. 审核通过, boolean
   5. 工号, int
3. 班级 - Class
   1. **班级编号**, int
   2. 班级名称, String
   3. 年级, int
   4. 工号, int
   5. 教学计划编号, int
4. 小组 - Team
   1. **小组编号**, int
   2. 小组名称, String
   3. 小组描述, String
   4. 学号, int
   5. 班级编号, int
5. 学员 - Student
   1. **学号**, int
   2. 姓名, String
   3. 性别, Enum(“男”,“女”)
   4. 年级, int
   5. 学院, String
   6. 专业, String
   7. 班级编号, int
   8. 小组编号, int
6. 文档 - Doc
   1. **文件名**, String
   2. 小组编号, int
   3. 学号, int
   4. 描述, String
   5. 评分, float
7. 教学任务 - Task
   1. **工号,** int
   2. **教学任务编号,** int
   3. 任务名称, String
   4. 任务描述, String
   5. 制定时间, Date
8. 教学计划 - Plan
   1. **工号,** int
   2. **计划编号,** int
   3. 时间, Date
   4. 计划名称, String
   5. 计划描述, String
9. 工作日志 - Log
   1. **日志编号**, int
   2. 时间戳, Date
   3. 日志名称, String
   4. 日志内容, String
   5. 小组编号, int
   6. 学号, int

## 5.3数据结构与程序的关系

程序实现完成不同的业务需要对相对应的数据结构进行操作；下面叙述不同业务流程中对数据结构的操作。

对于超级管理员来讲，当创建工程师/学员账号并添加对应信息或创建班级并填写对应信息时，后台程序会根据前端的输入创建相对应的数据结构并在数据库相对应的表中创建条目；超级管理员审核通过项目后，对项目中“通过”属性的值进行更改。在超级管理员对工程师、学员以及项目信息进行管理时，相对应的数据结构被删除或属性被修改，数据库中做相对应操作。

对于工程师来说，当其创建项目时，后台根据数据创建数据结构并再数据库中插入表项；当进行分组操作时，创建相对应的小组数据结构并在数据库中插入对应表项，并对被分组学员的数据进行修改以分配组；当对不同的提交的文档进行打分时，对文档数据结构以及数据表中对应的“分数”属性进行修改。

对于学员来说，当上传文档或添加了工作日志之后，创建对应的数据结构并在对应数据表中插入表项；当对文档或日志进行修改/删除时，删除对应的数据结构和表项或修改对应的属性。

# 6．运行设计

## 6.1运行模块的组合

## 6.2运行控制

## 6.3运行时间

# 7．出错处理设计

## 7.1出错输出信息

在程序运行过程中，可能会出现以下几种可能的错误。现将其类型以及输出信息列举如下：

1. 数据格式错误

当使用者在输入数据时，可能会产生输入格式上的错误，如为一个int类型的字段提供了String类型的输入。这种类型的错误应该在前端模块中就加以识别与规避，提醒用户检查输入，避免将错误的类型输入传入到后台产生更严重的后果。因此，在出现数据格式错误时，前端页面上会显示产生错误的数据并提醒用户检查其输入。

1. 网络错误

在进行与服务器的交互如提交更改、上传文档等等操作时，可能会遇到网络问题并造成操作超时。此时，应该输出网络错误的提示并提醒用户再次尝试同样操作。

1. 程序内部错误

当出现程序内部错误时，提醒用户产生了暂时无法解决的程序内部错误并在控制台打印错误具体信息以供未来的调试与修补。

1. 服务器运行错误

在非常罕见的情况下服务器的运行可能会出现错误和不稳定的情况。此时，如果探测到有类似情况的发生，应该提醒用户保存更改并尝试重新登陆或启动系统。

## 7.2出错处理对策

在这一部分中，对7.1 中所述的不同类型的错误提出不同的出错处理对策：

1. 数据格式错误

当发生数据格式错误时，直接提醒用户检查并再次输入以规避错误再次发生。

1. 网络错误

当发生网络错误时，提醒用户尝试重新操作或刷新页面；在程序实现中与数据库进行交互的模块中使用事务对数据库进行修改，以保证数据库的数据完整性。

1. 程序内部错误

当出现用户无法解决的程序内部错误时，告知用户并在控制台打印错误产生的相关信息；利用错误发生的信息对程序进行测试和调试，以修补可能的缺陷；对数据库的数据完整性进行检查，以排除数据不完整导致的程序错误。

1. 服务器运行错误

首先，使用稳定性良好的服务器配置；当出现服务器运行不稳定时，提醒用户保存自己的修改。必要时重新启动程序或重新启动服务器。

# 8.安全保密设计

# 9.维护设计

【说明为方便维护工作的设施，如维护模块等。】