**《Web项目实践》课程设计**

**（2018/2019学年第一学期第9周）**

**指导教师：肖荣**

**组员：**

**2014329620074 贺 娟 计算机科学与技术1班 15868154584**

**2015329620027 吴孝威 计算机科学与技术1班 15868154584**

**2015329620042 李嘉灵 计算机科学与技术2班 15868154584**

**2015329620044 刘 丹 计算机科学与技术2班 15868154584**

**2015329620078 金弋然 计算机科学与技术3班 15857152467**

目录

[一. 项目概况 5](#_Toc533755649)

[1.1. 所属项目组及任务分配记录 5](#_Toc533755650)

[1.2. 开发工具语言和框架 5](#_Toc533755651)

[1.3. 任务完成情况 6](#_Toc533755652)

[二. 需求分析 7](#_Toc533755653)

[2.1. 功能详细描述 7](#_Toc533755654)

[2.1.1. 用户管理 7](#_Toc533755655)

[2.1.2. 系统设置 7](#_Toc533755656)

[2.1.3. 站内消息 7](#_Toc533755657)

[2.1.4. 申报管理 7](#_Toc533755658)

[2.1.5. 中期检查管理 8](#_Toc533755659)

[2.1.6. 结题验收管理 8](#_Toc533755660)

[2.2. 用例图 9](#_Toc533755661)

[2.2.1. 超级管理员用例图 9](#_Toc533755662)

[2.2.2. 平台管理员用例图 9](#_Toc533755663)

[2.2.3. 教职工用例图 10](#_Toc533755664)

[2.2.4. 专家用例图 10](#_Toc533755665)

[2.2.5. 项目负责人用例图 11](#_Toc533755666)

[2.3. 类图和顺序图 12](#_Toc533755667)

[2.3.1. 类图 12](#_Toc533755668)

[2.3.2. 顺序图 15](#_Toc533755669)

[三. 数据库设计 16](#_Toc533755670)

[3.1. 实体关系图 16](#_Toc533755671)

[3.2. 数据库建模 18](#_Toc533755672)

[3.2.1. 实体清单 18](#_Toc533755673)

[3.2.2. 基本表格详细设计 18](#_Toc533755674)

[3.3. 数据库表结构 25](#_Toc533755675)

[四. 关键功能说明 25](#_Toc533755676)

[4.1. 关键脚本代码及说明 25](#_Toc533755677)

[4.2. 关键流程代码及说明 32](#_Toc533755678)

[五. 部署与演示 37](#_Toc533755679)

[5.1. 部署方式 37](#_Toc533755680)

[5.2. 操作流程演示 38](#_Toc533755681)

[5.2.1. 系统登录模块 38](#_Toc533755682)

[5.2.2. 系统设置模块 39](#_Toc533755683)

[5.2.3. 用户管理模块 40](#_Toc533755684)

[5.2.4. 申报管理模块 40](#_Toc533755685)

[5.2.5. 项目审核模块 41](#_Toc533755686)

[5.2.6. 站内消息模块： 42](#_Toc533755687)

[六. 心得与体会 42](#_Toc533755688)

# 项目概况

## 所属项目组及任务分配记录

|  |  |  |  |
| --- | --- | --- | --- |
| 任务分配 | | | |
| 负责人 | 主要负责模块 | 附加任务 | 组员角色 |
| 刘丹  2015329620044 | 系统设置模块  (角色管理、角色变更) | 1.完成对应模块的原型设计  2.完成对应模块相关设计  3.完成对应模块的报告 | 组员 |
| 贺娟  2014329620074 | 用户登录  用户管理（用户信息管理） | 1.完成对应模块的原型设计  2.完成对应模块相关设计  3.完成对应模块的报告 | 组员 |
| 吴孝威  2015329620027 | 申报管理模块  项目审核模块（申报类型管理） | 1.完成对应模块的原型设计  2.完成对应模块相关设计  3.完成对应模块的报告 | 组员 |
| 金弋然  2015329620078 | 项目审核模块（中期验收管理、结题验收管理）  系统设置模块（字典管理） | 1.完成对应模块的原型设计  2.完成对应模块相关设计  3.负责项目的日常进度监督和反馈  4.负责项目文档的最终整合 | 组长1 |
| 李嘉灵  2015329620042 | 项目审核模块（立项检查管理）  权限限制（待完善）  站内消息管理（待开发） | 1.项目原型设计规范、整合  2.数据库设计、建模、编写文档  3.项目框架搭建  4.负责项目的最终整合调试 | 组长2 |

## 开发工具语言和框架

高校项目管理系统的主要功能是结合实际的业务需求开展的，根据具体的实例需求基本的项目管理功能。

* 系统设计

借助Axure RP 8完成项目原型的设计

借助Power Designer完成数据库的逻辑设计

借助亿图工具完成项目E-R图、UML图等相关的业务流程设计

* 系统实现

主要是采用MVC设计模式，选取Spring+SpringMVC+MyBatis开源框架，借助MYSQL数据库完成整体系统架构的基本构建。

## 任务完成情况

|  |  |  |  |
| --- | --- | --- | --- |
| 模块功能完成情况分析 | | | |
| 主要模块 | 子模块 | 功能简介 | 完成情况 |
| 用户登录注册 | / | 实现用户的登录注册 | 完成 |
| 系统设置模块 | 角色管理 | 实现用户角色的增删改查 | 完成 |
| 角色变更 | 实现用户角色的一对一变更 | 完成 |
| 数据字典管理 | 实现基本数据的维护（CRUD） | 完成 |
| 权限管理 | 根据用户角色限制用户权限 | 待完善 |
| 用户管理模块 | 用户信息管理 | 实现用户信息的增删改查 | 完成 |
| 申报管理模块 | 申报入口 | 提供申报规则说明 | 完成 |
| 申报材料 | 用户提交申报材料 | 完成 |
| 申报进度 | 查看个人申报进度 | 完成 |
| 项目审核模块 | 申报类型管理 | 实现项目申报类型的增删改查 | 完成 |
| 立项检查管理 | 项目立项检查（通过、驳回） | 完成 |
| 中期验收管理 | 中期验收（通过、驳回） | 完成 |
| 结题验收管理 | 结题验收（通过、驳回） | 完成 |
| 站内消息 | 站内消息 | 实现消息的即时推送 | 待开发 |

说明：

目前实现了高校管理系统的基本业务流程，但部分模块在整合的过程还有待改善，部分细节没有很好地进行处理。站内消息模块由于时间限制没有进行开发。

# 需求分析

## 功能详细描述

### 用户管理

用户管理指能对平台的各种不同类型用户进行增、删、改、查、密码重置等操作、能对用户进行一对一或一对多的角色分配，用户信息包括：用户编码、登录密码、用户名称、所属部门、创建时间、更新时间、用户状态、用户角色、备注等；

### 系统设置

系统设置通常用来设置平台常用但不经常变化的一些数据，目前主要针对角色、菜单和权限进行设置。

角色管理指对平台用户角色进行增、删、改、查操作。字段包括：角色名称、角色描述、创建时间、更新时间、备注等；

菜单管理指对平台菜单进行增、删、改、查操作；

角色权限分配指对角色分配相应的菜单、目前不需要支持权限到操作按钮；

### 站内消息

站内消息指平台操作员针对教职工用户的项目记录进行重要状态更改时，由系统生成的发送给教职工用户的提醒消息。此类消息通常会在特定教职工登录系统后，以弹窗或醒目的消息数目作为提醒方式。例如，教务处管理人员确定某个项目被立项时，发送站内消息给相应项目负责人。

本平台的站内消息作为独立模块，提供给申报管理、中期检查管理、结题验收管理等模块使用。

### 申报管理

申报管理包括以下操作流程：

1、由平台管理人员设置项目的申报规则，提供申报入口；

2、平台用户在规定的时间期限内，填写项目内容并上传申报书电子版。项目信息包括，申报级别（校级一类、校级二类）、项目名称、负责人（默认登录用户）、手机号码、电子邮件、所属专业、项目成员（多个）、立项承诺等，填写完上述信息后，负责人还需上传申报书电子版。填写完的信息可以保存，也可以直接提交。

3、管理人员对已提交的项目进行初步审核，初审通过后，修改项目状态为初审通过；

4、对初审通过后的项目进行专家分配，分配完成后，项目记录状态变为立项评审中；

5、专家用户登录平台，审核分配给自己的项目记录，进行打分和填写评语，所有专家都评审完成后，项目状态自动变更为立项评审完成；

6、管理人员根据专家评审的结果确认那些项目适合立项，确认立项的项目状态修改为已立项、其它的设置为不立项；

### 中期检查管理

中期检查流程包括：

1、平台管理人员对已立项的项目设置中期检查时间期限和项目应该上传的材料说明；

2、项目负责人上传中期检查材料；

3、平台管理人员负责评审中期检查材料，并修改中期检查通过的项目为中期检查通过；对不通过的项目驳回修改，填写驳回原因，同时变更项目状态为中期检查待整改；项目负责人收到消息提醒后，重复第2步操作直到中期检查通过；

### 结题验收管理

结题验收流程包括：

1、平台管理人员对中期检查通过的项目设置结题验收时间期限和项目应该上传的材料说明；

2、项目负责人上传结题验收材料；

3、平台管理人员负责评审结题验收材料，并修改结题验收通过的项目为已结题；对不通过的项目驳回修改，填写驳回原因，同时变更项目状态为结题验收待整改；项目负责人收到消息提醒后，重复第2步操作直到结题验收完成。

## 用例图

### 超级管理员用例图

描述：能够管理用户、权限、角色。能够对用户、权限、角色的增删改查。

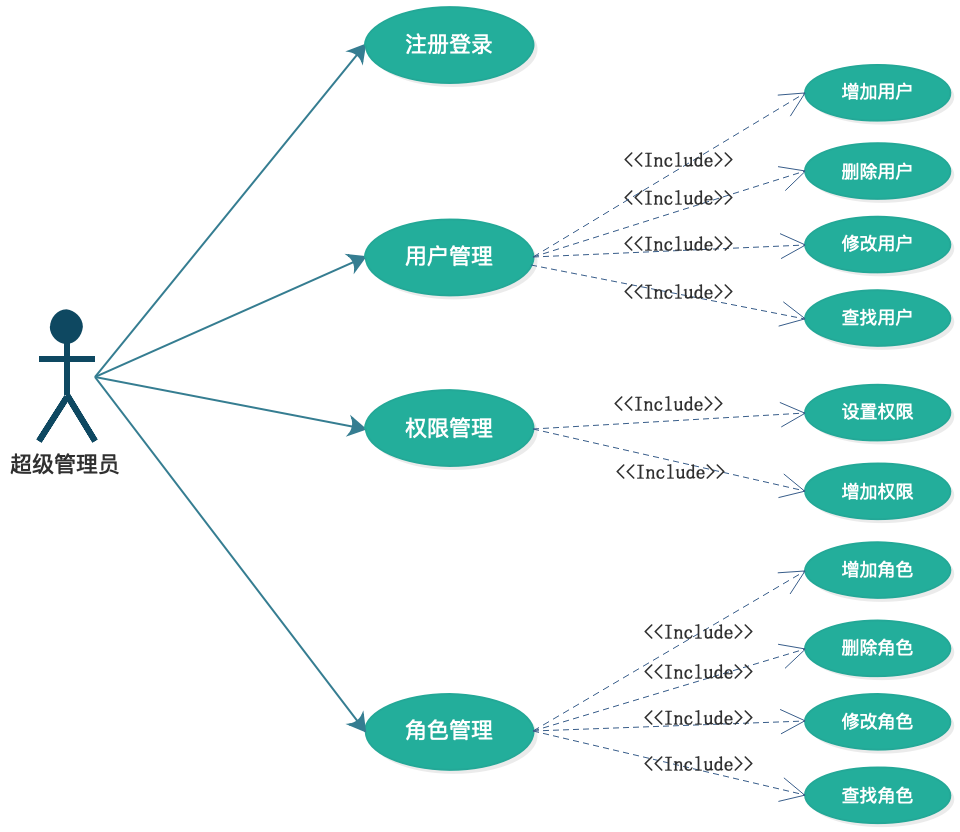


图 1超级管理员用例图

### 平台管理员用例图

描述：能够进行设置申报规则以及截止日期，审核项目，分配专家，查看材料等操作。

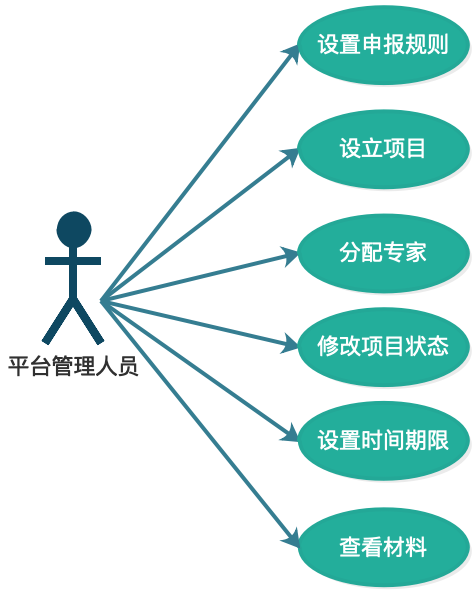


图 2平台管理员用例图

### 教职工用例图

描述：能够进行申报书填写，查看消息等操作。

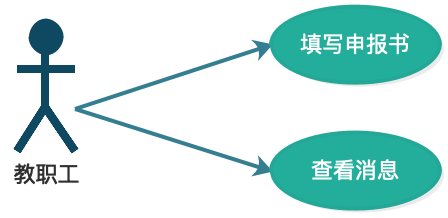


图 3教职工用例图

### 专家用例图

描述：能够进行评分，填写评语等操作。

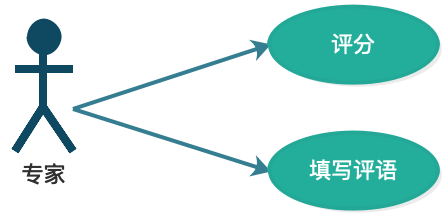


图 4专家用例图

### 项目负责人用例图

描述：能够进行上传材料、设置基本信息等操作。

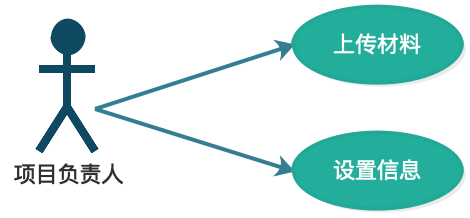


图 5项目负责人用例图

## 类图和顺序图

### 类图

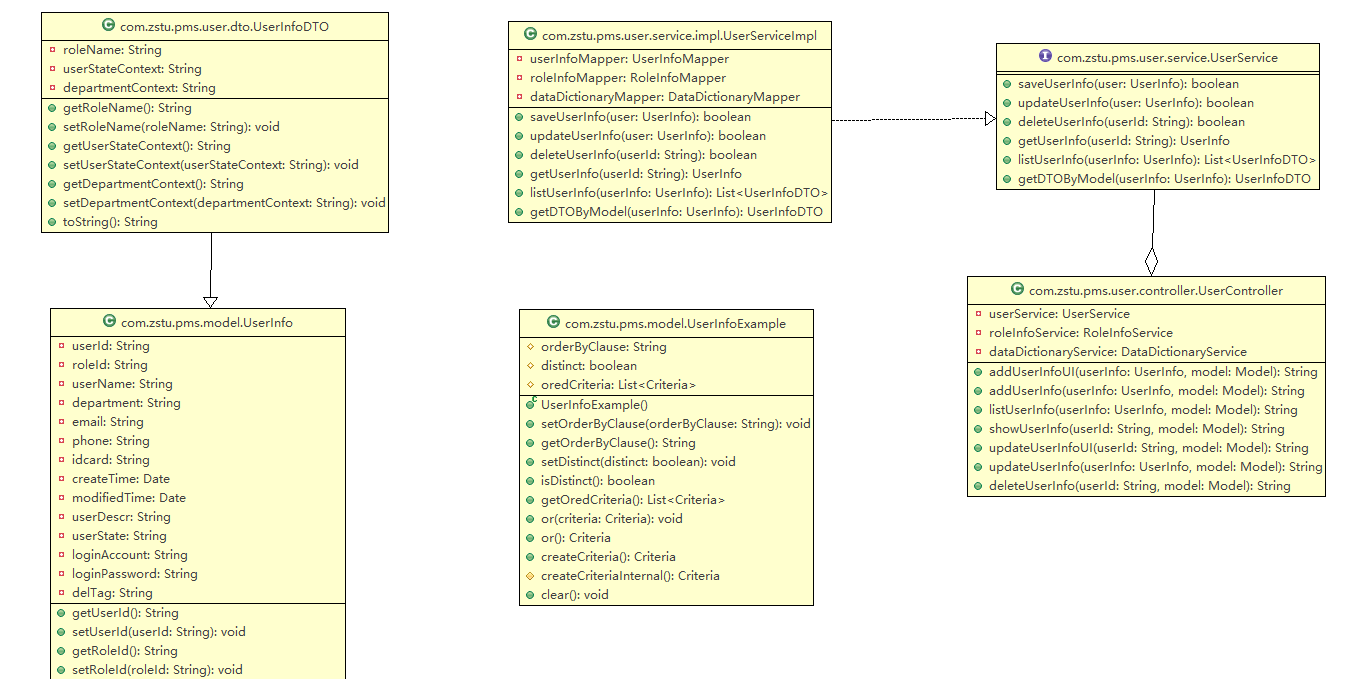


图 6用户管理类图

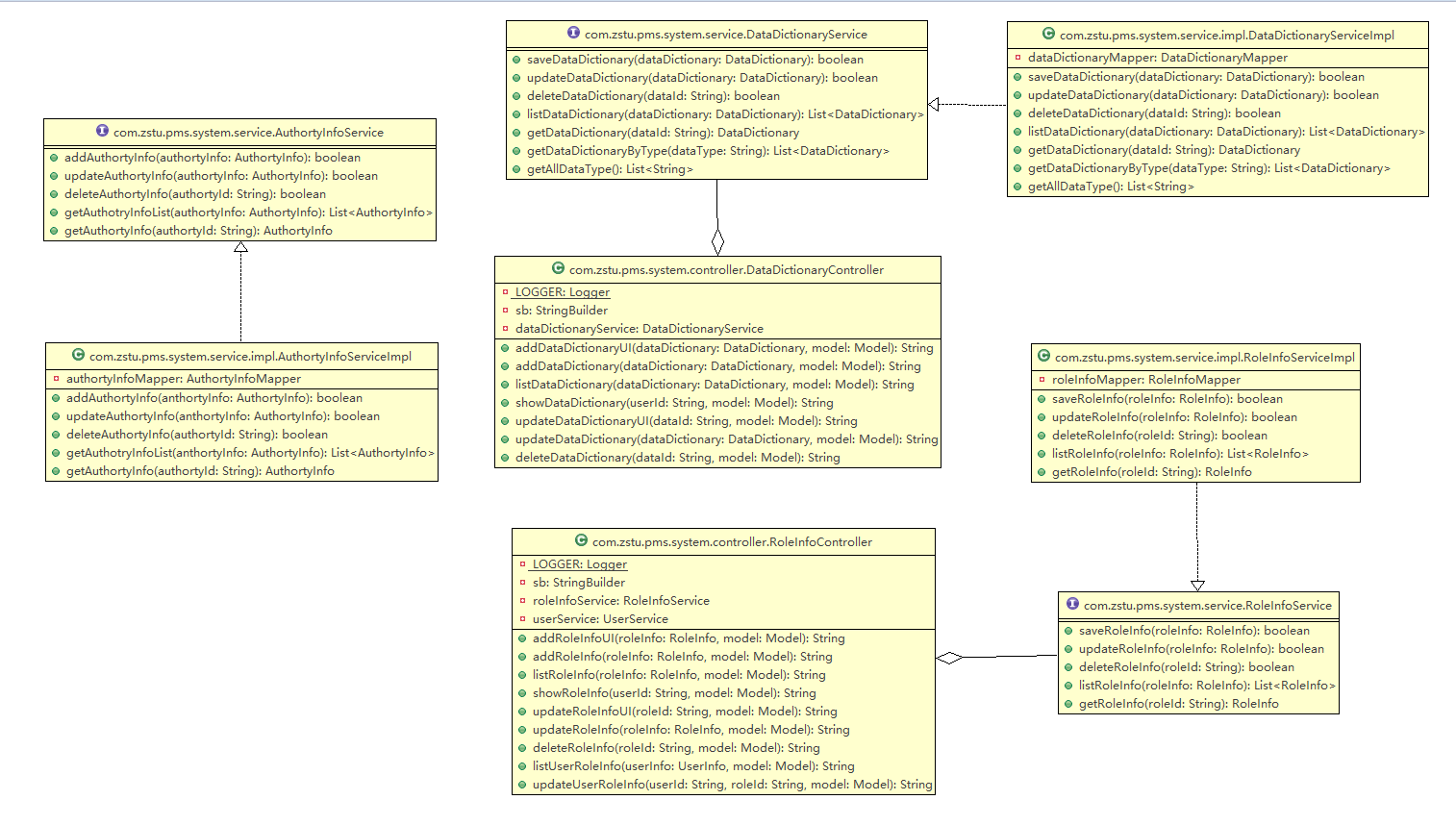


图 7系统管理类图

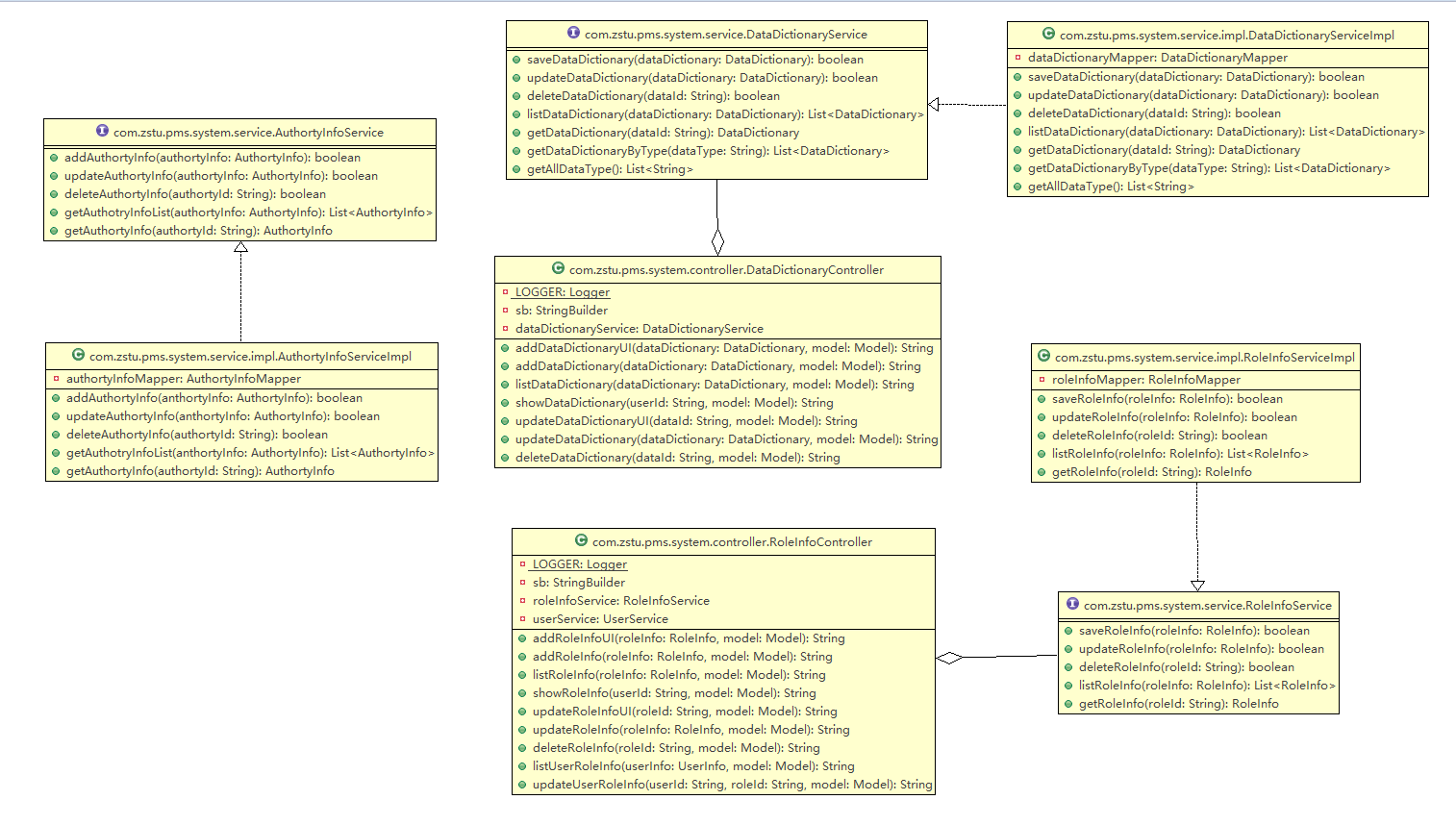


图 8申报管理类图

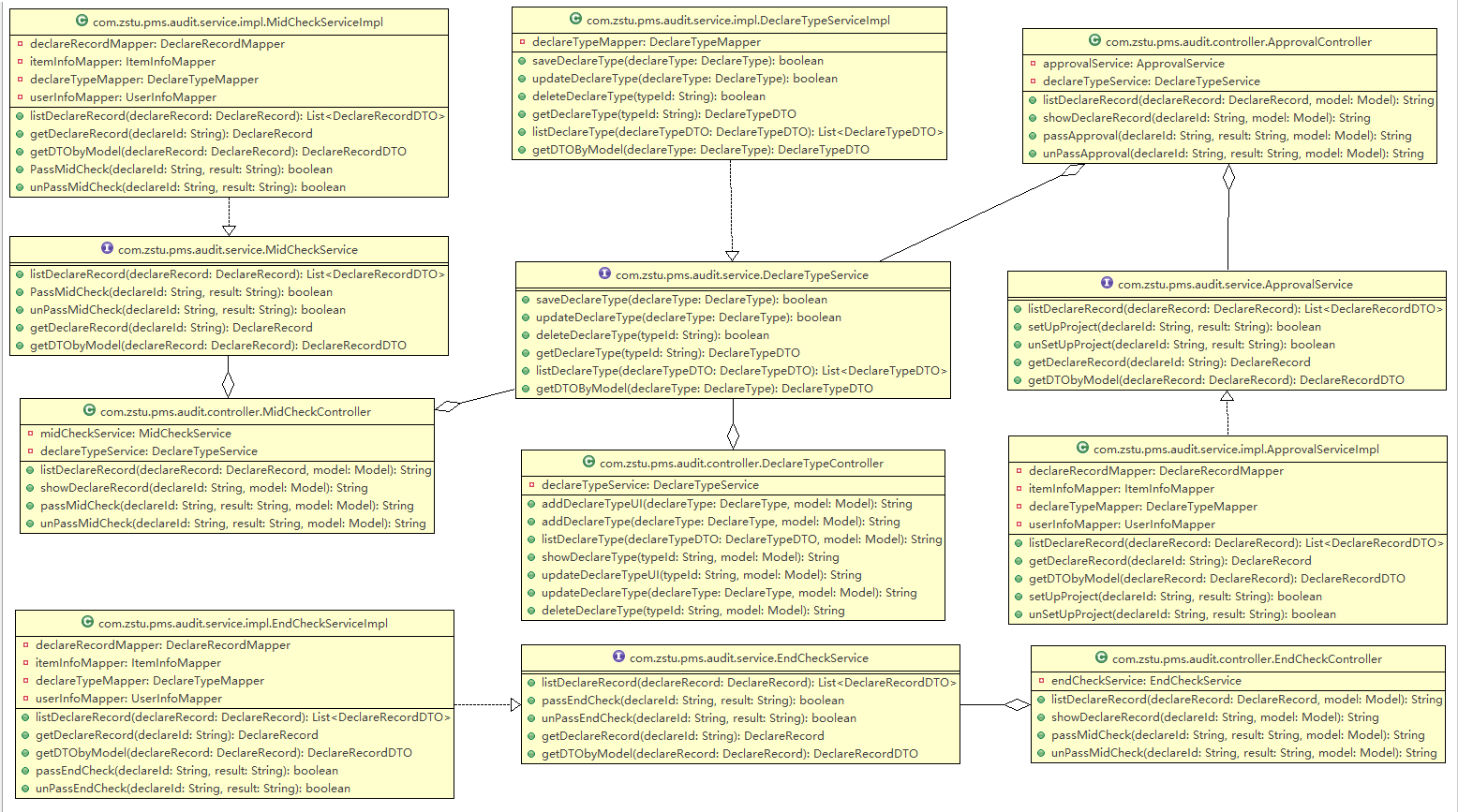


图 9项目审核类图

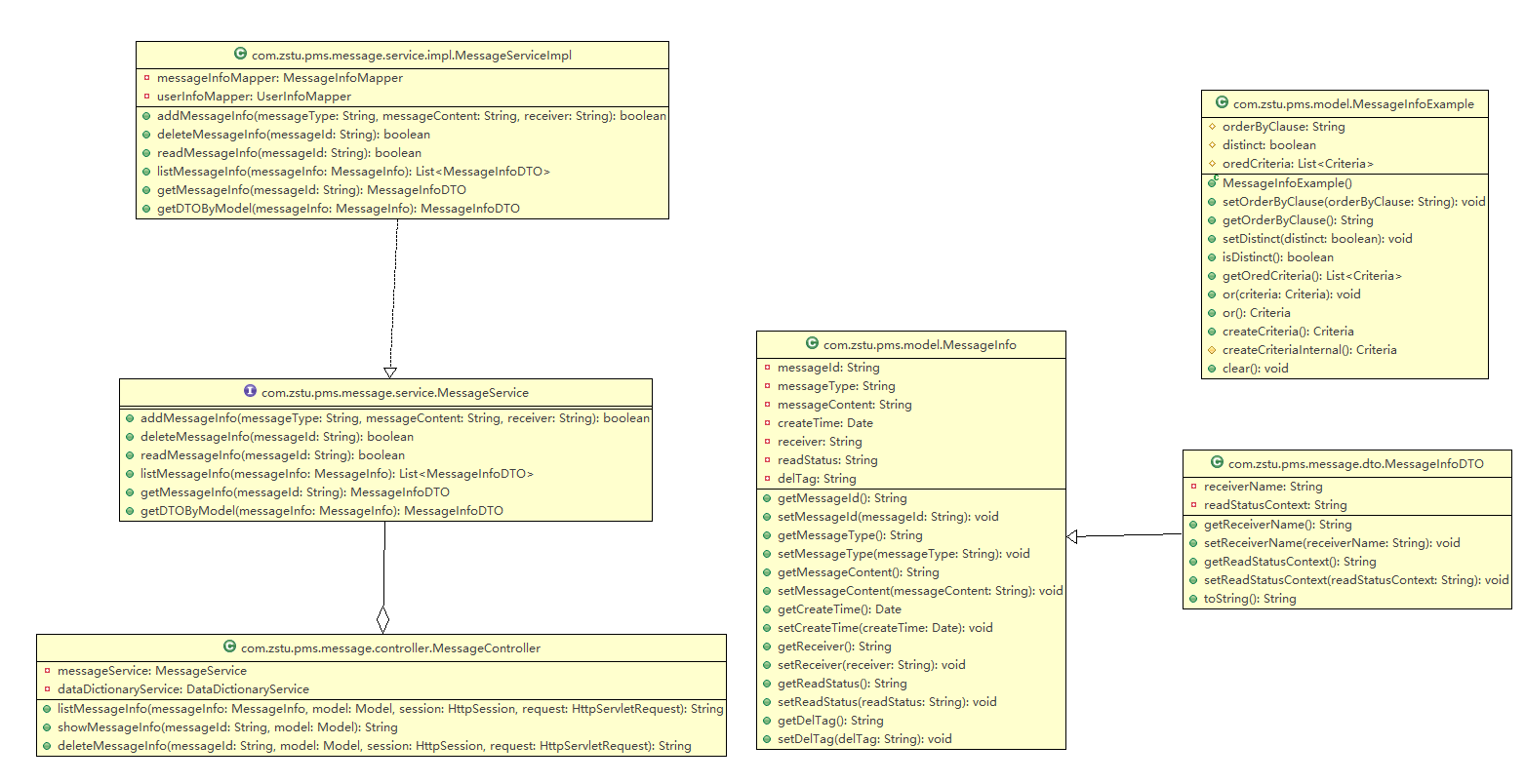


图 10消息管理类图

### 顺序图

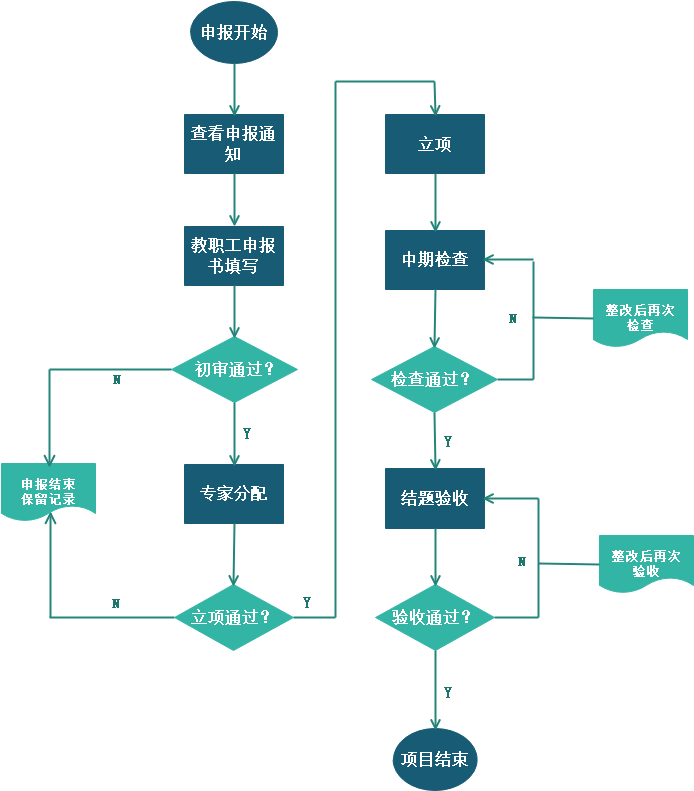


图 11项目审核流程图

# 数据库设计

## 实体关系图

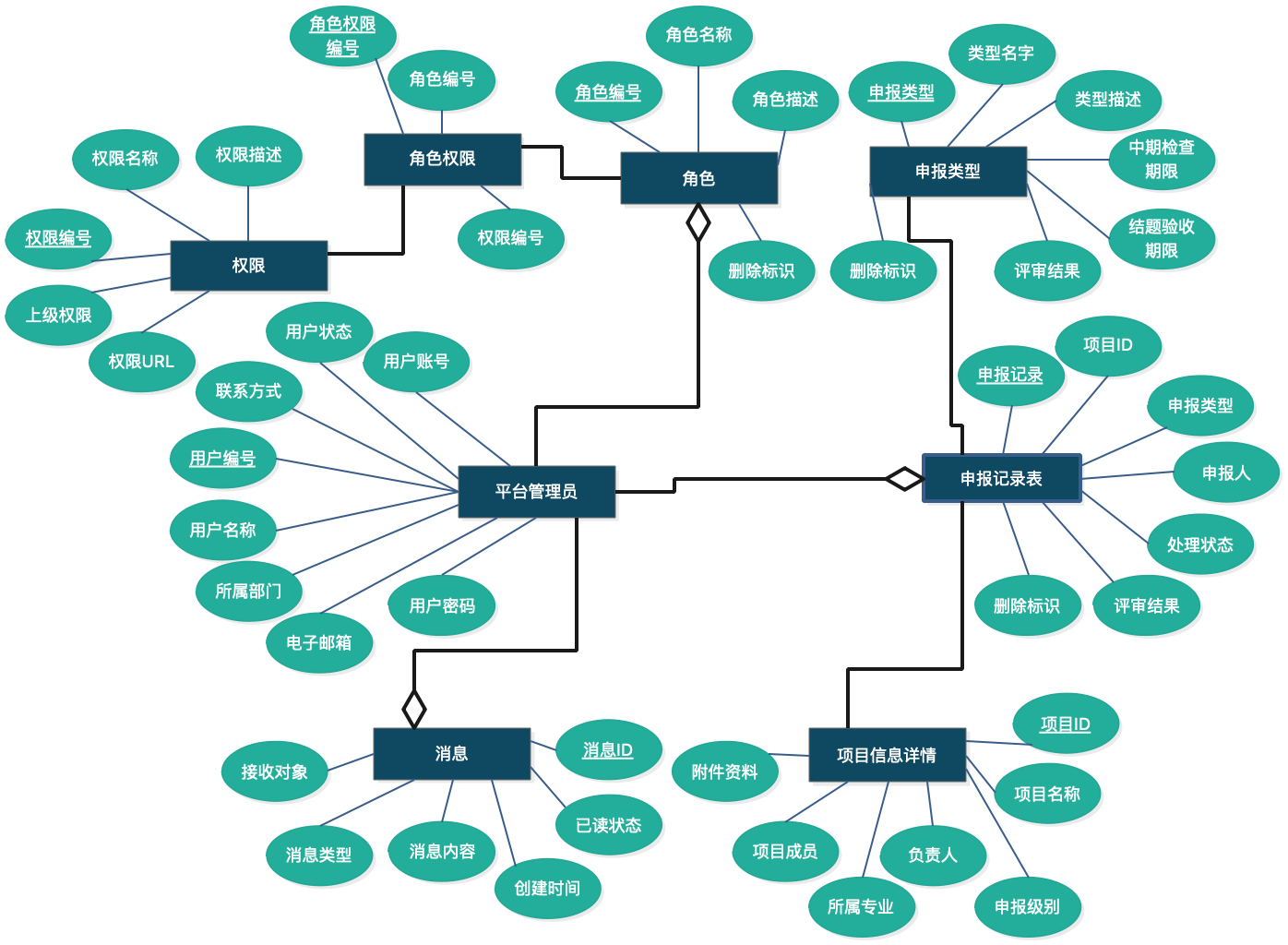


图 12平台管理员E-R图

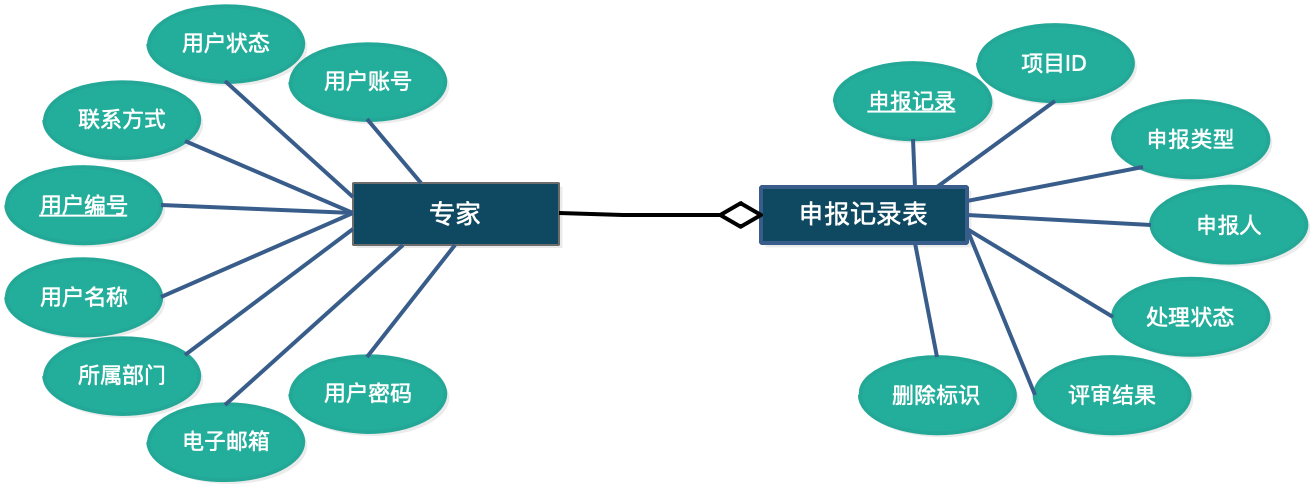


图 13专家E-R图

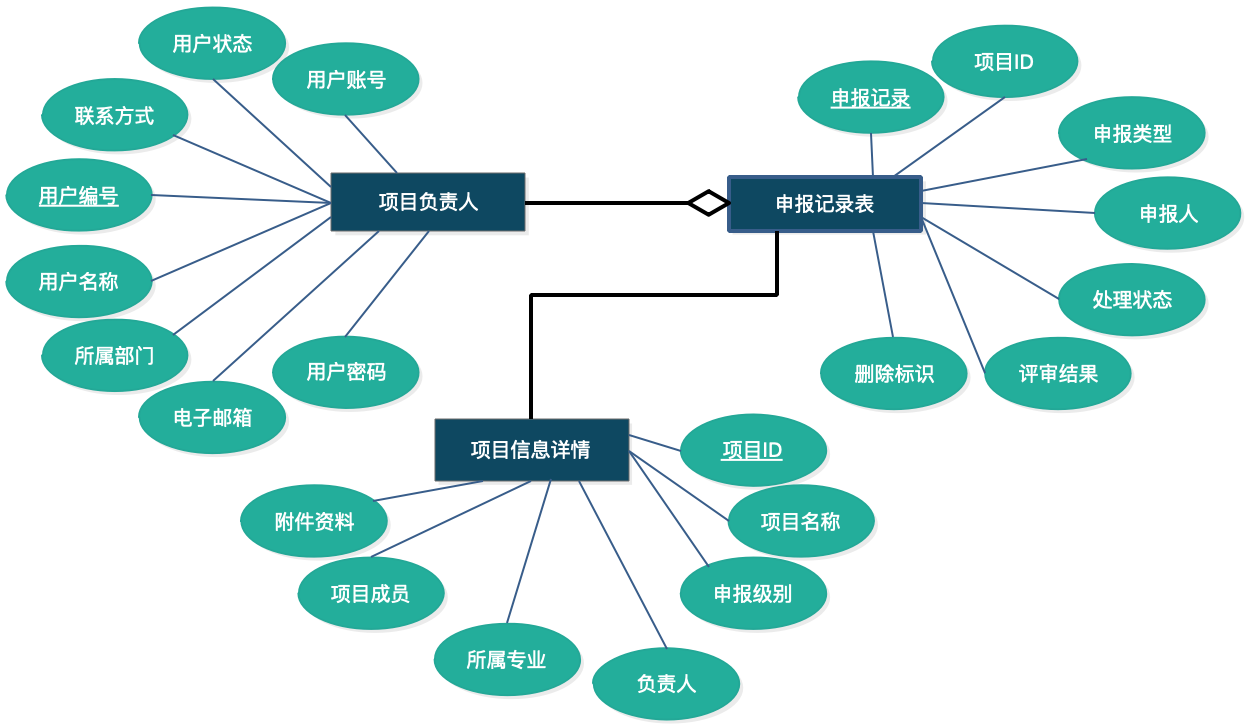


图 14项目负责人E-R图

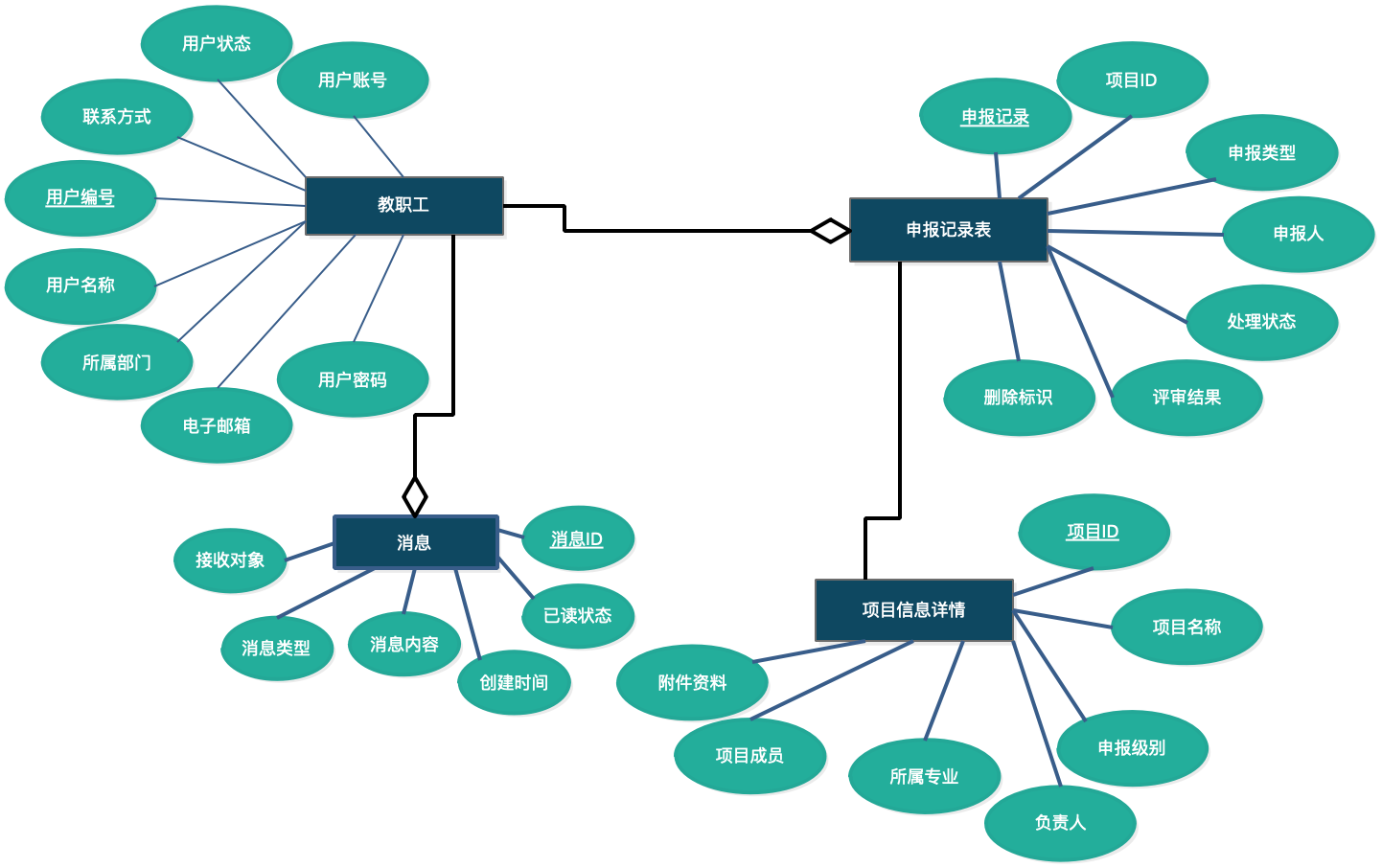


图 15教职工E-R图

## 数据库建模

### 实体清单

表 1实体清单

|  |  |
| --- | --- |
| Name | Code |
| 用户信息表 | user\_info |
| 角色信息表 | role\_info |
| 权限信息表 | authorty\_info |
| 角色权限信息表 | role\_authorty\_info |
| 数据字典信息表 | data\_dictionary |
| 消息表 | message |
| 申报类型表 | declare\_type |
| 项目信息表 | item\_info |
| 申报记录表 | declare\_record |

### 基本表格详细设计

表 2用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 用户信息表：user\_info | | | |
| **用户信息** | **定义变量** | **数据类型** | **约束** |
| 用户id | user\_id | varchar2(40) | primary key |
| 用户名称 | user\_name | varchar2(100) | not null |
| 所属部门 | department | varchar2(40) | not null |
| 联系方式 | phone | varchar2(40) | / |
| 电子邮箱 | email | varchar2(40) | / |
| 创建日期 | create\_time | datetime | / |
| 修改日期 | modified\_time | datetime | / |
| 用户状态 | user\_state | int(1) | check、default:0 |
| 备注信息 | user\_descr | varchar2(2000) | / |
| 登陆账号 | login\_account | varchar2(100) | unique |
| 登陆密码 | login\_password | varchar2(100) | defalut:000000 |
| 删除标识 | del\_tag | int(1) | check、default:0 |
| 用户角色id | role\_id | varchar2(40) | foreign key |
| 说明：用户id作为主键，每个用户拥有属于自己的基本信息，每个用户对应着不同的账号信息和角色信息（每个角色对应不同的权限） | | | |

表 3角色信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 角色信息表：role\_info | | | |
| 角色信息 | 定义变量 | 数据类型 | 约束 |
| 角色id（随机10char） | role\_id | varchar2(40) | primary key |
| 角色名称 | role\_name | varchar2(100) | not null |
| 角色描述 | role\_descr | varchar2(1000) | / |
| 创建时间 | create\_time | datetime | / |
| 修改时间 | modified\_time | datetime | / |
| 删除标识 | del\_flag | int(1) | check、default:0 |
| 说明：角色id作为主键，标识不同的用户拥有不同的角色信息（一个用户只能对应一个角色信息） | | | |

表 4权限信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 权限信息表：authorty\_info | | | |
| 权限信息 | 定义变量 | 数据类型 | 约束 |
| 权限id | authorty\_id | varchar2(40) | primary key |
| 权限名称 | authorty\_name | varchar2(100) | not null |
| 权限描述 | authorty\_descr | varchar2(1000) | / |
| 上级权限id | parent\_id | varchar2(40) | foreign key |
| 权限url | authorty\_url | varchar2(1000) | / |
| 说明：权限id作为主键，标识不同的角色拥有不同的权限信息（一个角色拥有不同的权限） | | | |

表 5角色权限信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 角色权限信息表：role\_authorty\_info | | | |
| 权限信息 | 定义变量 | 数据类型 | 约束 |
| 角色权限id | role\_authorty\_id | varchar2(40) | primary key |
| 角色id | role\_id | varchar2(40) | foreign key |
| 权限id | authorty\_id | varchar2(40) | foreign key |
| 说明：角色权限id作为主键，用于关联角色和权限 | | | |

表 6数据字典信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 数据字典信息表：data\_dictionary | | | |
| 数据字典信息 | 定义变量 | 数据类型 | 约束 |
| 数据编号 | data\_id | varchar2(40) | primary key |
| 内容 | data\_content | varchar2(100) | not null |
| 类型 | data\_type | varchar2(100) | / |
| 描述 | data\_desc | varchar2(2000) | / |
| 说明：此处设计数据字典表用以存放一些常用字段，用以相关数据的填充和维护 | | | |

表 7消息管理表

|  |  |  |  |
| --- | --- | --- | --- |
| 消息管理：message | | | |
| 消息管理 | 定义变量 | 数据类型 | 约束 |
| 消息id | message\_id | varchar2(40) | primary key |
| 消息类型 | message\_type | varchar2(100) | / |
| 消息内容 | message\_content | varchar2(2000) | / |
| 创建时间 | create\_time | datetime | / |
| 接收对象 | receiver | varchar2(40) | foreign key |
| 已读状态 | read\_status | int | / |
| 删除标识 | del\_tag | int | default:0 |
| 说明：提供消息管理模块信息，提供给申报管理、中期检查管理、结题验收管理等模块使用 | | | |

表 8申报类型表

|  |  |  |  |
| --- | --- | --- | --- |
| 申报类型：declare\_type | | | |
| 申报类型 | 定义变量 | 数据类型 | 约束 |
| 申报类型id | type\_id | varchar2(40) | primary key |
| 类型名称 | type\_name | varchar2(100) | not null |
| 类型描述（申报内容描述） | type\_descr | varchar2(2000) | / |
| 创建时间 | create\_time | datetime | / |
| 修改时间 | modified\_time | datetime | / |
| 中期检查期限 | middle\_time | datetime | / |
| 结题验收期限 | end\_time | datetime | / |
| 项目开启状态 | start\_status | int | / |
| 删除标识 | del\_tag | int | default:0 |
| 说明：由相关工作人员完成申报类型的管理：申报类型(包括每个类型对应的详细内容) | | | |

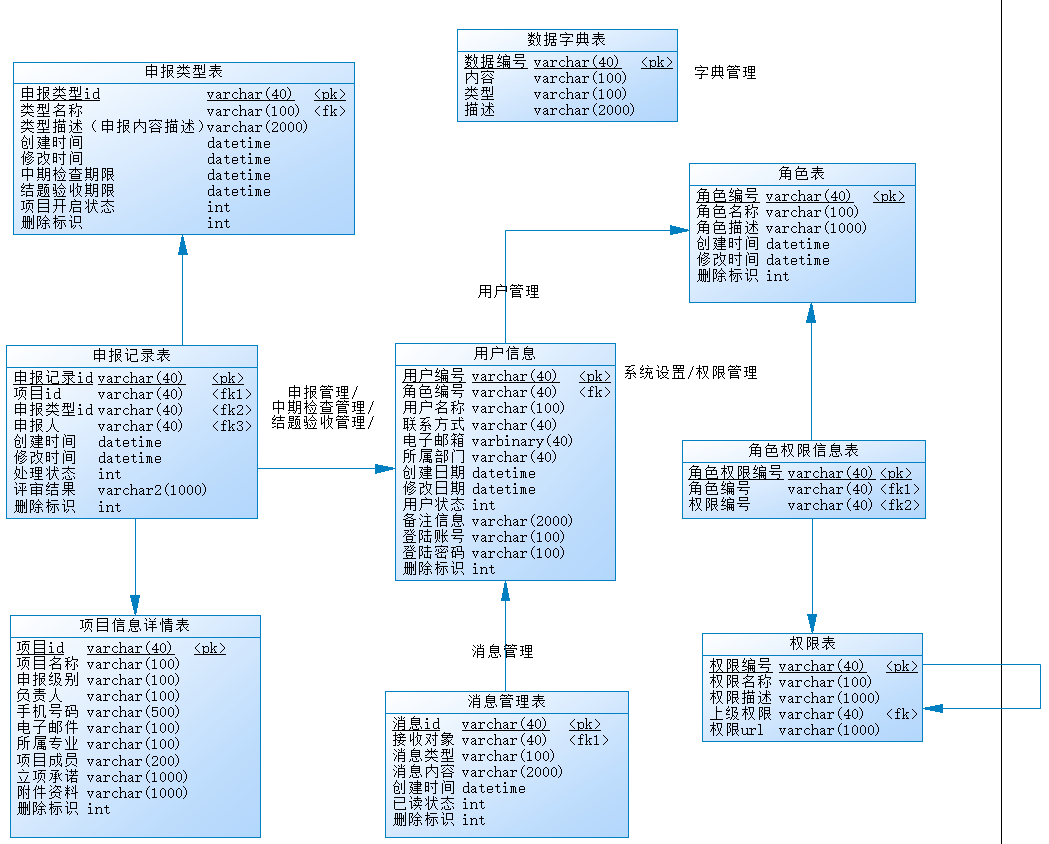
表 9项目信息详情表

|  |  |  |  |
| --- | --- | --- | --- |
| 项目信息详情：item\_info | | | |
| 项目信息详情 | 定义变量 | 数据类型 | 约束 |
| 项目id | item\_id | varchar2(40) | primary key |
| 项目名称 | item\_name | varchar2(100) | not null |
| 申报级别 | item\_class | varchar2(100) | / |
| 负责人 | principal | varchar2(100) | （默认为登陆用户） |
| 手机号码 | phone | varchar2(500) | / |
| 电子邮件 | email | varchar2(100) | / |
| 所属专业 | profession | varchar2(100) | / |
| 项目成员 | members | varchar2(200) | / |
| 立项承诺 | commitment | varchar2(1000) | / |
| 附件资料 | annex | varchar2(1000) | / |
| 删除标识 | del\_tag | int | default:0 |
| 说明：项目消息详情：包括申报项目的所有相关的信息 | | | |

表 10申报记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 申报记录表：declare\_record | | | |
| 申报记录 | 定义变量 | 数据类型 | 约束 |
| 申报记录id | declare\_id | varchar2(40) | primary key |
| 记录类型 | type\_id | varchar2(40) | foreign key |
| 申报项目 | item\_id | varchar2(40) | foreign key |
| 申报人 | user\_id | varchar2(40) | foreign key |
| 创建时间 | create\_time | datetime | / |
| 修改时间 | modified\_time | datetime | / |
| 处理状态 | process\_status | int | foreign key |
| 评审结果 | result | varchar2(1000) | / |
| 删除标识 | del\_tag | int | default:0 |
| 说明：申报记录id作为主键，此外，申报记录包括申报类型、对应申报的项目、相关的时间、记录的处理状态等字段，用以进行项目管理  处理状态通过数字标识：  **每个数字标识此申报项目处于不同的一个处理状态，每个管理模块通过对不同状态的申报项目进行不同的管理操作**  0标识项目申报中（已申报）  1标识项目通过初审，进入立项状态（已立项）  2标识项目通过中期检查，进入结题验收状态（中期检查通过）  3标识项目通过结题验收，进入审核结束状态（结题验收通过，即最终审核通过）  -1标识项目初审未通过（未立项，待整改）  -2标识项目中期检查未通过（中期检查未通过，待整改）  -3标识项目结题验收未通过（结题验收未通过，待整改）  评审结果：  此处评审过程中审核的结果通过消息管理机制进行控制，不作过审核过程的评审记录保存  只对完成整个流程的申报项目进行最终评审结果的判定，由评审专家进行记录的填充  删除标识：  0标识记录未删除  1标识记录待删除（考虑回收站机制，避免用户误删操作）  项目申报 -- 项目立项 -- 中期检查 -- 结题验收  归结为对项目记录的状态管理，统一到一张表进行处理，针对提交意见和理由以消息形式对用户进行通知 | | | |

## 数据库表结构



# 关键功能说明

## 关键脚本代码及说明

数据库脚本代码及说明：

/\*

Navicat MySQL Data Transfer

Source Server : hahabibu

Source Server Version : 50557

Source Host : localhost:3306

Source Database : pms

Target Server Type : MYSQL

Target Server Version : 50557

File Encoding : 65001

Date: 2018-12-23 21:24:26

\*/

SET FOREIGN\_KEY\_CHECKS=0;

**1.authorty\_info**

-- ----------------------------

-- Table structure for `authorty\_info`

-- ----------------------------

DROP TABLE IF EXISTS `authorty\_info`;

CREATE TABLE `authorty\_info` (

`authorty\_id` varchar(40) NOT NULL COMMENT '权限id',

`authorty\_name` varchar(100) DEFAULT NULL COMMENT '权限名称',

`authorty\_desc` varchar(1000) DEFAULT NULL COMMENT '权限描述',

`parent\_id` varchar(40) DEFAULT NULL COMMENT '上级权限id',

`authorty\_url` varchar(1000) DEFAULT NULL COMMENT '权限url',

`authorty\_class` varchar(40) DEFAULT NULL COMMENT '权限等级',

PRIMARY KEY (`authorty\_id`),

KEY `FK\_Reference\_2` (`parent\_id`),

CONSTRAINT `authorty\_info\_ibfk\_1` FOREIGN KEY (`parent\_id`) REFERENCES `authorty\_info` (`authorty\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**2.data\_dictionary**

-- ----------------------------

-- Table structure for `data\_dictionary`

-- ----------------------------

DROP TABLE IF EXISTS `data\_dictionary`;

CREATE TABLE `data\_dictionary` (

`data\_id` varchar(40) NOT NULL COMMENT '数据id',

`data\_content` varchar(100) DEFAULT NULL COMMENT '数据内容',

`data\_type` varchar(100) DEFAULT NULL COMMENT '数据类型',

`data\_descr` varchar(2000) DEFAULT NULL COMMENT '数据描述',

PRIMARY KEY (`data\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**3.declare\_record**

-- ----------------------------

-- Table structure for `declare\_record`

-- ----------------------------

DROP TABLE IF EXISTS `declare\_record`;

CREATE TABLE `declare\_record` (

`declare\_id` varchar(40) NOT NULL COMMENT '申报id',

`item\_id` varchar(40) DEFAULT NULL COMMENT '申报项目id',

`type\_id` varchar(40) DEFAULT NULL COMMENT '申报类型',

`user\_id` varchar(40) DEFAULT NULL COMMENT '申报用户id',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`modified\_time` datetime DEFAULT NULL COMMENT '修改时间',

`process\_status` varchar(40) DEFAULT NULL COMMENT '申报处理状态',

`result` varchar(1000) DEFAULT NULL COMMENT '申报结果',

`del\_tag` varchar(40) DEFAULT NULL COMMENT '删除标识（0.保存 1.删除）',

PRIMARY KEY (`declare\_id`),

KEY `FK\_Reference\_7` (`type\_id`),

KEY `FK\_Reference\_8` (`item\_id`),

KEY `FK\_Reference\_9` (`user\_id`),

CONSTRAINT `declare\_record\_ibfk\_1` FOREIGN KEY (`user\_id`) REFERENCES `user\_info` (`user\_id`),

CONSTRAINT `declare\_record\_ibfk\_2` FOREIGN KEY (`type\_id`) REFERENCES `declare\_type` (`type\_id`),

CONSTRAINT `declare\_record\_ibfk\_3` FOREIGN KEY (`item\_id`) REFERENCES `item\_info` (`item\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**4.declare\_type**

-- ----------------------------

-- Table structure for `declare\_type`

-- ----------------------------

DROP TABLE IF EXISTS `declare\_type`;

CREATE TABLE `declare\_type` (

`type\_id` varchar(40) NOT NULL COMMENT '申报类型id',

`type\_name` varchar(100) DEFAULT NULL COMMENT '类型名称',

`type\_descr` varchar(2000) DEFAULT NULL COMMENT '类型描述',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`modified\_time` datetime DEFAULT NULL COMMENT '修改时间',

`middle\_time` datetime DEFAULT NULL COMMENT '中期检查时间',

`end\_time` datetime DEFAULT NULL COMMENT '结题验收时间',

`start\_status` varchar(40) DEFAULT NULL COMMENT '开始状态',

`del\_tag` varchar(40) DEFAULT NULL COMMENT '删除标识(0.保存 1.删除）',

PRIMARY KEY (`type\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**5.item\_info**

-- ----------------------------

-- Table structure for `item\_info`

-- ----------------------------

DROP TABLE IF EXISTS `item\_info`;

CREATE TABLE `item\_info` (

`item\_id` varchar(40) NOT NULL COMMENT '项目id',

`item\_name` varchar(100) DEFAULT NULL COMMENT '项目名称',

`item\_class` varchar(100) DEFAULT NULL COMMENT '项目类型',

`principal` varchar(100) DEFAULT NULL COMMENT '负责人',

`phone` varchar(500) DEFAULT NULL COMMENT '联系方式',

`email` varchar(100) DEFAULT NULL COMMENT '电子邮箱',

`profession` varchar(100) DEFAULT NULL COMMENT '所属专业',

`members` varchar(200) DEFAULT NULL COMMENT '项目成员',

`commitment` varchar(1000) DEFAULT NULL COMMENT '立项承诺',

`annex` varchar(1000) DEFAULT NULL COMMENT '附件资料',

`del\_tag` varchar(40) DEFAULT NULL COMMENT '删除标识（0.保存 1.删除）',

PRIMARY KEY (`item\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**6.message\_info**

-- ----------------------------

-- Table structure for `message\_info`

-- ----------------------------

DROP TABLE IF EXISTS `message\_info`;

CREATE TABLE `message\_info` (

`message\_id` varchar(40) NOT NULL COMMENT '消息id',

`message\_type` varchar(100) DEFAULT NULL COMMENT '消息类型',

`message\_content` varchar(2000) DEFAULT NULL COMMENT '消息内容',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`receiver` varchar(40) DEFAULT NULL COMMENT '接收人',

`read\_status` varchar(40) DEFAULT NULL COMMENT '已读标识（0.未读 1.已读）',

`del\_tag` varchar(40) DEFAULT NULL COMMENT '删除标识（0.保存 1.删除）',

PRIMARY KEY (`message\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**7.role\_authorty\_info**

-- ----------------------------

-- Table structure for `role\_authorty\_info`

-- ----------------------------

DROP TABLE IF EXISTS `role\_authorty\_info`;

CREATE TABLE `role\_authorty\_info` (

`role\_authorty\_id` varchar(40) NOT NULL COMMENT '角色权限id',

`role\_id` varchar(40) DEFAULT NULL COMMENT '角色id',

`authorty\_id` varchar(40) DEFAULT NULL COMMENT '权限id',

PRIMARY KEY (`role\_authorty\_id`),

KEY `FK\_Reference\_10` (`authorty\_id`),

KEY `FK\_Reference\_3` (`role\_id`),

CONSTRAINT `role\_authorty\_info\_ibfk\_1` FOREIGN KEY (`role\_id`) REFERENCES `role\_info` (`role\_id`),

CONSTRAINT `role\_authorty\_info\_ibfk\_2` FOREIGN KEY (`authorty\_id`) REFERENCES `authorty\_info` (`authorty\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**8.role\_info**

-- ----------------------------

-- Table structure for `role\_info`

-- ----------------------------

DROP TABLE IF EXISTS `role\_info`;

CREATE TABLE `role\_info` (

`role\_id` varchar(40) NOT NULL COMMENT '角色id',

`role\_name` varchar(100) DEFAULT NULL COMMENT '角色名称',

`role\_descr` varchar(1000) DEFAULT NULL COMMENT '角色描述',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`modified\_time` datetime DEFAULT NULL COMMENT '修改时间',

`del\_tag` varchar(40) DEFAULT NULL COMMENT '删除标识（0.保存 1.删除）',

PRIMARY KEY (`role\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**9.user\_info**

-- ----------------------------

-- Table structure for `user\_info`

-- ----------------------------

DROP TABLE IF EXISTS `user\_info`;

CREATE TABLE `user\_info` (

`user\_id` varchar(40) NOT NULL COMMENT '用户id',

`role\_id` varchar(40) DEFAULT NULL COMMENT '角色id',

`user\_name` varchar(100) DEFAULT NULL COMMENT '用户名称',

`department` varchar(40) DEFAULT NULL COMMENT '用户所属部门',

`email` varchar(40) DEFAULT NULL COMMENT '电子邮箱',

`phone` varchar(40) DEFAULT NULL COMMENT '联系方式',

`idCard` varchar(40) DEFAULT NULL COMMENT '身份证号',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`modified\_time` datetime DEFAULT NULL COMMENT '修改时间',

`user\_descr` varchar(2000) DEFAULT NULL COMMENT '用户描述',

`user\_state` varchar(40) DEFAULT NULL COMMENT '用户状态（0.启用 1.禁用）',

`login\_account` varchar(100) DEFAULT NULL COMMENT '用户登录名',

`login\_password` varchar(100) DEFAULT NULL COMMENT '用户密码',

`del\_tag` varchar(40) DEFAULT NULL COMMENT '删除标识',

PRIMARY KEY (`user\_id`),

KEY `FK\_Reference\_5` (`role\_id`),

CONSTRAINT `user\_info\_ibfk\_1` FOREIGN KEY (`role\_id`) REFERENCES `role\_info` (`role\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## 关键流程代码及说明

系统基于SSM框架和MVC设计模式实现，基本的业务设计流程都是类似的，实现基本的增删改查功能，此处选择简单说明MVC设计流程和拦截器的内容

* 项目工程包说明：

实体类：com.zstu.pms.model

dao层：com.zstu.pms.dao

Service层：

用户管理：com.zstu.pms.user.service/com.zstu.pms.user.service.impl

系统设置：com.zstu.pms.system.service/com.zstu.pms.system.service.impl

申报管理：com.zstu.pms.declare.service/com.zstu.pms.declare.service.impl

项目审核：com.zstu.pms.audit.service/com.zstu.pms.audit.service.impl

消息管理：com.zstu.pms.message.service/com.zstu.pms.message.service.impl

Controller层：

用户管理：com.zstu.pms.user.controller

系统设置：com.zstu.pms.system.controller

申报管理：com.zstu.pms.declare.controller

项目审核：com.zstu.pms.audit.controller

消息管理：com.zstu.pms.message.controller

* 拦截器设置：实现登录拦截、权限控制（借助spring的拦截器实现）

**public** **class** LoginInterceptor **implements** HandlerInterceptor {

// 定义字符串输出日志信息

**private** **static** **final** Logger ***LOGGER*** = Logger.*getLogger*(LoginInterceptor.**class**);

**private** StringBuilder sb = **new** StringBuilder();

@Override

**public** **boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)

**throws** Exception {

// 获取用户请求的路径信息

String url = request.getRequestURI();

// 如果是登录、注册操作，则放行资源

**if** (url.indexOf("loginUI.action") >= 0) {

sb.append("LoginInterceptor拦截器-->执行操作：跳转登录页面操作，放行资源....");

***LOGGER***.info(sb.toString());

**return** **true**;

}

**if** (url.indexOf("login.action") >= 0) {

sb.append("LoginInterceptor拦截器-->执行操作：登录操作，放行资源....");

**return** **true**;

}

**if** (url.indexOf("registerUserUI.action") >= 0) {

sb.append("LoginInterceptor拦截器-->执行操作：跳转注册页面操作，放行资源....");

***LOGGER***.info(sb.toString());

**return** **true**;

}

**if** (url.indexOf("registerUser.action") >= 0) {

sb.append("LoginInterceptor拦截器-->执行操作：注册操作，放行资源....");

***LOGGER***.info(sb.toString());

**return** **true**;

}

/\*

\* if (url.indexOf("/login/") >= 0) {

\* sb.append("LoginInterceptor拦截器-->执行操作：登录操作，放行资源....");

\* LOGGER.info(sb.toString()); return true; } if (url.indexOf("/register/") >=

\* 0) { sb.append("LoginInterceptor拦截器-->执行操作：注册操作，放行资源....");

\* LOGGER.info(sb.toString()); return true; }

\*/

// 如果不是登录操作，则根据判断当前用户是否已经登录及相应的角色决定资源的放行

HttpSession session = request.getSession();

UserInfo loginUser = (UserInfo) session.getAttribute("loginUser");

**if** (loginUser != **null**) {

// loginUser不为null，说明用户已经登录则根据当前用户的角色管理相应的访问模块

sb.append("LoginInterceptor拦截器-->执行操作：当前用户" + loginUser.getUserName() + "已登录，放行资源....");

***LOGGER***.info(sb.toString());

// 主页页面访问

**if** (url.indexOf("/login/index.action") >= 0) {

sb.append("LoginInterceptor拦截器-->执行操作：跳转到首页....");

***LOGGER***.info(sb.toString());

**return** **true**;

}

// 登录退出

**if** (url.indexOf("/login/logout.action") >= 0) {

sb.append("LoginInterceptor拦截器-->执行操作：跳转到首页....");

***LOGGER***.info(sb.toString());

**return** **true**;

}

// 系统管理员角色能够操作所有的模块

String roleId = loginUser.getRoleId();

**if** (ConstantUtils.***ROLE\_ADMIN***.equals(roleId)) {

**return** **true**;

}

// 根据不同的模块结合用户角色决定相应的使用权限

**if** (url.indexOf("/system/") >= 0 || url.indexOf("/user/") >= 0) {

// 系统管理、用户管理模块只能超级管理员使用

**return** **false**;

}

**if** (url.indexOf("/pms/declaration") >= 0) {

// 申报管理模块供学生使用

**if** (ConstantUtils.***ROLE\_STUDENT***.equals(roleId)) {

**return** **true**;

}

**return** **false**;

}

**if** (url.indexOf("/audit/declaretype/") >= 0) {

// 项目审核的项目类型管理（中期管理员、结题验收管理员可以使用）

**if** (ConstantUtils.***ROLE\_MIDCHECK***.equals(roleId) || ConstantUtils.***ROLE\_ENDCHECK***.equals(roleId)) {

**return** **true**;

}

**return** **false**;

}

**if** (url.indexOf("/audit/approval/") >= 0) {

// 项目审核的立项检查管理（立项检查管理员可以使用）

**if** (ConstantUtils.***ROLE\_SET***.equals(roleId)) {

**return** **true**;

}

**return** **false**;

}

**if** (url.indexOf("/audit/midcheck/") >= 0) {

// 项目审核的中期检查管理（中期检查管理员可以使用）

**if** (ConstantUtils.***ROLE\_MIDCHECK***.equals(roleId)) {

**return** **true**;

}

**return** **false**;

}

**if** (url.indexOf("/audit/endcheck/") >= 0) {

// 项目审核的结题验收管理（结题验收管理员可以使用）

**if** (ConstantUtils.***ROLE\_ENDCHECK***.equals(roleId)) {

**return** **true**;

}

**return** **false**;

}

// 用户已登录，放行资源

**return** **false**;

}

// 如果用户没有登录则进行强制登录操作

sb.append("LoginInterceptor拦截器-->执行操作：当前用户未登录，强制登录....");

***LOGGER***.info(sb.toString());

request.getRequestDispatcher("/login/loginUI.action").forward(request, response);

**return** **true**;

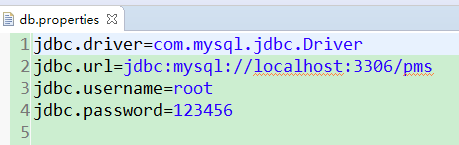
}

}

# 部署与演示

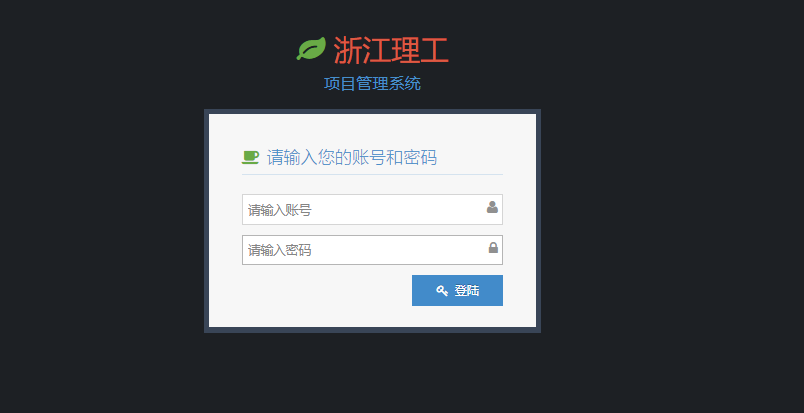
## 部署方式

连接mysql数据库，刷pms.sql脚本，修改PMS\src\main\resources\config\db.properties数据库连接的用户名、密码、连接、端口号，启动tomcat连接测试

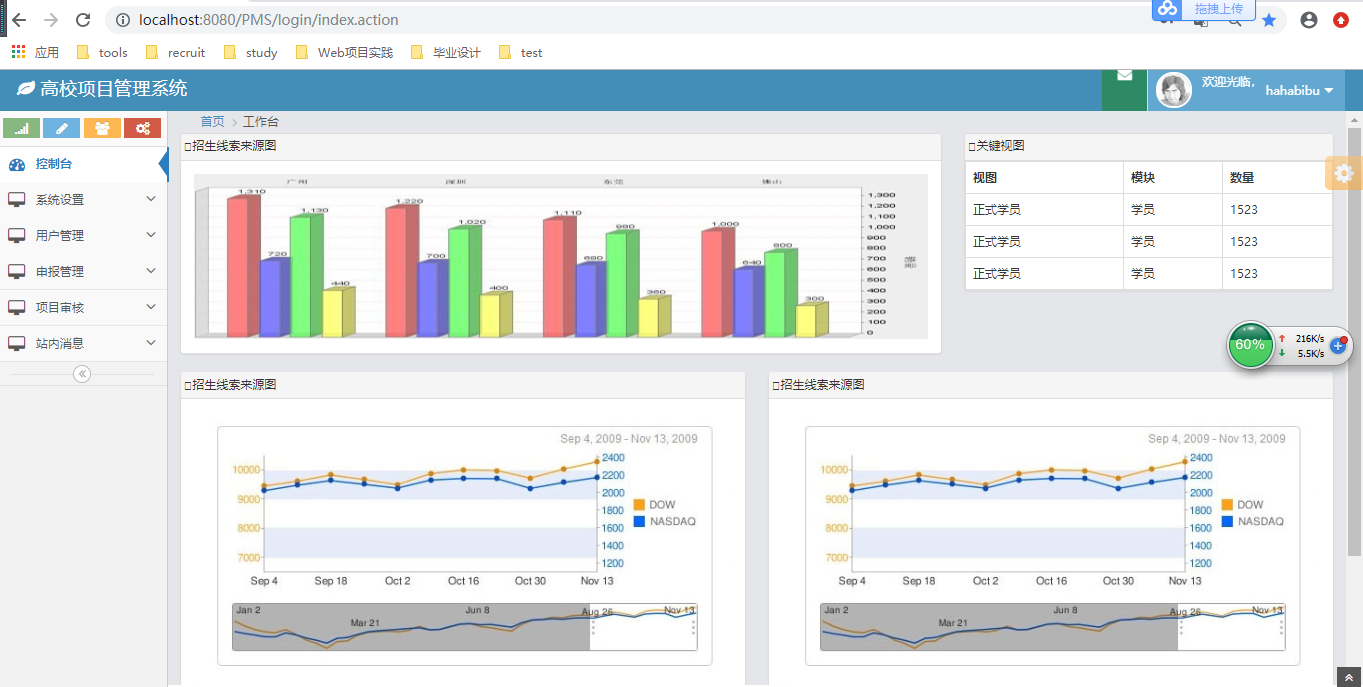


## 操作流程演示

### 系统登录模块

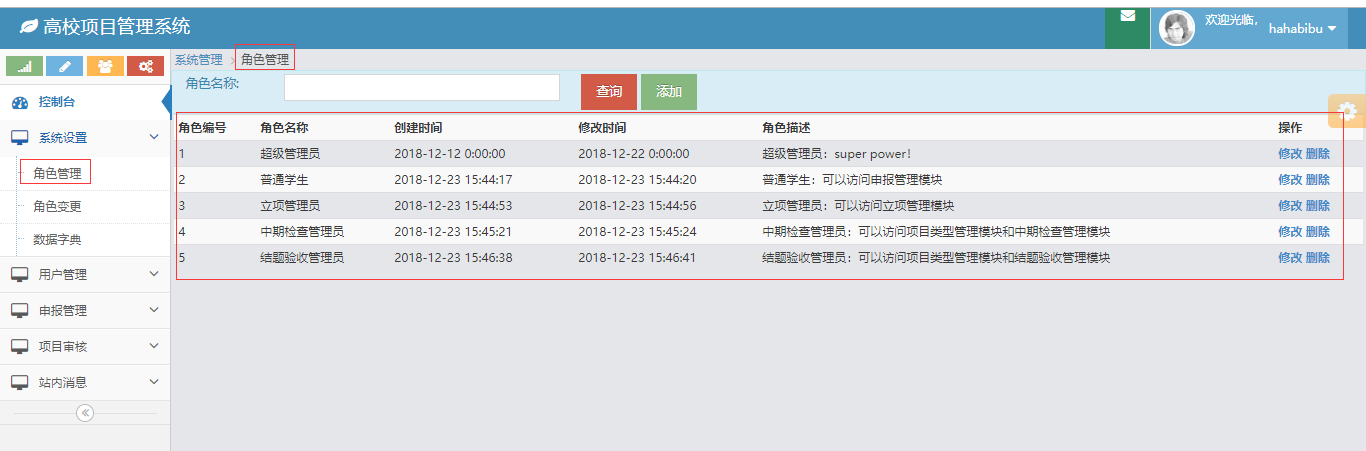


系统登录

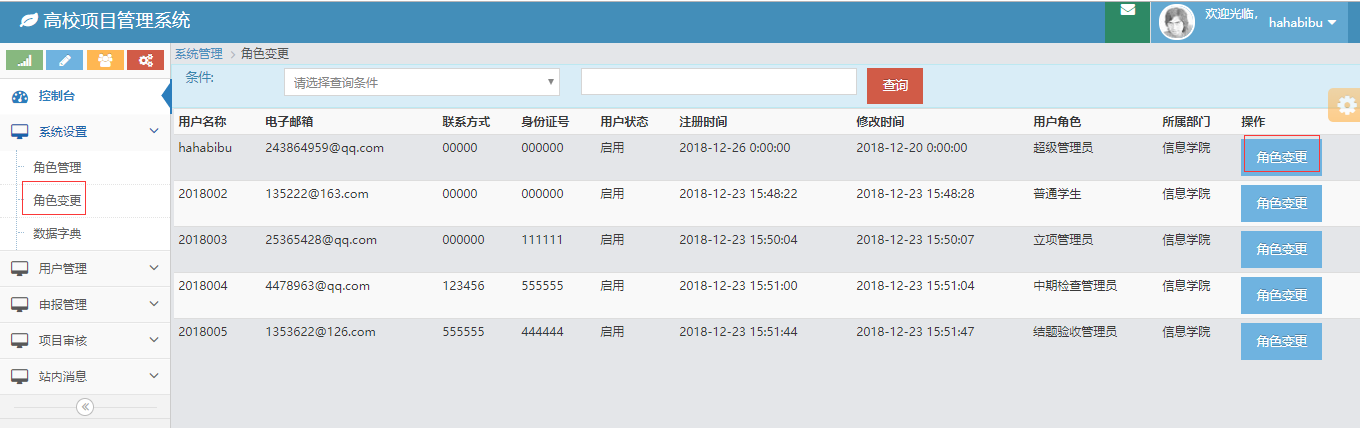


系统主页面

### 系统设置模块



角色管理

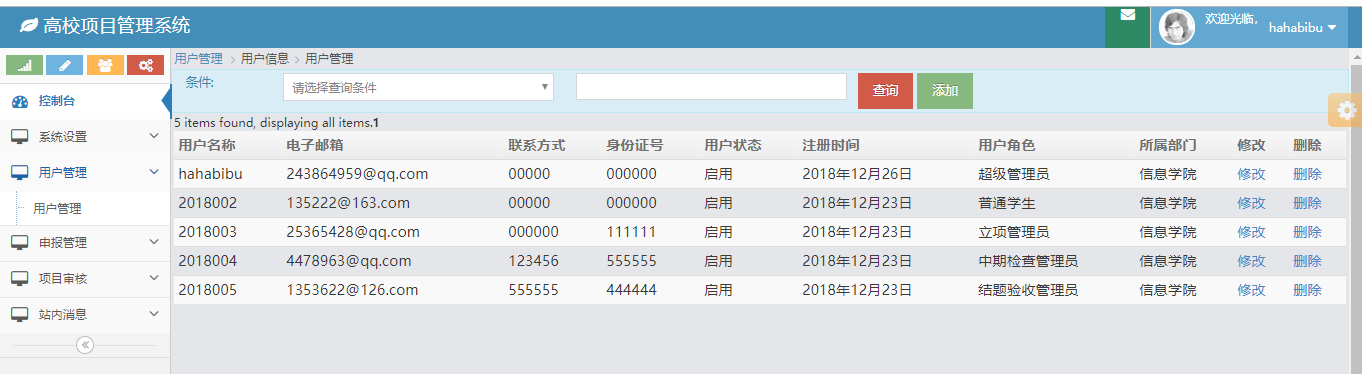


角色变更



数据字典管理

### 用户管理模块



用户信息管理

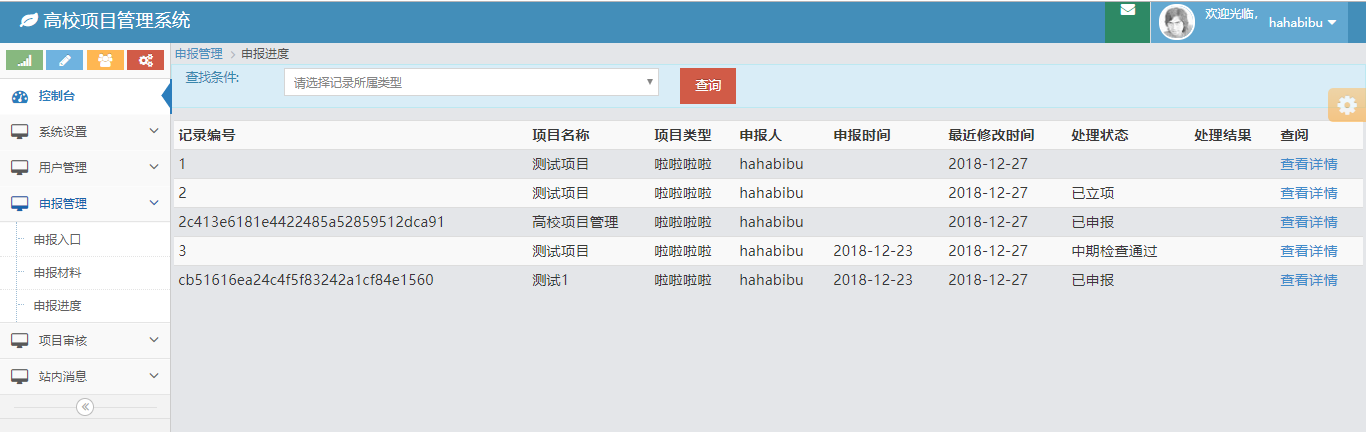
### 申报管理模块



申报入口



申报材料填写



个人申报记录查看

### 项目审核模块



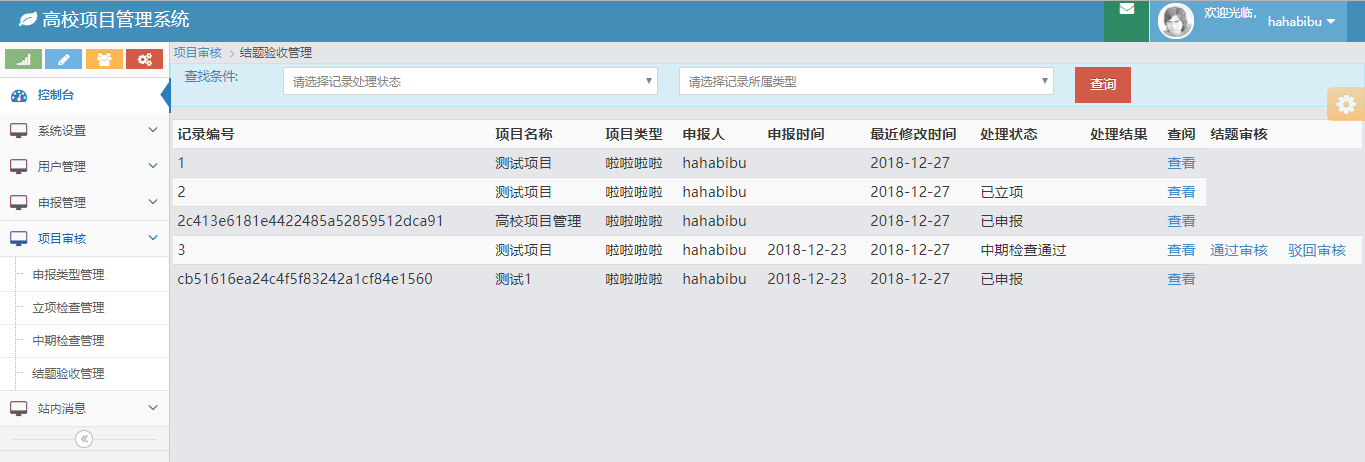
申报类型管理



立项检查管理

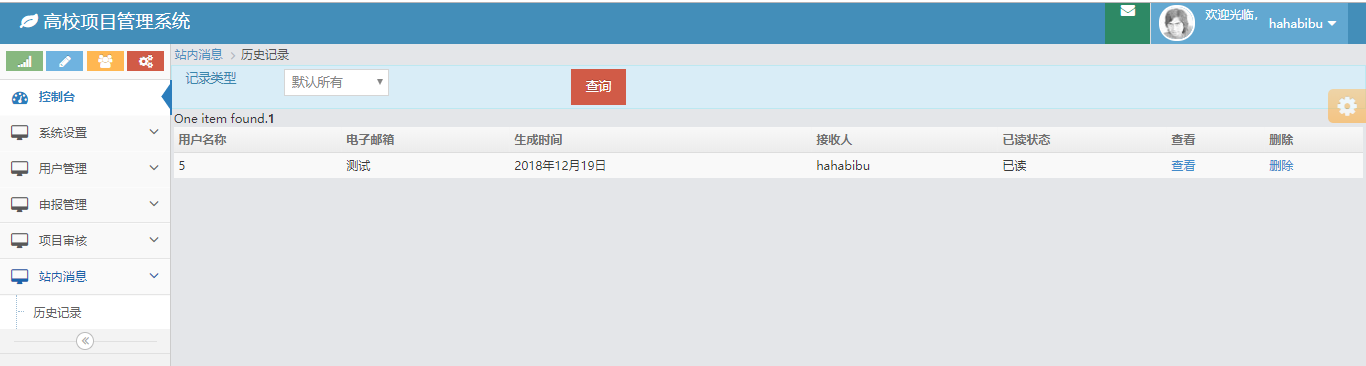


中期检查管理



结题验收管理

### 站内消息模块：



历史记录

# 心得与体会

这次的课程设计主题是高校项目管理系统，从一开始确定项目主题、进行系统规划到系统的可行性分析、需求分析，再到数据库的概念、逻辑、物理的分析、设计，结合实际业务需求，一步步建立起一个高校项目管理系统的数据库模型，虽然在这探索过程中每个成员对系统模块的设计有许多不同的看法和见解，但最后我们总能够通过查阅资料、结合实际情境去冷静思考、分析我们的问题所在，统筹兼顾，尽可能地去调整、完善我们的项目，从而更好地协调我们的项目设计工作。

由于考虑成员在时间和开发环境的协调上有诸多限制，且部分成员对git代码管理工具的使用并不熟悉，因此考虑设置代码规范，由各个成员按照一定的要求各自完成相应模块的设计，方便后期代码的统一整合。

通过这次综合性的数据库应用系统的设计，我们小组更好地意识到团队协作的重要性，更加意识到一个完整、合理的系统设计对于项目的实现来说具有何等重要的地位，由于一开始只是初步做个简单的设计方案，没有很好地进行详细的扩展和分析，导致在项目真正实现的时候发现了许多令人困惑的问题，在此过程中，我们重新审视了自己的数据库模型和系统设计，于是对数据库模型进行调整、优化，重新进行了数据库的建模和系统的详细设计。通过不断的探讨，对我们的系统设计进一步进行优化，尽可能地去完善我们的项目。

虽然最后我们小组也算是基本完成预期的功能，但由于时间和资源关系，加上前期遇到困难的阻碍，导致实现阶段有些延迟，对于部分功能我们只是做了一个简单的展示，也还有许多待开发的空间，之后会结合老师的意见和建议对我们的项目进行调整和分析。