

**面向对象分析与设计报告**

**专业班级 信管1401**

**组 别 4组**

**小组成员 撒旭 贠艳冰 王鑫 孔佑然**

2017年6月 7 日

目录

[4.4系统用例建模 3](#_Toc487738829)

[4.4.1 总体用例建模 3](#_Toc487738830)

[4.4.2 子用例建模 3](#_Toc487738831)

[4.4.3 活动图 6](#_Toc487738832)

[4.5 健壮性分析 9](#_Toc487738833)

[4.6 系统静态建模 11](#_Toc487738834)

[4.6.1 系统类图建模 11](#_Toc487738835)

[4.6.2 系统类关系图建模 11](#_Toc487738836)

[4.7 系统顺序图建模 12](#_Toc487738837)

[5.1系统界面设计与实现 15](#_Toc487738838)

[5.1.1登陆页面 15](#_Toc487738839)

[5.1.2主页面 16](#_Toc487738840)

[5.1.3帐号管理页面 16](#_Toc487738841)

[5.1.4项目申报页面 16](#_Toc487738842)

[5.1.5中期检查页面 18](#_Toc487738843)

[5.1.6项目结题页面 18](#_Toc487738844)

[5.1.7审核页面 18](#_Toc487738845)

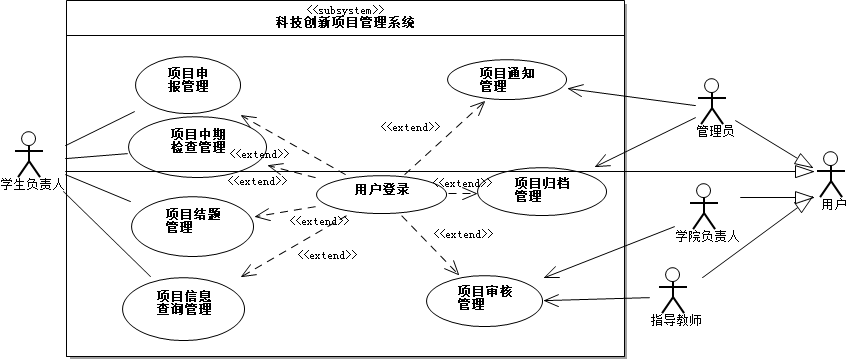
# 4.4系统用例建模

## 4.4.1 总体用例建模

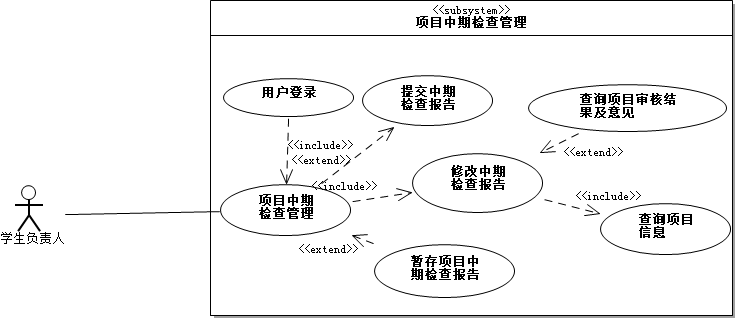
用例图是外部用户（参与者）所能观察到的系统功能的模型图，可以显示系统中的用例与角色及其相互关系，主要用于对系统、子系统或类的功能行为进行建模。用例是系统提供的高级功能块，角色是与所建系统交互的对象。

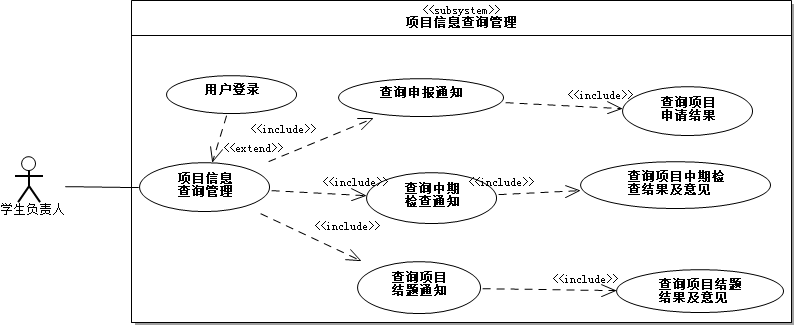
用例图展示了用例之间以及同用例参与者之间是怎样相互联系的。用例图用于对系统、子系统或类的行为进行可视化，使用户能够理解如何使用这些元素，并使开发者能够实现这些元素。

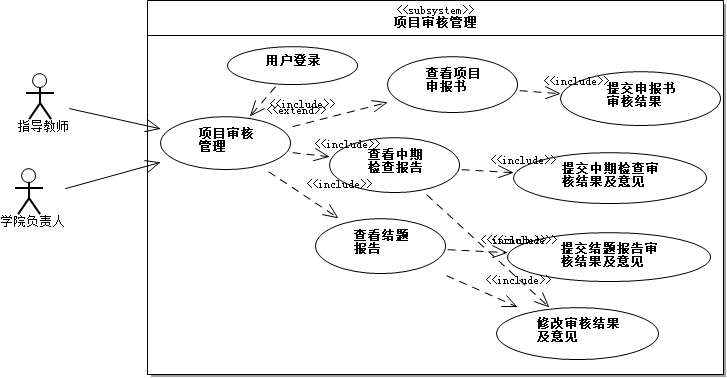
如图：

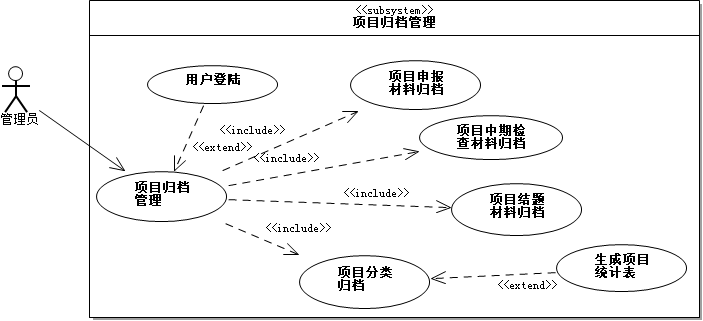


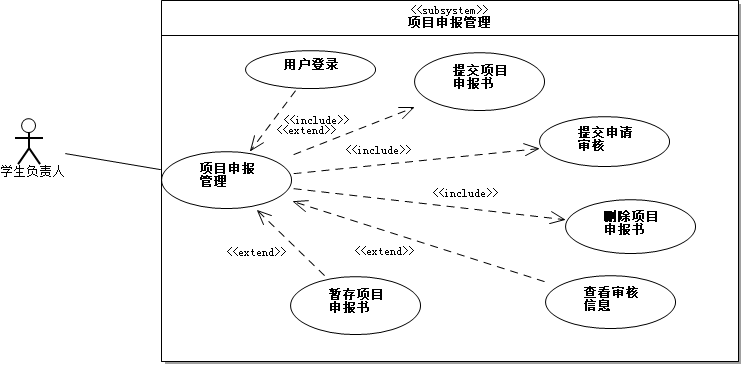
## 4.4.2 子用例建模

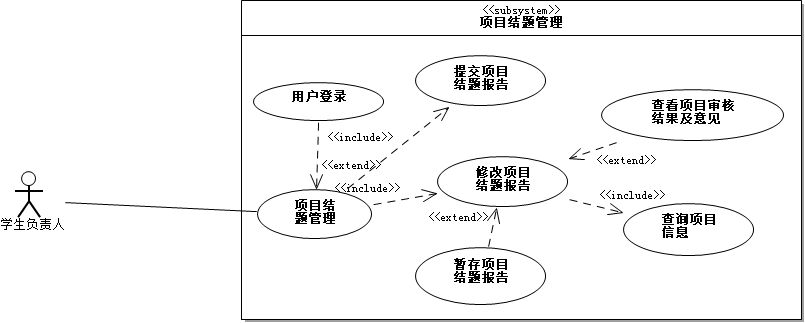


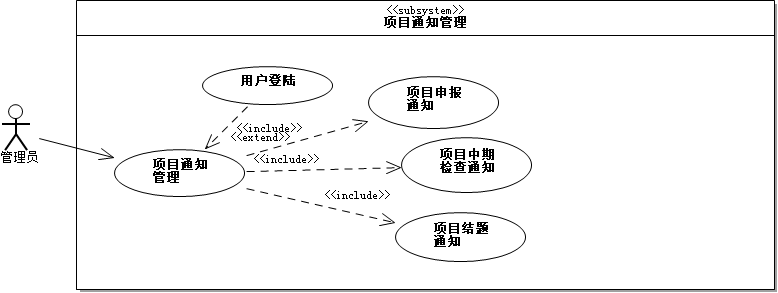












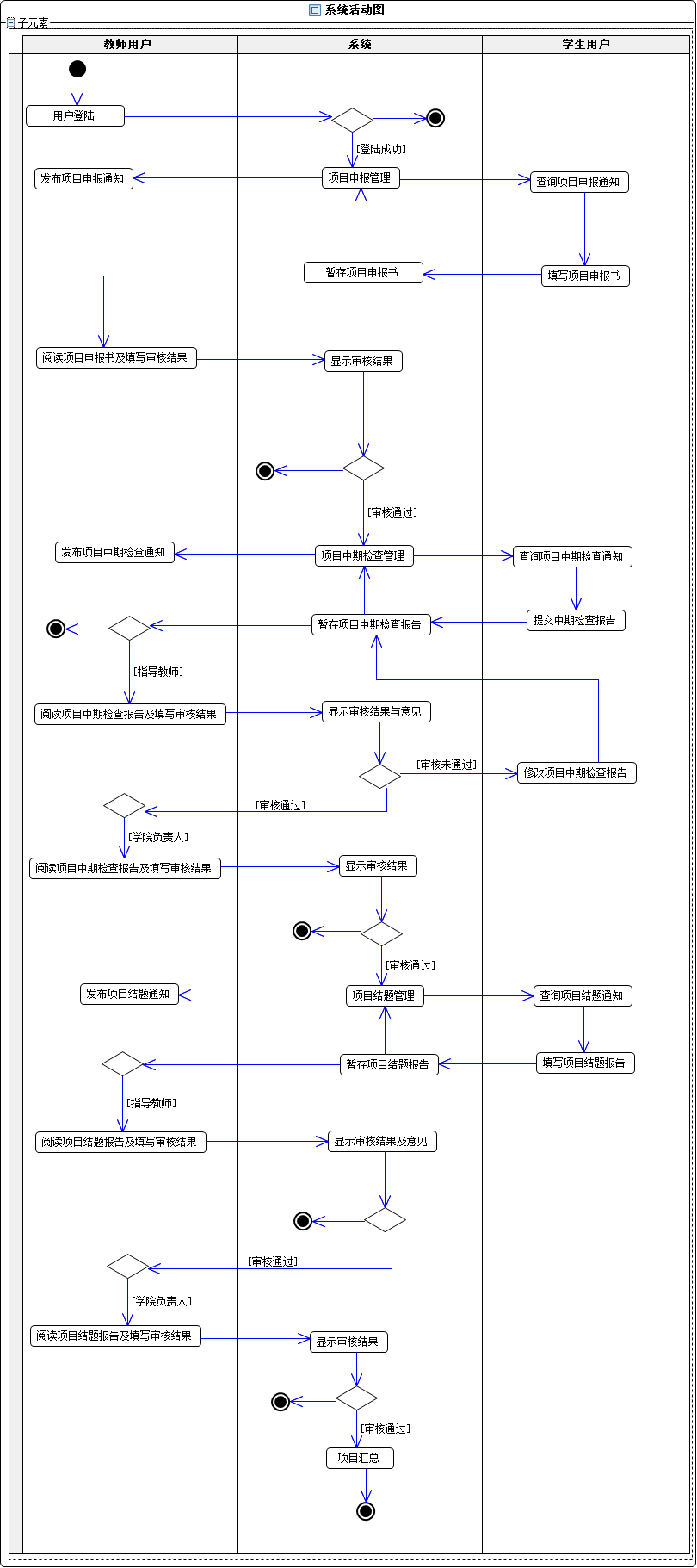
## 4.4.3 活动图

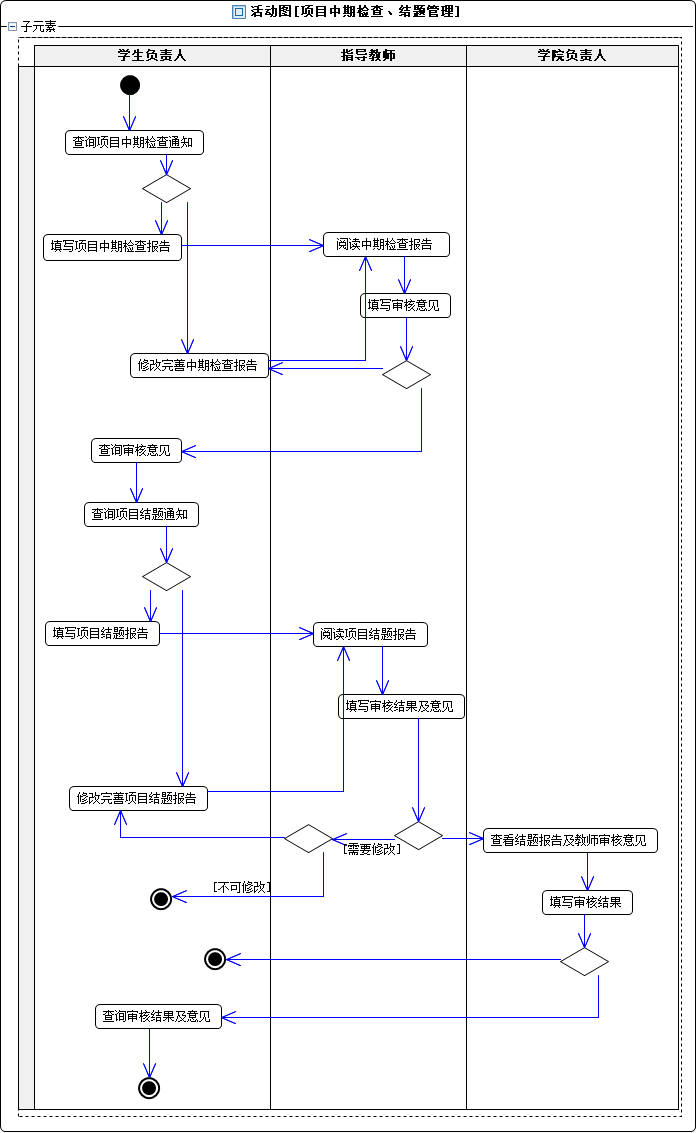
活动图是用来描述一系列顺序动作、结果及其它们之间关系的图，需求分析阶段可以来和用户交流，以辅助需求采集。

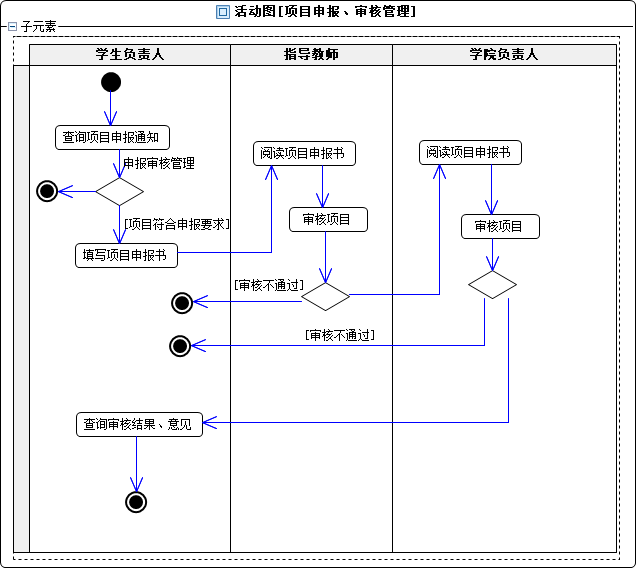
活动图的典型应用：

用例中控制流和用例间的控制流，操作和算法的细节以及业务流程。

如图：







# 4.5 健壮性分析

健壮性又称[鲁棒性](http://baike.baidu.com/item/%E9%B2%81%E6%A3%92%E6%80%A7)，是指软件对于规范要求以外的输入情况的处理能力。

所谓健壮的系统是指对于规范要求以外的输入能够判断出这个输入不符合规范要求，并能有合理的处理方式。

健壮性分析的建模元素：边界对象、控制对象、实体对象、角色

（1）边界对象

一个系统要和系统外的其它事物进行信息交流，需要通过边界对象，分为人机边界和其他系统交互的系统边界。

主要有：主界面、项目申请界面、项目结题界面、项目中期检查界面、学校审核情况界面、教师审核情况界面、查询通知界面、发布通知界面、登录界面

（2）控制对象

被用来封装一些复杂的控制行为，可能变化的行为以及需要事务管理的行为；或者作为一种桥梁，连接边界对象和实体对象。

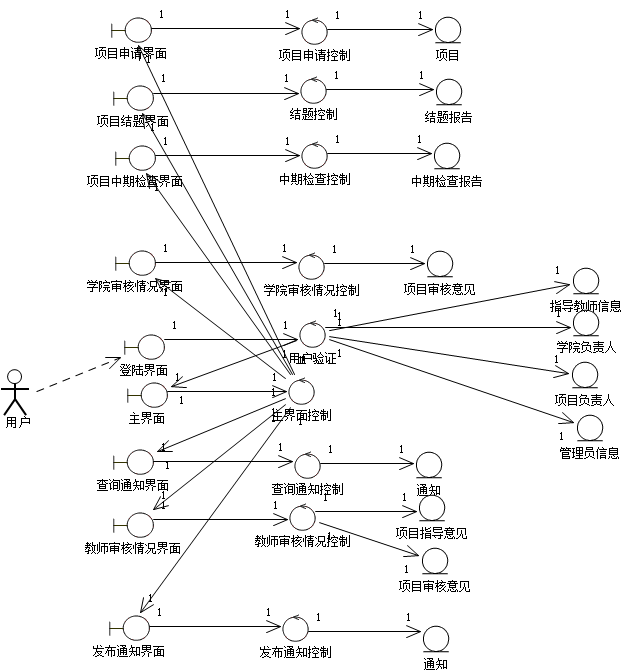
主要有：主界面控制、用户验证、中期检查控制、查询通知控制、结题控制、发布通知控制、学院审核情况控制、项目申请控制、教师审核情况控制

（3）实体对象

在永久保存媒体上永远保存的“类”的快照或映像，数据来自边界对象。

主要有：项目、结题报告、中期检查报告、项目审核意见、用户、通知、项目指导意见

如图：



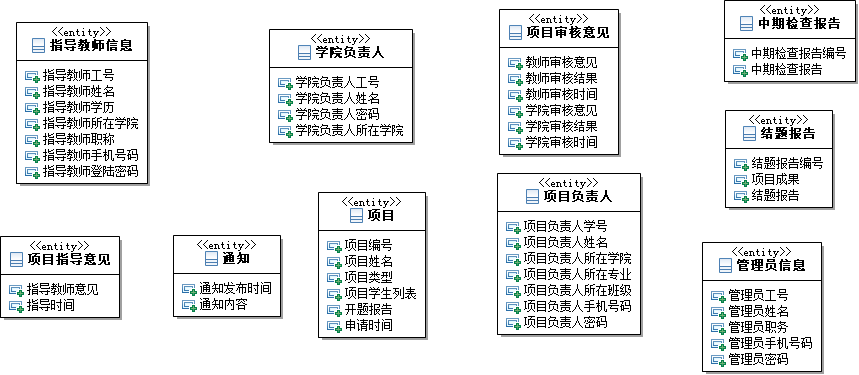
# 4.6 系统静态建模

## 4.6.1 系统类图建模

类图(Class diagram)由许多（静态）说明性的模型元素（例如类、包和它们之间的关系，这些元素和它们的内容互相连接）组成。类图可以组织在（并且属于）包中，仅显示特定包中的相关内容。

类图(Class diagram)是最常用的[UML图](http://baike.baidu.com/item/UML%E5%9B%BE)，显示出类、接口以及它们之间的静态结构和关系；它用于描述系统的结构化设计。

如图：



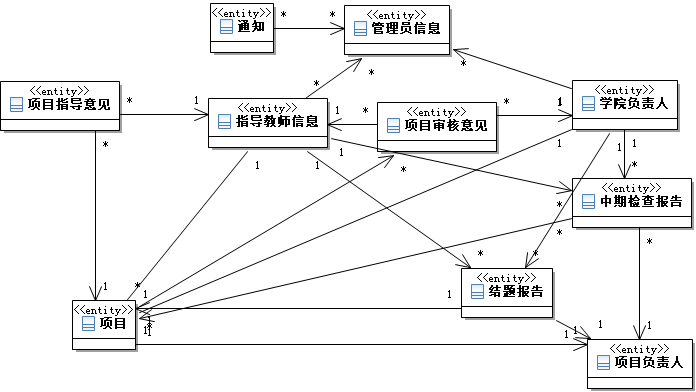
## 4.6.2 系统类关系图建模

客观世界中，对象不是独立存在的对象间存在各种关系，类间存在关系。因此，需要对类图分析结果进行进一步完善类的特征。

类关系主要包括以下几种：

关联（单向关联和双向关联）、聚合和组合、泛化、依赖（使用依赖和实例依赖）

如图：

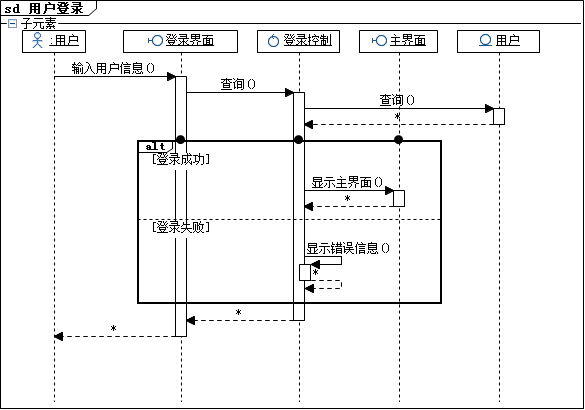


# 4.7 系统顺序图建模

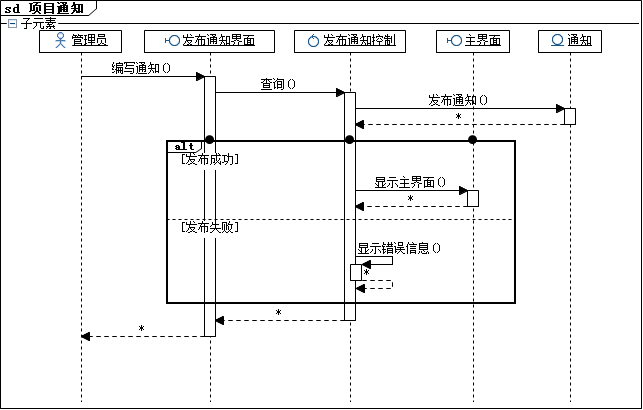
顺序图是将交互关系表示为一个二维图。纵向是[时间轴](http://baike.baidu.com/item/%E6%97%B6%E9%97%B4%E8%BD%B4)，时间沿竖线向下延伸。横向轴代表了在协作中各独立对象的类元角色。类元角色用生命线表示。当对象存在时，角色用一条虚线表示，当对象的过程处于激活状态时，生命线是一个双道线。

如图:

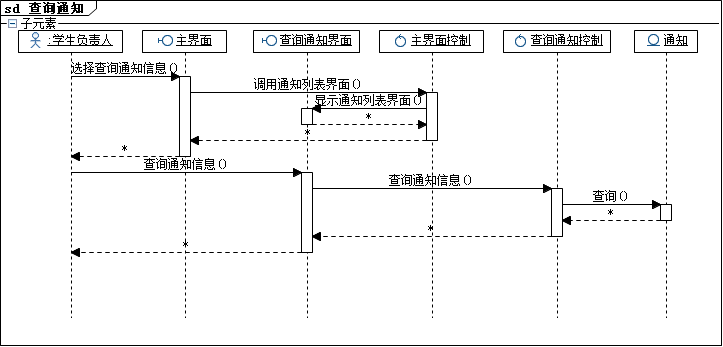
用户登录：



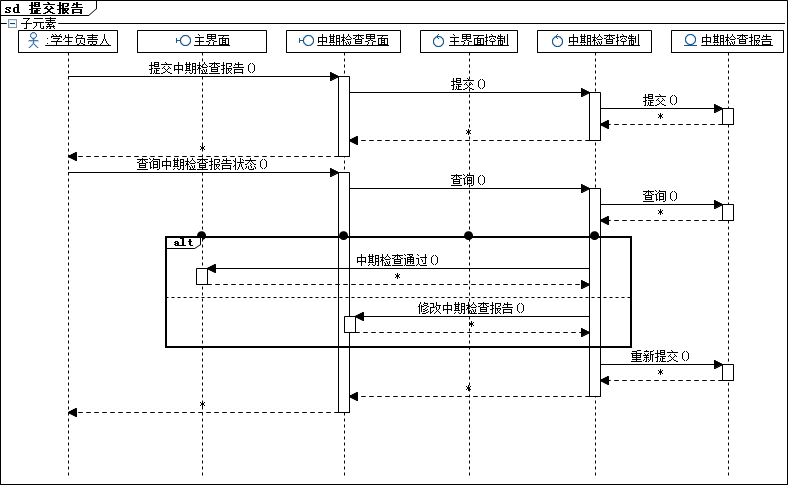
项目通知：



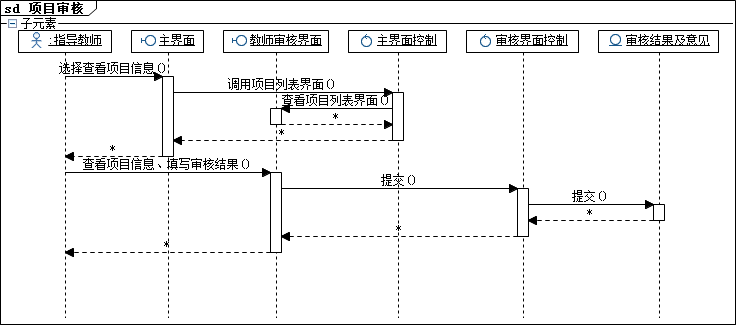
查询通知：



提交报告：



项目审核：

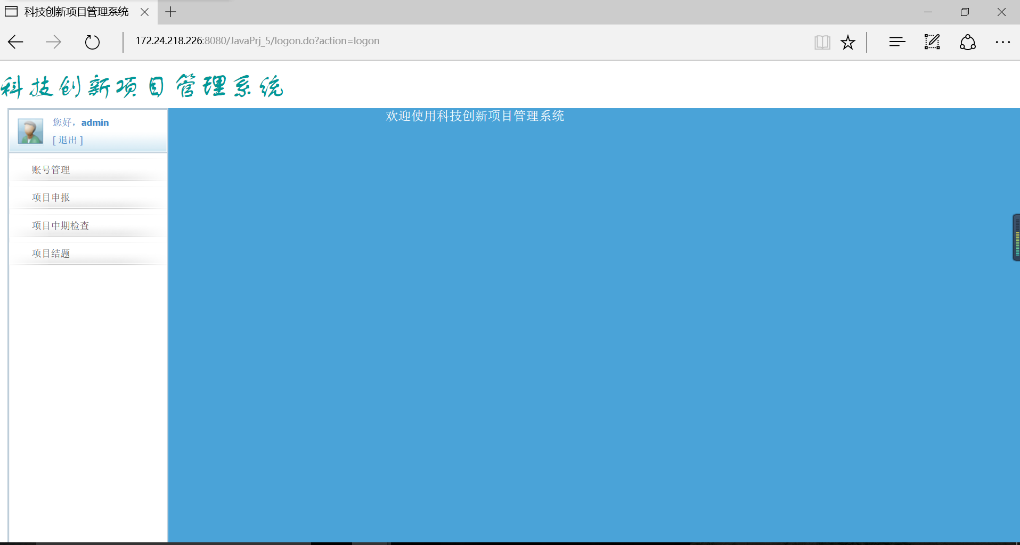


# 5.1系统界面设计与实现

## 5.1.1登陆页面



## 5.1.2主页面



## 5.1.3帐号管理页面



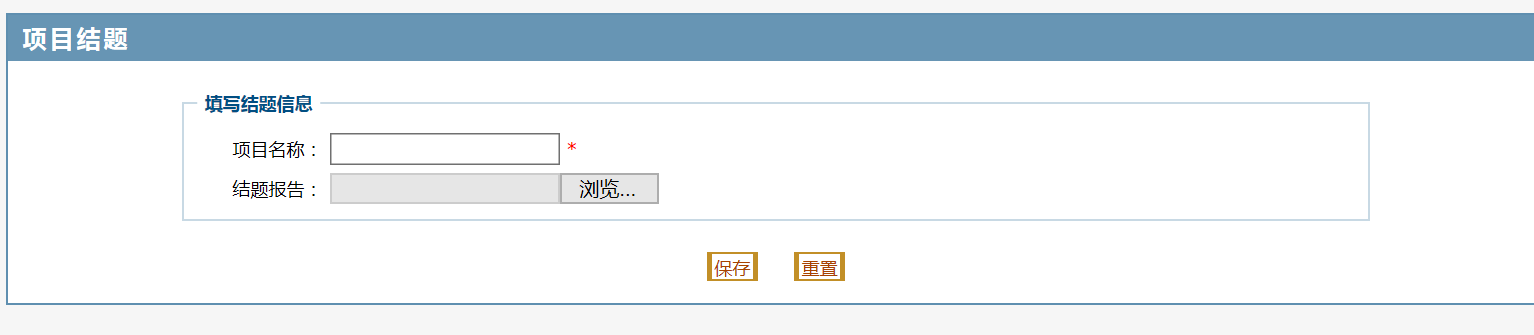
## 5.1.4项目申报页面



## 5.1.5中期检查页面



## 5.1.6项目结题页面



## 5.1.7审核页面

