**毕业设计（论文）**

**题目： 某市发展和改革委员会综合分析展示系统**

**目 录**

[**1 绪 论 1**](#_Toc12121)

[1.1 研究的背景及意义 1](#_Toc8638)

[1.1.1 选题的背景 1](#_Toc31682)

[1.1.2 国内外研究现状 1](#_Toc16136)

[1.1.3 研究的意义 2](#_Toc4719)

[1.2 系统目标 3](#_Toc19714)

[**2 需求分析 4**](#_Toc6565)

[2.1 业务需求 4](#_Toc11929)

[2.1.1主要业务流程 4](#_Toc28500)

[2.2 功能需求 11](#_Toc7635)

[2.2.1 角色分析 11](#_Toc12788)

[2.2.2 业务功能 11](#_Toc423)

[2.3 非功能需求 23](#_Toc28028)

[2.3.1环境需求 23](#_Toc23458)

[2.3.2 性能需求 24](#_Toc29933)

[2.3.3 安全需求 24](#_Toc21446)

[2.3.4 软件需求 25](#_Toc30014)

[2.3.5 硬件需求 25](#_Toc23615)

[2.3.6 外部接口需求 25](#_Toc29874)

[2.4 系统需求优先级 26](#_Toc29668)

[**3 总体设计 27**](#_Toc18914)

[3.1系统设计的原则 27](#_Toc1437)

[3.2 系统体系结构设计 28](#_Toc12282)

[3.3 系统功能结构设计 31](#_Toc5161)

[**4 数据库设计 33**](#_Toc2555)

[4.1 概念结构设计 33](#_Toc7382)

[4.1.1 设计思路 33](#_Toc13574)

[4.1.2 E-R图 33](#_Toc12578)

[4.2 逻辑结构设计 34](#_Toc2551)

[4.2.1 设计思路 34](#_Toc1646)

[4.2.2 逻辑模型 36](#_Toc28887)

[4.2.3 逻辑表格设计 37](#_Toc29727)

[4.3 物理结构设计 39](#_Toc15604)

[4.3.1 存取方式 39](#_Toc23516)

[4.3.2 存储结构 39](#_Toc32414)

[**5 界面设计 41**](#_Toc5337)

[5.1 界面关系图 41](#_Toc8149)

[5.2.1 主界面 42](#_Toc31302)

[5.2.2 子界面 42](#_Toc2932)

[**6 详细设计 45**](#_Toc23189)

[6.1 系统主要功能模块介绍 45](#_Toc28301)

[6.2 用户管理模块设计 45](#_Toc8779)

[6.3 数据主题管理模块设计 50](#_Toc15124)

[6.4 个性化定制模块设计 54](#_Toc3036)

[6.5 统计分析结果模块设计 59](#_Toc25225)

[6.6 展示综合联动模块设计 65](#_Toc29732)

[**7 编码 70**](#_Toc27739)

[7.1 代码实现与核心算法 70](#_Toc29392)

[7.2 代码优化分析 75](#_Toc28566)

[7.2.1 代码优化目的 75](#_Toc28803)

[7.2.2 具体优化实例 75](#_Toc20722)

[**8 测试 77**](#_Toc27438)

[8.1 测试方案设计 77](#_Toc31311)

[8.1.1 测试策略 77](#_Toc16928)

[8.1.2 测试进度安排 77](#_Toc7935)

[8.1.3 测试资源 77](#_Toc28569)

[8.1.4 关键测试点 78](#_Toc31641)

[8.2 测试用例构建 78](#_Toc992)

[8.2.1 测试用例编写约定 78](#_Toc13606)

[8.2.2 测试用例设计 79](#_Toc5682)

[8.2.3 关键测试用例 80](#_Toc23268)

[8.2.4 测试用例维护 83](#_Toc16899)

[**9 总结与展望 84**](#_Toc26355)

[9.1 设计工作总结 84](#_Toc15208)

[9.2 未来工作展望 84](#_Toc15789)

[**谢 辞 85**](#_Toc7071)

[**参考文献 86**](#_Toc26744)

[**附录A 外文翻译—原文部分 88**](#_Toc20242)

[**附录B 外文翻译—译文部分 94**](#_Toc28743)

[**附录C 软件使用说明书 97**](#_Toc6221)

[**附录D 主要源代码 100**](#_Toc7479)

# 1 绪 论

## 1.1 研究的背景及意义

### **1.1.1 选题的背景**

由于众所周知的原因，我国在一定时间段内已相对落后世界先进国家一段距离，通过四十余年的改革开放之后，目前我国与世界先进国家的差距越来越小，GDP曾一度超越日本，位列全球第二。然而，我国当前的发展依然具有一定的不平衡性，在政府部门的线上服务方面略逊色于发达国家。信息技术在西方国家拥有数十年的发展历史，线上服务功能已经发展得比较完善。目前正处于经济全球化的新阶段，应该在此处与世界领先水平齐平，从而消除发展的不对等性。

自2015年以来，在国内经济进入新常态的新形势下，根据国务院“稳增长、促改革、调结构、惠民生、补短板、防风险”总体要求，国家发改委重点着力推进重大项目建设，加快启动一批具有全局性、基础性、战略性的重大工程，加强协调调度，加快推进重大工程建设；同时加强政府投资项目储备，加快推进入库项目前期工作，逐步实现从“资金等项目”到“项目等资金”的转变，做实做细三年滚动计划。为此，国家发改委先后启动了投资项目在线审批监管平台和国家重大建设项目库系统等信息化系统的建设，在项目建设过程中提出并积累了许多新的思路和意见，从而推进地方发改委投资项目管理信息化建设步入快车道。

当前国家发改委已建设国家重大建设项目库，实现对全国三年滚动计划投资项目的管理。以内蒙古自治区为例，该自治区建设了自治区在线审批监管平台、网上稽查通道等信息化系统，以及旗县建设的投资项目管理系统。其下辖各市三年滚动计划项目、亿元以上五年规划项目、审核备项目等投资相关数据分散在多个系统中，不利于分析利用，亟需实现数据共享、融合、挖掘分析，为决策服务。

### **1.1.2 国内外研究现状**

所谓综合分析，指运用各种统计综合指标来反映和研究社会经济现象总体的一般特征和数量关系的研究方法。该研究方法存在数据量大的特点，单靠人工完成相关工作的效率相对较低，且存在较大的误差。随着目前计算机技术的发达，此类数据量大的研究方法可利用对应软件完成，开发者可利用软件的用户友好性让相关工作人员的工作量减轻不少。

在很长一段时间里，国内政府部门网站的使用率远不及常规网站，故网站设计者并不重视此类网站的研究与开发，部分网站甚至还停留在2.0时代，如此一来直接导致目前多数政府网站的设计均较为简单，有时候还会出现崩溃问题。自2012年开始，尤其是自2017年开始，由于网络的进步以及智能设备的全面普及，越来越多民众开始依赖互联网，从最初的娱乐需求变成了现在的办公需求。应广大人民的需求，国家在政府部门网站投入的资源也越来越多。现如今，政府部门网站的完善程度已经很高了，除了传统的公开政府的新闻资讯及最新政策之外，现在已经全面支持普通民众与政府机关工作人员的互动，从而反馈情况及办公。此外，单一机关网站可以实现省内跨部门跳转、异地相同单位跳转等功能，如此可避免在外部搜索引擎上耗费时间。

就目前而言，由于国外计算器技术发展历史悠久，虽说国内在近年以来于技术方面已经取得了很大的进步，但在某些方面相比国外系统还是有一定的差距。随着互联网技术的发展，当前国外大多数政府机构及国际组织已有其官网，可全天候运转，如此一来可让相关工作人员在网站上办公，而不需要考虑自身所处地点问题，在有如因疫情而需要居家隔离办公的条件下可起到非常大的作用。

我国已处于进入全面小康的决胜阶段，除常规的点对点脱贫之外，还需要建立完善的网络环境以为决胜阶段服务。根据中央各文件精神及当前群众的需要，数字化、便民化为当下政府机构亟需达到的目标，从而可让群众足不出户即可了解实时政府动态，提高群众对政府的满意度。

### **1.1.3 研究的意义**

为优化市发展与改革委员会的工作，把握项目投资运行趋势，总揽全市投资项目和发展的全局，为全市投资项目有效调控、管理决策和提供科学依据，紧跟时代步伐，当前迫切需要综合汇聚、有效集成、统一整合已有各个系统中投资项目及其相关的数据资源，充分运用一定的先进技术和理念，建立一套符合城市投资项目特色要求的综合分析与展现系统，从而实现对全市投资项目数据进行横向比对、纵向挖掘、统计分析、预测预警、审批监管，为相关领导和投资项目管理部门加强投资项目综合分析、预警监管、科学调节、辅助决策提供综合信息化服务。

设计一个使用SSM架构的发展与改革委员会综合分析展示系统，令其拥有当前项目库数据及其负责人的管理、项目个性化定制展示、项目投资等功能，并可通过条形统计图、扇形统计图等数学图表展示当前项目的具体数据。

在很长的一段时间以来，各类项目投资文件通常使用的是纸质稿，而传统的纸质文档存在易修改、易丢失及易损坏等问题，本系统在将文件数据化之后即可有效解决上述问题。此外，若相关材料存在错误，工作人员也可以快速将其修改之，如此大幅提升了工作效率。

## 1.2 系统目标

发展与改革委员会综合分析展示系统的主要面向对象为发改委工作人员、项目建立者、项目投资者。

为使投资者便于查询与及时修改投标项目的相关内容，且使民众便于了解当前发改委所投标的项目，作为用户的立项人员在将项目信息发布后遂由发改委相关负责人员进行审核，审核过后即可对公众展示之，存在特定参数的项目亦可通过图表形式呈现出其项目各参数信息。

利用当前编程语言及其相关框架，从而完成一个具有展示政府相关部门综合参数分析的系统。本系统提供用户管理模块、数据管理模块、维度与统计值配置管理模块，立方体计算与结果缓存模块，同比、环比、趋势计算模块，统计分析结果可视化展现模块，个性化定制展示模块，关键字搜索及结果展示模块，展示综合联动交互模块，具有管理用户、数据，以及展示各数据资料的功能。

# 2 需求分析

所谓需求分析(Requirement Analysis)，即客户对软件产品目标的高层次要求，是软件行业开发人员针对用户所需求的项目进行细致的调查研究与分析，准确地理解用户对项目所要求的功能、性能、可靠性等相应内容，并将上述内容转化为完整的需求定义，从而确定本系统具体如何实现的过程。

## 2.1 业务需求

业务需求(Business Requirement)，为整个项目的宏观需求，提供当前项目需要走的大方向，给出项目所需要达到的目标。

当前系统的业务需求如下：

1. 实现项目创立者进行项目管理、个性化设计等功能，完善项目具体信息；
2. 根据项目创立者所提供的信息形成特定的数学模型；
3. 允许投资者于主页查询当前系统中所收录的项目，若有青睐的项目则可为其投资；
4. 具有配套的系统后台管理维护功能，拥有管理员权限的用户允许对当前存在的项目、个性化设计、数学建模图进行管理，且可对系统内所有用户的个人信息进行处理及审核；
5. 具有合理的响应速度及一定的安全性；
6. 具有稳定且易维护特点。

### **2.1.1主要业务流程**

1. 用户管理活动图

现如今，大部分网站提供了用户功能，注册的用户拥有比游客更多的权限，

用户在登录注册之后，于该子系统完善其个人信息。在修改个人信息时，用户将自身信息填写完毕后提交，并传于管理员审核，若审核通过则使个人信息修改成功。

用户管理子系统活动图如图2-1所示。

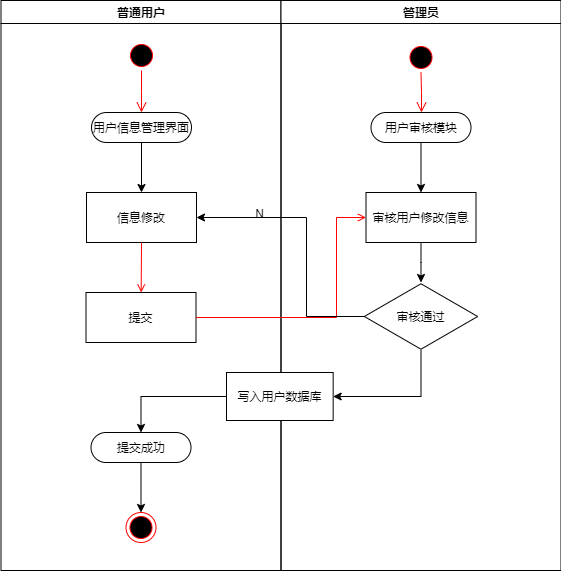
**

图2-1 用户管理子系统活动图

普通用户于用户信息管理界面处进行信息修改，在存在更改需求的特定的信息框内修改，完善信息后提交之，遂将信息传至用户审核模块。

拥有管理权限的管理员对普通用户提交的信息进行审核操作，若审核通过，点击通过选项，使得数据被传入于用户数据库。

当用户信息审核通过之后，则会在对应用户的个人主页处收到“信息审核通过”提醒；若审核未通过，则在对应用户的个人主页处收到“信息审核未通过”提醒，此时该具有信息修改需求的用户应当再次进行信息修改操作。

1. 数据主题管理活动图

与大多数用于数据管理的项目一样，数据主题管理是本系统最为核心的部分。所谓数据主题，即由立项者所立的项目。普通用户可以在该子系统处提交其所立项目，在完善项目具体信息后提交，提交后由管理员审核，审核通过即可展示。若现有项目存在一定问题，需要进行更改，则可以在该子系统内修改特定内容，并提交给管理员审核，审核通过即可展示。若该项目失去时效性，需要删除，普通用户则需要发出删除请求，待管理员审核后删除。此外，普通用户可在本系统内查找该账号已上传的项目。

数据主题管理子系统活动图如图2-2所示。

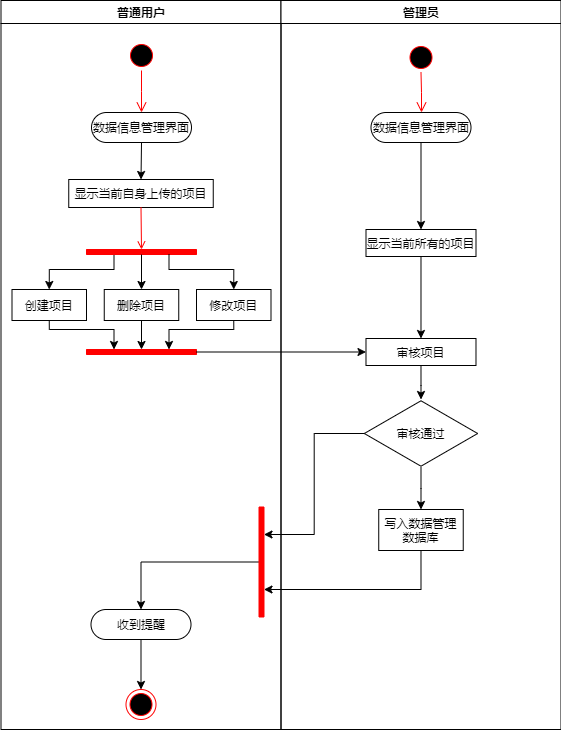


图2-2 数据主题管理子系统活动图

普通用户在进入数据主题管理界面后，先可浏览自身上传的所有项目，若未找到符合目前自己需求的项目，则可以使用搜索功能查询之。若自身当前并未提交项目，则会显示“无项目”字样，此时只能在该子系统内进行上传项目操作。若自身当前存在项目，则可以点击特定项目，修改该项目的具体内容。若当前项目失去时效性而因删除，普通用户则可发出删除请求。在普通用户发出上传、删除、修改的请求之后，都应由管理员进行进一步的审核。

管理员在进入数据主题展示界面后，可见当前所有用户上传的项目，其中待审核的项目旁会出现审核提醒，该内容待管理员审核通过后即可展示。若接到普通用户提出的删除请求，管理员需要在审核无误后确认删除。此外，管理员亦可在个性化展示界面中搜索当前本系统内上传的项目。

1. 个性化定制活动图

个性化定制，指代的是在生产过程中添加一部分突出特色的内容，从而赋予其吸引力的特质。在投标项目中，投资者需要充分了解所投资项目的具体内容，一般来说，具有特色的内容往往是投资者青睐的对象，立项者应当在需要投资的项目中表现出该有的特色，如此一来可有助于招商引资的成功。

普通用户在已有自身提交的数据主题的情况下，则可以针对特定项目制作相应的个性化内容，在提交后需由管理员审核，审核通过即可展示。

个性化定制子系统活动图如图2-3所示。

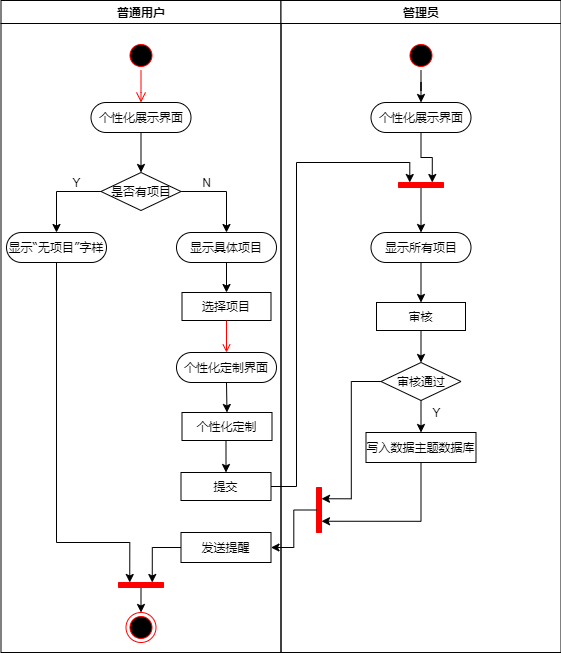


图2-3 个性化定制子系统活动图

普通用户在进入个性化展示界面后，若自身当前并未提交项目，则会显示“无项目”字样，此时将无法对该子系统进行操作。若自身当前存在项目，则可以点击特定项目，并于该项目的个性化定制界面中进行个性化定制，如上传文字说明、图片、音频等素材，随后提交，转于管理员审核。此外，普通用户可在个性化展示界面中搜索当前自身所上传的项目。

管理员在进入个性化展示界面后，可见当前所有用户上传的项目，其中待审核的项目旁会出现审核提醒，该内容待管理员审核通过后即可展示。此外，管理员亦可在个性化展示界面中搜索当前本系统内上传的项目。

1. 统计分析结果活动图

本系统利用数学图表直观性的特点，提供了针对特定数据主题的数据以图表化显示的功能。管理员通过设定一定的统计学函数，令该函数对当前数据主题数据库中的数据进行数学分析，以形成特定的数学图表，以上统计学函数将可在维度与统计值配置管理模块下完成设置。当数据主题数据库中有数据写入或修改时，该子系统会调用该数据库中的数据，进行一定的数学分析，形成特定的数学图表。上述计算过程将在立方体计算与结果缓存模块中完成，计算结果亦将缓存于此模块中。若需要计算该特定数据的同比、环比、趋势等数据，则会由同比/环比/趋势计算模块完成上述数据的计算过程。管理员可在可视化展现模块中，于后台浏览目前已生成的数学图表，也可以在此界面删除特定的数学图表。

统计分析结果子系统活动图如图2-4所示。

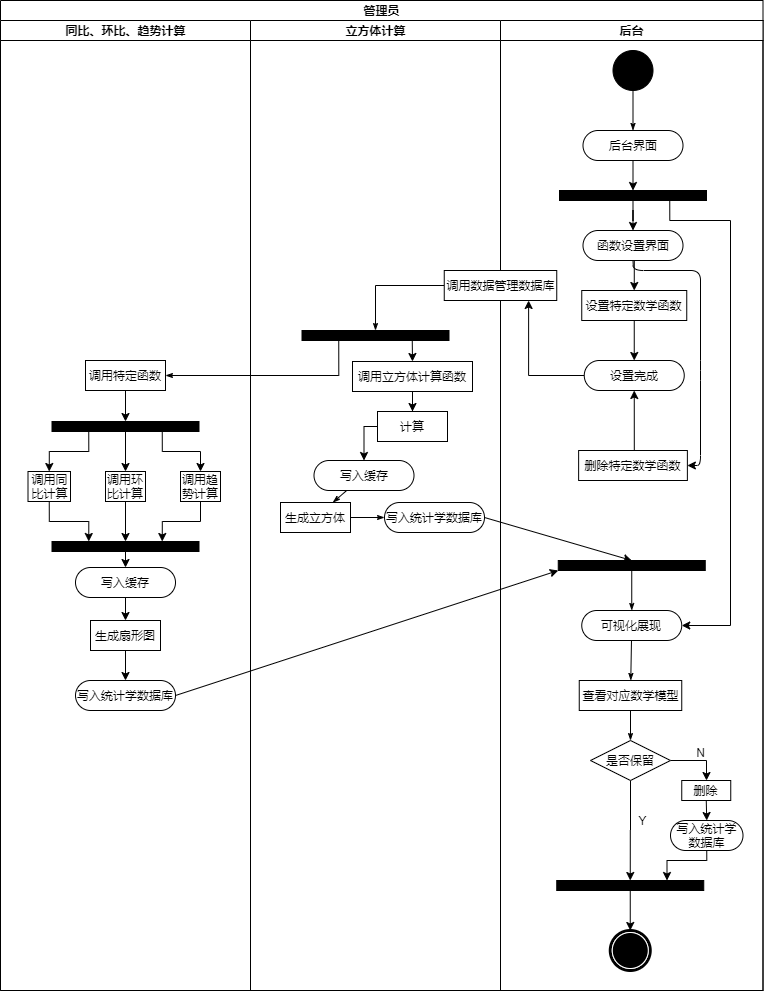


图2-4 统计分析结果子系统活动图

管理员进入后台界面后，可直接通过可视化展现模块查看当前统计学数据库中所含有的数学模型，若发现有不合适的模型则可删除之。

管理员也可以通过维度与统计值管理模块以在函数设置模块中配置相应的数学函数，亦可删除该数学函数。当数学函数发生修改后，此时将调用立方体计算函数、同比计算函数、环比计算函数、趋势计算函数，对现有数据管理数据库中的数据进行计算，写入缓存，并得出对应的数学模型，将该数学模型覆盖掉统计学数据库中原有的数学模型，于可视化展现模块展示之。

1. 展示综合联动

该模块为系统主页面，提供对当前数据库中的数据主题进行的查询功能，以及对项目的投资功能。

展示综合联动子系统如图2-5所示。

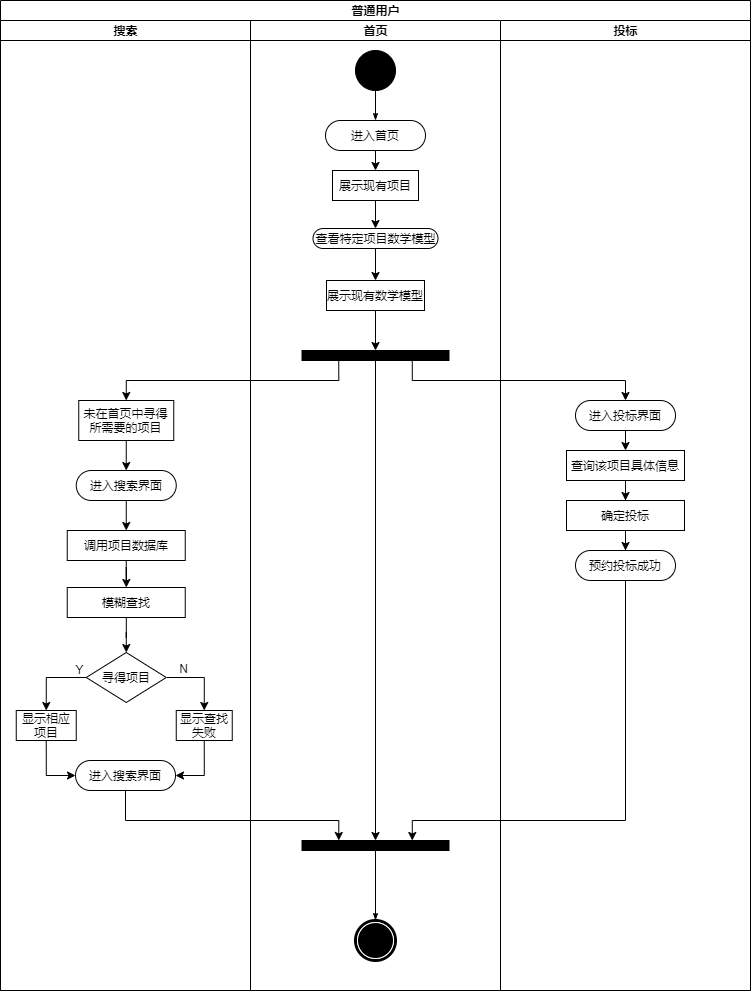


图2-5 展示综合联动子系统活动图

首页将调用数据主题数据库以及统计学数据库中的数据，并予以展示，若该数据主题绑定了个性化定制内容，则调用之。

普通用户在首页查看到当前所含有的项目，若有符合自身投资意愿的项目，可通过投标模块进行投标预约，随后由发改委相关工作人员联系投标者，进行下一步的合作洽谈事宜。

若未在首页寻得符合自身投资意愿的项目，则可通过关键字搜索模块进行模糊查询，直到符合自身需求为止。

## 2.2 功能需求

功能需求属于软件需求(Software Requirement)的一个子部分。软件需求包括用户需求(User Requirement)部分及功能需求(Functional Requirement)部分。用户将对所指定系统提出所有详细业务说明、业务流程等相关要求，形成用户需求。而功能需求则为软件开发人员针对用户需求归化分解为计算机可实现的子系统和功能模块，并用设计的语言来解释用户需求，从而达到指导系统设计的目的。

### **2.2.1 角色分析**

普通用户：通常为立项者，可更改自身用户信息，提交、修改其项目的数据主题，对自身项目进行个性化定制操作。

管理员：更改所有用户信息，对普通用户提交的用户信息更改请求、数据主题提交请求、数据主题更改请求、个性化定制提交请求进行审核操作。

游客：通常为投资者，可以在首页搜索其所需的项目，若有中意项目则可进行投标预约。

### **2.2.2 业务功能**

本系统主要由用户管理子系统，数据主题管理子系统，统计分析结果子系统，个性化定制展示子系统，展示综合联动子系统组成。

本系统的用户根据其权限差异共分为三类：普通用户、管理员、游客。其中，普通用户可对用户管理子系统、个性化定制展示子系统、数据主体管理子系统、展示综合联动子系统进行操作，管理员可对用户管理子系统、统计分析结果子系统、数据主体管理子系统、个性化定制展示子系统进行操作，游客只可对展示综合联动子系统进行操作。

本系统功能的大致描述如下：

1. 用户管理：根据用户登录时的权限差异来区分所进入的网页。若该用户为普通用户，则会进入自身的信息填写界面，可在此处进行登录密码修改的操作。若该用户为管理员，则除可进入更改自身信息的界面之外，亦可展示出当前所有用户的信息，若当前有用户提交了其个人信息修改，则需要由管理员在此页面进行审核，审核通过后反馈回普通用户处。
2. 数据主题管理：即项目管理，根据用户登录时的权限差异来区分所进入的网页。若该用户为普通用户，则可在自身账号处发布项目，若项目信息有误，亦可在此进行修改，若该项目失去时效性，则可在此删除之；此外，普通用户也可以在自身的数据主题库处进行查询操作。若该用户为管理员，其拥有查询的权限，待收到用户的数据主题提交或更新请求时对其内容进行审核，审核通过后反馈回普通用户处。
3. 统计分析结果：由于涉及到函数管理等内容，故此子系统只有管理员才有浏览权限。管理员将在该子系统中制定生成数学图表的函数，若管理员修改了某一函数，则该函数会在设置完成后对现有数据进行计算，并覆盖原有结果。该子系统涉及到维度、统计值、趋势、同比、环比、立方体计算等内容。
4. 个性化定制展示：根据用户登录时的权限差异来区分所进入的网页。普通用户在拥有自身创建的项目之后，可在该项目处进行个性化定制操作，如修改样式、插入图片、音频等。在管理员界面中出现的是所有项目及其所属用户的列表信息，以及该项目是否有个性化定制提交，待普通用户提交其个性化定制内容后，将由管理员对其进行审核，审核通过后反馈回普通用户处。
5. 展示综合联动：主要指代首页，此处将展示来自立项者于该发改委综合分析系统中发布的项目，首页展示内容具有时效性，若该项目不在首页内，则可通过搜索功能进行模糊查询，直到搜索出其所需项目为止。投标者可以对自己青睐的项目进行投标预约。

以下从业务角度出发，给出的系统总用例图如图2-6所示。

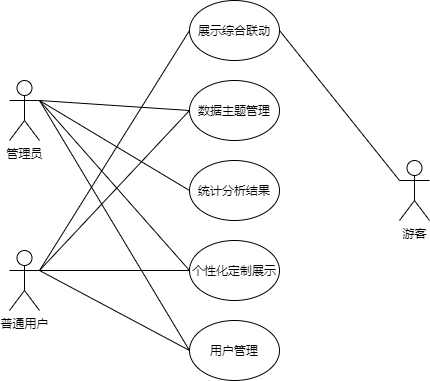


图2-6 系统总用例图

1. 用户管理

用户管理子系统用例图如图2-7所示。

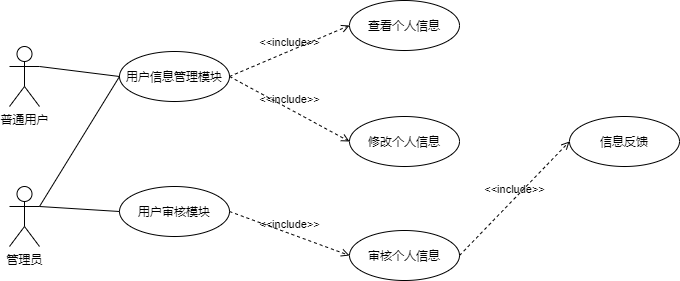


图2-7 用户管理子系统用例图

表2-1 用户管理子系统用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0001 | 用例名称 | 用户管理 |
| 用例描述 | 用户管理子系统包括用户信息管理模块和用户审核模块。  用户管理模块包括查看个人信息功能和修改个人信息功能，用户可在该模块中查看其目前已有的个人信息，若个人信息存在问题，亦可在此处进行修改，修改提交并待审核通过之后将更新，更新内容可在该模块内再次查看。  用户审核模块为管理员所使用的模块，其具有审核个人信息功能，管理员在收到普通用户于用户信息管理模块中提交的个人信息修改请求后，将对修改的个人信息具体内容进行审核，当审核无误后将通过审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。 | | |
| 执行者 | 普通用户、管理员 | | |
| 前置条件 | 用户存在自身账号且处于登录情况下 | | |
| 后置条件 | 允许普通用户进行数据主题提交操作 | | |
| 涉众利益 | 1. 个人信息泄露问题 2. 审核信息反馈即时性 3. 个人信息时效性 | | |
| 用例场景 | 用户管理界面 | | |
| 基本事件流 | **基本流程（Base Flow）**   1. 进入用户管理界面 2. 查看自身账号信息 3. 若账号信息有误、需要更改用户名及密码，则进入用户信息修改界面 | | |

续表2-1

|  |  |
| --- | --- |
| 基本事件流 | 1. 将具有时效性的个人信息或具有安全性的密码填入用户信息修改界面 2. 进入用户管理界面 3. 查看自身账号信息 4. 若账号信息有误、需要更改用户名及密码，则进入用户信息修改界面 5. 将具有时效性的个人信息或具有安全性的密码填入用户信息修改界面 6. 提交信息，交予管理员审核 7. 管理员对当前提交的信息进行审核工作 8. 若审核无误，则更新用户数据库，并自动向普通用户发送“审核通过”提醒 9. 若审核未过，则自动向普通用户发送，不对数据库进行任何操作 10. 用户可在用户管理界面查看到已更新的信息。   **扩展流程（Extend Flow）**  1a. 修改内容处存在空缺：  1a1. 系统提示个人信息不充分  1a2. 用例结束  2a. 验证码错误：  2a1. 提示重新输入  2a2. 用例结束  **字段列表（Filed List）**  用户信息 ＝ 用户名 ＋ 密码 ＋ 姓名 + 隶属单位 + 联系电话 + 联系地址  **业务规则（Business role）**   1. 用户名、姓名不得为空，最长不超过20个字符 2. 密码不得为空，最长不超过20个字符，最短不少于8个字符 3. 密码必须作加密处理 4. 联系电话必须由数字组成 5. 隶属单位不允许为空 6. 联系电话、联系地址允许不填 |
| 特殊需求（Special Requirement） | 不得于提交后在链接处出现用户密码此类低级错误 |
| 编写人 |  |

1. 数据主题管理

数据主题管理子系统用例图如图2-8所示。

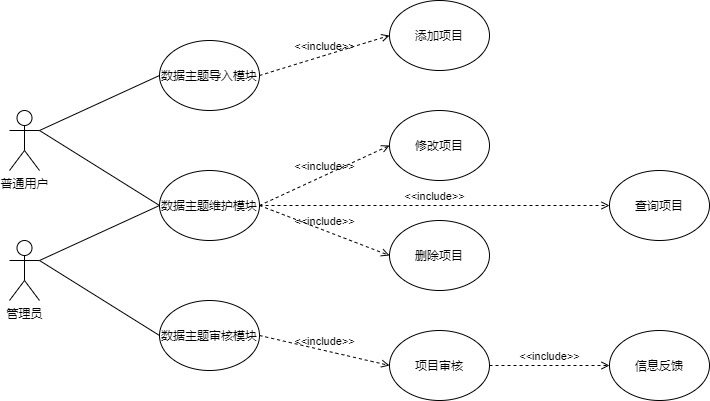


图2-8 数据主题管理子系统用例图

表2-2 数据主题管理子系统用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0002 | 用例名称 | 数据主题管理 |
| 用例描述 | 数据主题管理子系统包括数据主题导入模块、数据主题维护模块，以及数据主题审核模块。  数据主题导入模块包括添加项目功能。普通用户若需要立项，则可在该模块中提交自身提供的项目信息，待提交并待审核通过之后将成功上传，该上传内容可在数据主题维护模块内查看。  数据主题维护模块包括修改项目、删除项目以及查询项目功能。用户可在该模块中查看当前自身已提交的所有项目，若未找到，则可用模糊查询功能搜索之。若当前提交的项目存在一定的问题，需要修改其中一部分信息，则可进入特定项目的维护界面，对该项目进行信息维护工作，待提交并待审核通过之后将成功上传，该上传内容可在数据主题维护模块内查看。若当前存在失去时效性的项目，需要将其删除，则可在维护界面提交删除请求，待管理员审核通过后将成功删除。  数据主题审核模块为管理员所使用的模块，其具有审核项目信息功能，管理员在收到普通用户于数据主题导入模块及数据主题维护模块中提交的项目信息上传、修改请求后，将对上述符合条件的项目具体内容进行审核，当审核无误后将通过审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。此外，管理员在收到普通用户提交的项目删除请求后，将对该项目时效性等内容进行审核，审核无误后将通过删除审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。 | | |

续表2-2

|  |  |
| --- | --- |
| 执行者 | 普通用户、管理员 |
| 前置条件 | 自身已拥有账号，且处于登录状态 |
| 后置条件 | 允许对特定项目进行个性化定制操作、项目可在主页展示、数据将反馈至统计分析结果子系统进行可视化处理 |
| 涉众利益 | 1. 项目信息时效性 2. 项目信息准确性 3. 审核反馈即时性 |
| 用例场景 | 数据主题管理界面 |
| 基本事件流 | **基本流程（Base Flow）**   1. 进入数据主题管理界面 2. 查看当前自身提交的所有项目信息 3. 若未找到自己所需项目，则使用模糊查询功能，显示对应项目 4. 若需要添加新的项目，则进入项目上传界面 5. 填写项目具体信息 6. 提交信息，交予管理员审核 7. 若需要修改特定项目的具体信息，则进入该项目的项目信息修改界面 8. 将具有时效性的项目信息填入项目信息修改界面 9. 提交信息，交予管理员审核 10. 若需要删除特定项目，则点击删除请求，交予管理员审核 11. 管理员对当前提交的信息进行审核工作 12. 若审核无误，则更新数据主题数据库，完成特定的上传、删除、修改操作，并自动向普通用户发送“审核通过”提醒 13. 若审核未过，则自动向普通用户发送，不对数据库进行任何操作 14. 用户可在数据主题管理界面查看到已更新的信息。   **扩展流程（Extend Flow）**  1a. 修改内容处存在空缺：  1a1. 系统提示项目信息不充分  1a2. 用例结束  2a. 验证码错误：  2a1. 提示重新输入  2a2. 用例结束  3a. 未查询到项目  3a1. 提示“未查询到项目”  3a2. 用例结束  **字段列表（Filed List）**  项目信息 = 项目名 + 项目类型 + 项目简介 + 项目所需资金 + 对口单位 + 立项人 + 立项人联系电话 + 项目所需施工人数  **业务规则（Business role）**   1. 项目名、项目类型、立项人不得为空，最长不超过20个字符 |

续表2-2

|  |  |
| --- | --- |
|  | 1. 立项人联系电话、项目所需施工人数必须由数字组成 2. 对口单位不允许为空 3. 项目简介、项目所需资金、项目所需施工人数允许为空 |
| 特殊需求（Special Requirement） | 暂无 |
| 编写人 |  |

1. 个性化定制展示

个性化定制展示子系统用例图如图2-9所示。

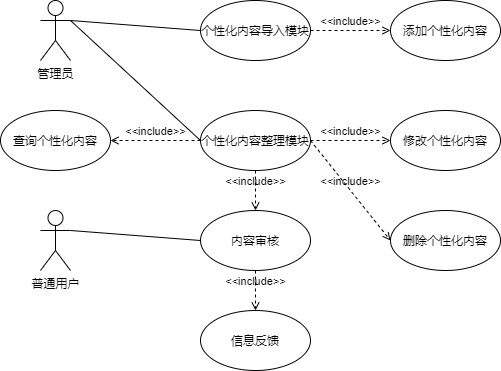


图2-9 个性化定制展示子系统用例图

表2-3 个性化定制展示子系统用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0003 | 用例名称 | 个性化定制展示 |
| 用例描述 | 个性化定制展示子系统包括个性化内容导入模块以及个性化内容整理模块。  个性化内容导入模块包括添加个性化内容功能。普通用户若需要在自身已提交的项目为基础之上进行个性化包装，则可在该项目对应的该模块中提交自身提供的个性化内容，待提交并待审核通过之后将成功上传，该上传内容可在个性化内容整理模块内查看。  个性化内容整理模块包括审核、修改个性化内容、删除个性化内容以及查询个性化内容功能，与数据主题维护模块功能大体相当。普通用户可在该模块中查看当前自身已提交的所有项目，若未找到，则可用 | | |

续表2-3

|  |  |
| --- | --- |
|  | 模糊查询功能搜索之。若需要对当前特定项目进行个性化内容修改，则可进入特定项目的个性化定制界面，对该项目进行个性化定制操作，待提交并待审核通过之后将成功上传，该上传内容可在个性化内容整理模块内查看。若需要删除某个个性化内容，则可在整理界面提交删除请求，待管理员审核通过后将成功删除。若管理员进入该模块，则会在其页面中显示所有已上传个性化内容的项目，其具有审核个性化内容功能，管理员在收到普通用户于个性化内容导入模块及个性化内容整理模块中提交的项目信息上传、修改请求后，将对上述符合条件的项目具体内容进行审核，当审核无误后将通过审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。此外，管理员在收到普通用户提交的个性化内容删除请求后，将在审核无误后将通过删除审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。 |
| 执行者 | 普通用户、管理员 |
| 前置条件 | 对应账号已提交过数据主题 |
| 后置条件 | 可在主页展示 |
| 涉众利益 | 项目的个性化体现 |
| 用例场景 | 个性化定制展示界面 |
| 基本事件流 | **基本流程（Base Flow）**   1. 进入个性化定制展示界面 2. 查看当前自身提交的所有项目信息 3. 若未找到自己所需项目，则使用模糊查询功能，显示对应项目 4. 若需要为特定项目添加个性化内容，则进入个性化定制界面 5. 为该项目进行个性化定制，写入部分信息 6. 提交信息，交予管理员审核 7. 若需要修改特定项目的个性化内容，则进入该项目的个性化定制界面 8. 为该项目进行个性化定制，写入部分信息 9. 提交信息，交予管理员审核 10. 若需要删除特定项目的个性化内容，则点击删除请求，交予管理员审核 11. 管理员对当前提交的信息进行审核工作 12. 若审核无误，则更新数据主题数据库，完成特定的上传、删除、修改操作，并自动向普通用户发送“审核通过”提醒 13. 若审核未过，则自动向普通用户发送，不对数据库进行任何操作 14. 用户可在个性化定制展示界面查看到已更新的个性化内容。   **扩展流程（Extend Flow）**  1a. 当前账号并无提交项目：  1a1. 系统提示无项目  1a2. 用例结束 |

续表2-3

|  |  |
| --- | --- |
|  | 2a. 验证码错误：  2a1. 提示重新输入  2a2. 用例结束  3a. 未查询到项目：  3a1. 提示“未查询到项目”  3a2. 用例结束  **字段列表（Filed List）**  个性化定制内容 = 项目信息 + 项目展示图片 + 音频  **业务规则（Business role）**  允许不进行个性化定制，若三项均未填，则不发出审核请求 |
| 特殊需求（Special Requirement） | 暂无 |
| 编写人 |  |

1. 统计分析结果

统计分析结果子系统用例图如图2-10所示。

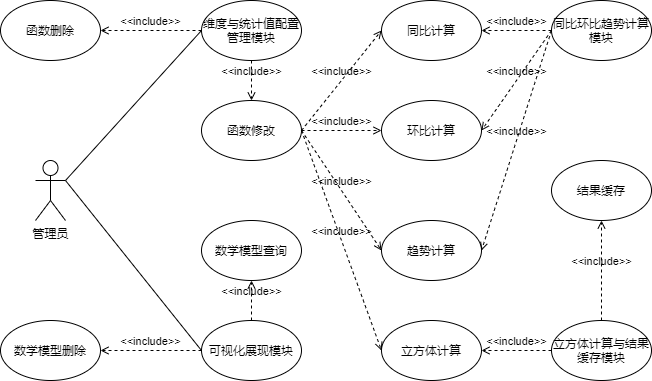


图2-10 统计分析结果子系统用例图

表2-4 统计分析结果子系统用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0004 | 用例名称 | 统计分析结果 |
| 用例描述 | 统计分析结果子系统包括维度与统计值配置管理模块、同比环比趋势计算模块、立方体计算与结果缓存模块以及可视化展现模块。  维度与与统计值配置管理模块拥有函数修改及函数删除功能。管理员可在此模块下对本系统所涉及到的函数进行管理，若某个函数表达式有误，则可以进入特定函数的修改界面进行修改操作。若某个函数表 | | |

续表2-4

|  |  |
| --- | --- |
|  | 达式失去时效性，则可以删除该表达式。  立方体计算与结果缓存模块包括立方体计算及结果缓存功能。管理员在设置好新的函数之后，或普通用户提交新的项目之后，将由该模块对现有数据主题数值进行计算处理，并生成特定的立方体统计图，其中间结果将缓存至缓存区域中。  同比环比趋势计算模块，顾名思义，包括同比计算、环比计算以及趋势计算三个功能。管理员在设置好新的函数之后，或普通用户提交新的项目之后，将由该模块对现有数据主题数值进行计算处理，并生成特定的同比、环比、趋势数据。  可视化展现模块拥有数学模型查询及数学模型删除功能。经上述两个模块处理后所生成的数学图表将在该界面显示，并由管理员审核，若审核无误，则可令该数学图表发布至主页上。 |
| 执行者 | 管理员 |
| 前置条件 | 无 |
| 后置条件 | 数据处理后的图表可展示至主页 |
| 涉众利益 | 1. 数学图表的直观性 2. 数学图表的低误差 3. 统计学函数的准确性 |
| 用例场景 | 可视化展现界面 |
| 基本事件流 | **基本流程（Base Flow）**   1. 进入可视化展现界面 2. 查看当前已生成的数学图表 3. 若需要查看函数，则进入维度与统计值配置界面 4. 查看统计学函数 5. 若需要修改函数，则进入特定函数配置界面 6. 修改函数表达式，提交 7. 函数修改成功后，由立方体计算、同比计算、环比计算、趋势计算等函数对现有数据主题数值进行重新计算，生成新的图表 8. 新的图表展示于可视化展现界面中 9. 若需要删除特定图表，点击删除按键   **扩展流程（Extend Flow）**  1a. 当前无用户上传项目：  1a1. 系统提示无项目  1a2. 用例结束  2a. 函数表达式修改错误：  2a1. 提示重新输入  2a2. 用例结束  **字段列表（Filed List）**  函数信息 = 函数名 + 函数表达式 |

续表2-4

|  |  |
| --- | --- |
| 基本事件流 | **业务规则（Business role）**   1. 不得删除所有函数表达式 2. 函数表达式中不可出现汉字   3. 不得删除默认函数（同比计算、环比计算、立方体计算、趋势计算）  **基本流程（Base Flow）**   1. 进入可视化展现界面 2. 查看当前已生成的数学图表 3. 若需要查看函数，则进入维度与统计值配置界面 4. 查看统计学函数 5. 若需要修改函数，则进入特定函数配置界面 6. 修改函数表达式，提交 7. 函数修改成功后，由立方体计算、同比计算、环比计算、趋势计算等函数对现有数据主题数值进行重新计算，生成新的图表 8. 新的图表展示于可视化展现界面中 9. 若需要删除特定图表，点击删除按键   **扩展流程（Extend Flow）**  1a. 当前无用户上传项目：  1a1. 系统提示无项目  1a2. 用例结束  2a. 函数表达式修改错误：  2a1. 提示重新输入  2a2. 用例结束  **字段列表（Filed List）**  函数信息 = 函数名 + 函数表达式  **业务规则（Business role）**   1. 不得删除所有函数表达式 2. 函数表达式中不可出现汉字 3. 不得删除默认函数（同比计算、环比计算、立方体计算、趋势计算） |
| 特殊需求（Special Requirement） | 函数计算时间应保持在10s反应时间之内 |
| 编写人 |  |

1. 展示综合联动

展示综合联动子系统用例图如图2-11所示。

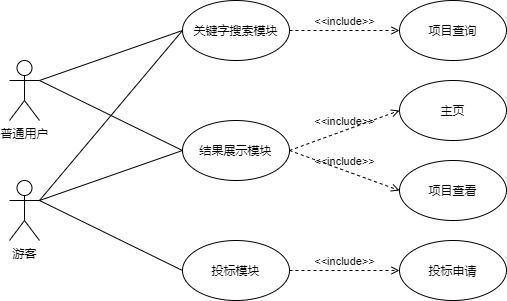


图2-11 展示综合联动子系统用例图

表2-5 展示综合联动子系统用例描述

|  |  |  |  |
| --- | --- | --- | --- |
| 用例编号 | 0005 | 用例名称 | 展示综合联动 |
| 用例描述 | 展示综合联动包括结果展示模块、关键字搜索模块，以及投资模块。  结果展示模块位于主页，拥有项目查看功能。用户可以在该模块查看特定项目的具体信息。  关键字搜索模块包括项目查询功能。若用户未在首页找到自己所需要的项目，则可通过该模块进行模糊查询，搜索自己所需要的项目，并查看之。  投资模块包括投资申请功能。若为游客权限的投资者找到了自己感兴趣的项目，需要投资，则需要在特定项目的投标界面中输入投标者自身的详细个人信息，并提交投标信息，待发改委工作人员审核并以电话形式反馈结果。 | | |
| 执行者 | 普通用户、游客 | | |
| 前置条件 | 当前存在项目 | | |
| 后置条件 | 无 | | |
| 涉众利益 | 1. 项目时效性 2. 投资者资金稳定性 3. 发改委工作人员审核即时性 4. 投资者与立项者之间的联系稳定性 | | |
| 用例场景 | 系统登录界面 | | |
| 基本事件流 | **基本流程（Base Flow）**   1. 进入主页 2. 主页展示一定量的项目及其简介 3. 若用户未找到所需要的项目，则使用模糊查询功能搜索项目，直到发现自己所需要的项目为止。 4. 查看特定项目的具体信息 | | |

续表2-5

|  |  |
| --- | --- |
|  | 1. 若用户对该项目存在兴趣，需要投资，则进入投资申请界面 2. 填写投资者具体信息 3. 提交投资申请   **扩展流程（Extend Flow）**  1a. 模糊查询无结果：  1a1. 系统提示“未查询到项目”  1a2. 用例结束  2a. 投资人信息不完全：  2a1. 提示重新输入  2a2. 用例结束  3a. 验证码输入有误：  3a1. 提示“验证码输入有误”  3a2. 用例结束  **字段列表（Filed List）**  投标人员信息 = 投标人姓名 + 所属单位 + 投标项目 + 联系电话 + 联系地址 + 对接负责人员 + 对接联系电话  **业务规则（Business role）**   1. 投资人姓名不得为空，且不超过20个字符 2. 投资项目、对接负责人员不得为空 3. 联系电话、对接联系电话需由数字组成   所属单位、联系地址允许为空 |
| 特殊需求（Special Requirement） | 暂无 |
| 编写人 |  |

## 2.3 非功能需求

非功能需求(Non-Functional Requirement)包括产品必须遵从的标准、规范和合约；外部界面的具体细节和性能要求；设计或实现的约束条件及质量属性。

### **2.3.1环境需求**

表2-6 服务器端环境

|  |  |
| --- | --- |
| 系统所使用数据库 | 5.7.21-log MySQL Community Server (GPL) |
| 操作系统 | Windows 10 |
| Web服务器 | Tomcat 9.0.6 |

表2-7 客户端环境

|  |  |
| --- | --- |
| 操作系统 | Windows 10或其以上版本操作系统 |
| 浏览器 | Google Chrome 57.0.2987.110 |
| 分辨率 | 最佳效果为1024×768像素 |

### **2.3.2 性能需求**

1. 本系统和现在主流的B/S架构的系统基本风格一致，能给用户提供更好的体验，而且本系统反应速度在1-2s内，系统操作平均响应时间不超过4.0s，页面响应时间≤50ms，事务处理时间≤0.5s，查询等待时间≤1.0s。
2. 系统平均年故障时间应控制在1h以内，即可用性达到99.99%，操作平均响应时间不超过3.0s。
3. 系统应有成熟的出错处理机制，以便在出错时会有及时的弹窗提醒用户，让用户能根据提示解决系统的出错问题，并且用户的数据可以在此期间得到保障，同时保留日志文件，方便后期彻底解决该问题。
4. 系统应提供运行监控和故障恢复机制，建立系统运行日志文件，能跟踪系统的所有操作。
5. 各业务系统对用户的操作顺序、输入的数据进行正确性检查，并以显著方式提示错误信息。
6. 在系统设计时，应当为以后该项目的新增模块以及优化功能提供扩展几口，保证系统具有一定的扩展能力。
7. 前端界面采用Bootstrap和Jquery搭配，采用了Bootstrap中的响应式布局，界面简约，自适应于窗口大小，可以提供良好的用户体验。后台界面采用主流的Easyui框架，提供扁平化界面，操作符合用户习惯。

### **2.3.3 安全需求**

1. 系统自身容错性

当用户输入或误操作导致非法数据产生时，系统应具有一定的容错机制。在这种情况下，系统应给出友好的提示，提示用户重新输入或者进行自动的修复校正。

1. 系统的外在环境安全

安全系统要以充分考虑网络的高级别，多层次的安全性措施为前提，包括系统的备份，防火墙，用户权限和其他措施，以确保数据安全和机密信息不被泄露；考虑到系统的硬件和软件故障恢复等应急措施，以保障网络的安全和处理安全性。形成相对独立的安全机制，以防止来自系统外的未经授权的访问。

1. 系统内部安全

确保外部系统安全的同时，本系统还必须确保授权用户的合法使用。

1. 系统运行安全

从逻辑上讲，本系统应具有抵抗非法入侵的能力。

### **2.3.4 软件需求**

本系统基于Java技术SSM框架开发，由于JVM的跨平台性，使得本系统可以直接在Windows、Linux、MacOS三大主流操作系统上进行部署运行，而无需大幅修改代码来适应不同平台，实现了良好的可扩展性和跨平台性。同时本系统整体还采用开源框架及技术实现，并且对运行环境的配置要求不高，同时在后期维护上不需要更多的人力成本，具备良好的经济可行性。

### **2.3.5 硬件需求**

本系统基于B/S架构，由服务器端和客户端（浏览器）组成，不同部分的具体运行环境如下表2-8所示。

表 2-8 硬件需求

|  |  |
| --- | --- |
| 要求 | 内容 |
| 操作系统 | Windows 7及以上版本 |
| JDK | 1.8及其以上 |
| 应用服务器 | Apache Tomcat 8.0及以上 |
| 应用服务器 | Mysql 5.7及以上版本 |
| 开发框架 | Spring MVC、Spring、Mybatis、Bootstrap、Jquery等 |

### **2.3.6 外部接口需求**

(1) 硬件需求

服务器：

处理器（CPU）：Pentium 2G

内存容量（RAM）：2G以上

客户端：

处理器（CPU）：Pentium 1G以上

内存容量（RAM）：256M以上

1. 软件接口

数据库接口：

MySQL数据库，接口8081

Tomcat：

版本9.0.6

浏览器接口：

谷歌Chrome版本57.0.2987.110

网络配置模拟器：

Cisco Packet Chaser版本7.0.0.0202

1. 硬件接口

USB 3.0接口

HDMI / Micro HDMI / Mini HDMI接口

1. 通讯接口

TCP/IP通信协议接口

GSM/CDMA无线通信协议接口

## 2.4 系统需求优先级

表2-9 系统需求优先级

|  |  |  |  |
| --- | --- | --- | --- |
| **需求编号** | **功能名称** | **优先级** | **说明** |
| 0001 | 用户管理 | 1 | 本系统核心模块之一，管理用户相关的功能 |
| 0002 | 数据主题管理 | 1 | 本系统核心模块之一，管理项目相关的功能 |
| 0003 | 个性化定制展示 | 3 | 为项目进行个性化定制的功能 |
| 0004 | 统计分析结果 | 2 | 统计数据的功能 |
| 0005 | 展示综合联动 | 1 | 主页，投资功能 |

3 总体设计

## 3.1系统设计的原则

在充分考虑到本系统的总体规划目标，以及系统的应用环境前提下，本系统的设计将严格遵循以下的设计原则：

1. 实用性原则。政要网站是当今政府的门户之一，其优劣直接影响到政府的面貌。故在进行本系统的设计时，需要将系统的实用性要求排在首位。本系统力求能够紧密切合政府机关和普通群众的实际业务需求，以求彻底解决传统群众办公模式的效率低的通病。
2. 先进性原则。在强调系统的实用性同时，在设计过程中还要考虑系统的先进性。时至今日，互联网信息技术的发展日新月异，故在每一次信息技术上的革新之上都会推动信息化建设的大幅度发展。故为与时俱进，有一个较高的起点，在本系统的设计过程中应该采用目前较为先进和成熟的技术和理念，从而使系统在性价比、同类软件上拥有一定的优势。
3. 规范性原则。系统中的所有设计均遵循现有的国家标准、行业标准，符合软件工程中的设计原则与规范。所有文档的书写均需要遵照国标《软件需求说明书（GB856T—88）》要求。
4. 安全性原则。系统提供安全机制以防止非法授权操作，为各子系统提供严格灵活的用户和权限划分，对于数据库管理，建立起完备的数据备份恢复机制。对关键的信息的操作提供日志记录。
5. 易于集成的原则。出于对发改委信息化建设的全盘考虑和整体规划，本系统要能够与其他业务系统进行集成。本系统为其他系统预留有应用接口和数据传输规范，方便其他系统与我系统间的信息交互。本系统在数据库设计上同样遵循此原则，在数据访问层以WebServices的方式提供数据访问服务。
6. 可扩展性原则。出于对后续升级工作的考虑，系统的所有功能模块的设计都预留了扩展接口。因为人们对政府机关单位办公的需求和应用目标会随着技术的进步和政策的调整而发生转变，在设计初期就考虑到可能发生的变化，是十分必要的。

## 3.2 系统体系结构设计

目前的系统设计中，主要采用的B/S结构及三层架构，使用SSM框架集，下面简单谈谈B/S结构、三层架构及SSM框架集的特点。

B/S结构，即所谓的“浏览器(Browser)/服务器(Server)”结构，浏览器(Browser)是指客户端的应用程序的都要在浏览器上运行，也就是说，客户端若需要运行此类结构的应用程序，客户端则至少需要安装一个浏览器软件。服务器(Server)是指代开发好的应用程序只需要部署在服务器上运行即可，客户端通过运行浏览器，输入指定的服务器IP地址，即可使用应用程序。B/S结构的特点包括客户端免安装、更新快、容易集中控制、安全风险大、响应速度慢且需要靠硬件提升等特点。

三层架构，自上而下分别为界面层(User Interface Layer)、业务逻辑层(Business Logic Layer)，以及数据访问层(Data Access Layer)。在此其中，数据访问层用于访问程序中的文件并实现对数据库的增删改查基本操作，业务逻辑层将用户输入的数据进行处理并保存，界面层用于接收用户传输数据及展示数据。三层架构体现了软件设计中“高内聚低耦合”思想，即减少类内部对其它类的调用，提高各模块之间的独立性。

SSM框架集，共由Spring、SpringMVC及MyBatis组成，它们在三层架构中的功能不尽相同。其中，Spring用于管理整个程序中所有组件的生命周期行为，即整个程序中所有对象的创建、初始化、销毁，以及对象间关联的维护行为等；SpringMVC作为界面层的实现者，用于完成用户请求的转发及对用户的响应；MyBatis作为数据访问层的实现者，实现对数据库进行的增删改查基本操作。综上所述，在SSM框架集内，页面向控制器发送请求，随后控制器调用业务层来处理逻辑，逻辑层向持久层发送请求，随后持久层与数据库进行交互，后将结果返回给业务层，再由业务层将处理逻辑发送给控制器，控制器再调用视图展现数据。

根据本系统的具体需求，旨望本系统能达到维护的简易、方便性，可直接于网页中使用。经综合评价后，决定令本系统采用B/S构架并使用SSM框架集，B/S构架图、SSM框架集构架图分别如图3-1、图3-2所示。

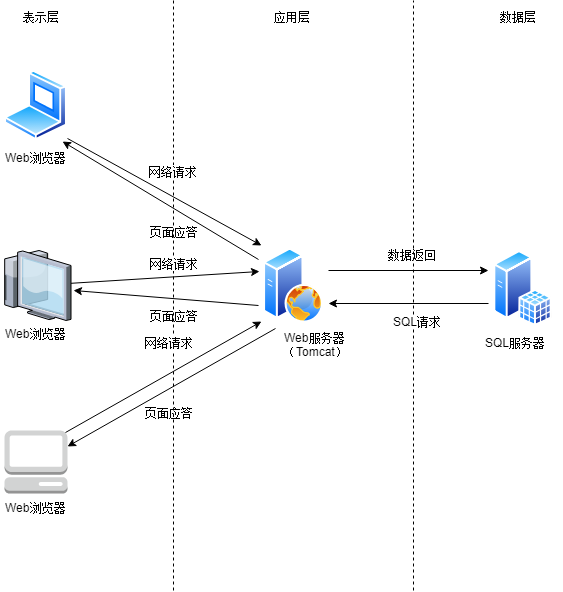


图3-1 B/S系统结构图

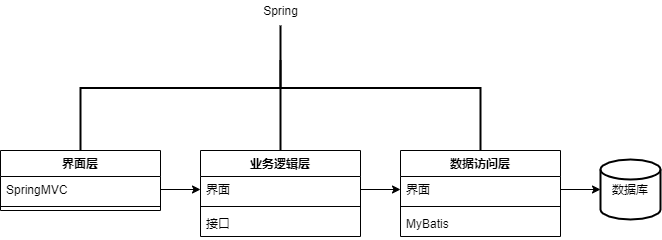


图3-2 SSM框架集构架图

根据需求分析的相关要求，由此得出本系统的技术架构设计及应用架构设计，具体内容如图3-3、图3-4所示。

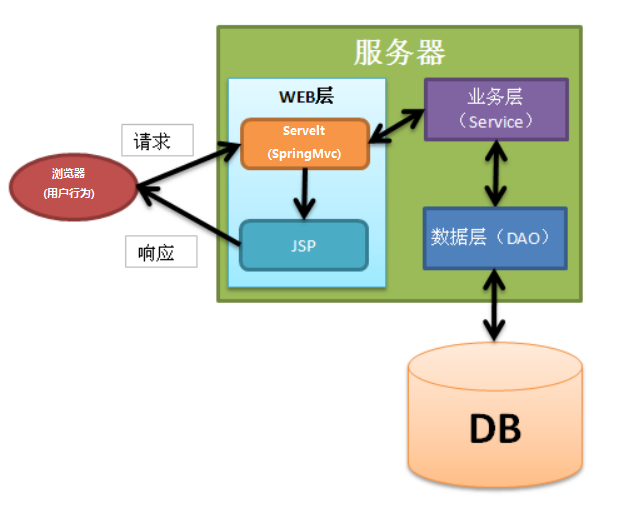


图3-3 技术架构设计展示简图

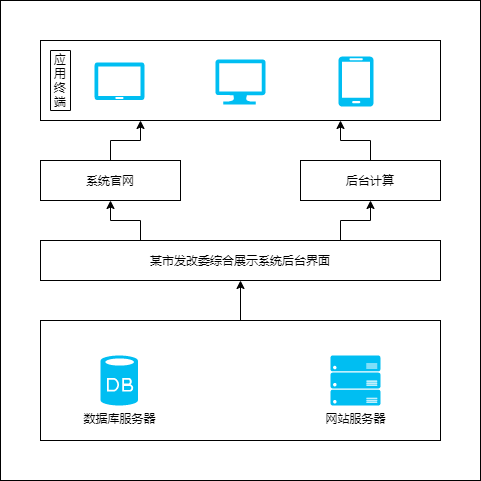


图3-4 应用架构设计展示图

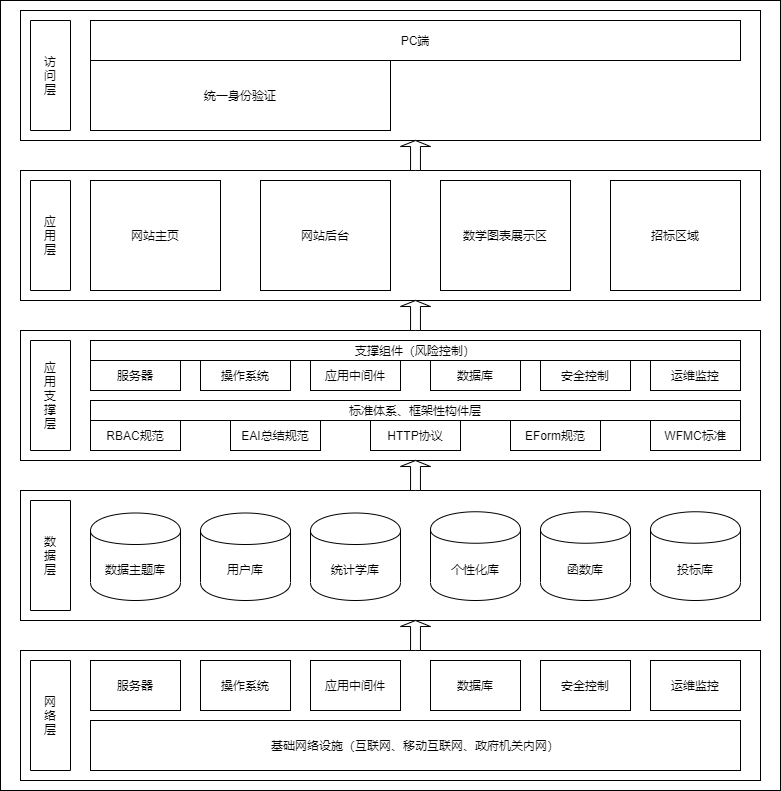


图3-5 技术架构设计展示全图

## 3.3 系统功能结构设计

本系统分为用户管理子系统、数据主题管理子系统、个性化定制展示子系统、统计分析结果子系统、可视化展现子系统。

用户管理子系统包括查看个人信息功能、修改个人信息功能，以及审核个人信息功能。

数据主题管理子系统包括添加项目功能、修改项目、删除项目、查询项目，以及审核项目信息功能。

个性化定制展示子系统包括添加个性化内容、审核、修改个性化内容、删除个性化内容以及查询个性化内容功能。

统计分析结果子系统函数修改、函数删除、同比计算、环比计算、立方体计算、趋势计算、数学模型查询及数学模型删除功能。

展示综合联动子系统包括项目查看、项目查询以及投标申请功能。

本系统整体结构图及总体流程图分别如图3-6及图3-7所示。

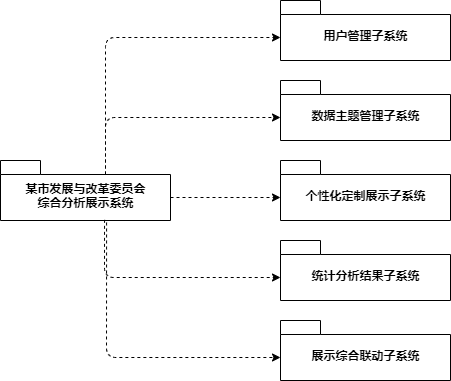


图3-6 系统总结构图

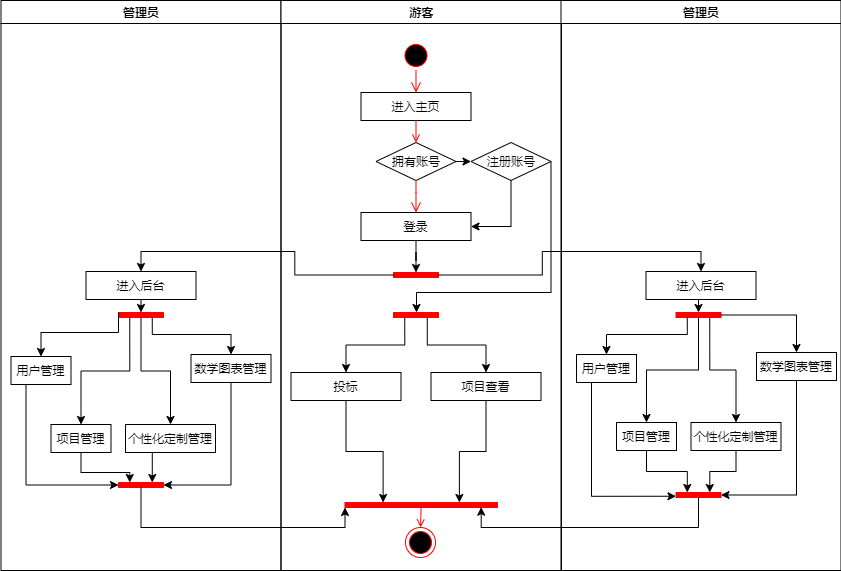


图3-7 系统总结构图

# 4 数据库设计

## 4.1 概念结构设计

### **4.1.1 设计思路**

所谓概念设计，即用来反映现实世界中的实体、属性和它们之间的关系等的原始数据形式，建立数据库的用户视图。

一般来说，采用E-R方法来进行数据库概念设计，其中E和R分别代表“Entity”和“Relationship”，即“实体”“联系”，其大致可按以下步骤进行：首先设计局部E-R模式，其次把各局部E-R模式综合成一个全局的E-R模式，最后对全局E-R模式进行优化，得到最终的E-R模式，即概念模式。而所得的E-R模型具有体现“联系”功能，用于刻画实体之间的关联。一种完整的方式是对局部结构中任意两个实体类型，依据需求分析的结果，考察局部结构中任意两个实体类型之间是否存在联系。若有联系，进一步确定是1:1、1:N、N:N的关系。还要考察一个实体类型内部是否存在联系，两个实体类型之间是否存在联系，多个实体类型之间是否存在联系等。

本系统实体根据使用本系统的用户及用户主要操作对象来决定。目前的实体包括可使用本系统的普通用户、管理员及游客，以及本系统主要操作对象项目、个性化内容、数学图表、统计学函数。其中使用本系统的实体可对剩余实体进行操作。

### **4.1.2 E-R图**

本系统总实体关系图（E-R图）如图4-1所示。

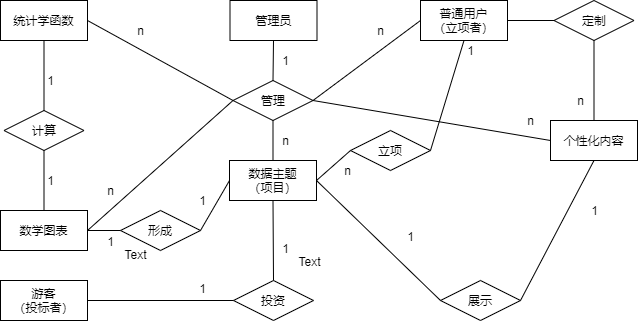


图4-1：系统总实体关系图

## 4.2 逻辑结构设计

### **4.2.1 设计思路**

根据需求分析及招标文件要求，可知本系统实体关系，如下所示。

1. 个性化实体图

实体：个性化；

属性：项目编号、对应项目、项目介绍、音频、展示图片；

其实体关系图如图4-2所示。

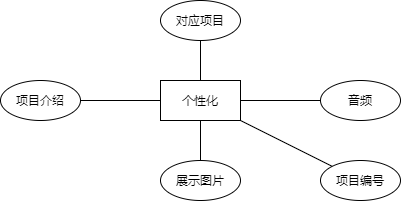


图4-2：个性化实体关系图

1. 函数实体图

实体：函数；

属性：函数编号、函数名、函数类型、表达式；

其实体关系图如图4-3所示。

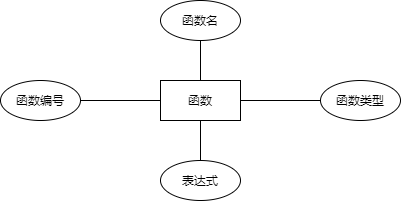


图4-3：函数实体关系图

1. 数据主题实体图

实体：数据主题；

属性：项目编号、对口单位、所需资金、项目名、施工人数、立项人电话、项目类型、立项人、项目简介；

其实体关系图如图4-4所示。

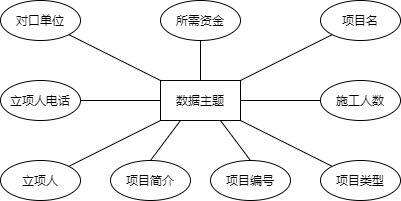


图4-4：数据主题实体关系图

1. 统计学实体图

实体：统计学；

属性：图表编号、对应项目、图表名、图表类型、图标内容；

其实体关系图如图4-5所示。

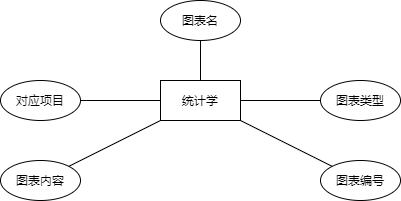


图4-5：统计学实体关系图

1. 投标实体图

实体：投标；

属性：项目编号、投标项目、投标人姓名、所属单位、联系地址、联系电话、负责人姓名、负责人电话；

其实体关系图如图4-6所示。

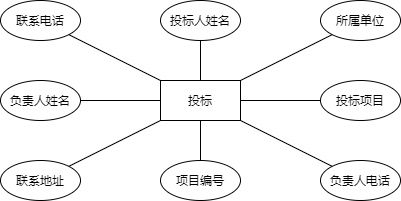


图4-6：投标实体关系图

1. 用户实体图

实体：用户；

属性：用户编号、用户名、联系电话、姓名、隶属单位、联系邮箱、权限、密码；

其实体关系图如图4-7所示。

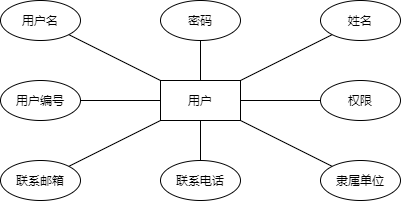


图4-7：用户实体关系图

### **4.2.2 逻辑模型**

用户表：（用户编号、用户名、联系电话、姓名、隶属单位、联系邮箱、权限、密码）；

数据主题：（项目编号、对口单位、所需资金、项目名、施工人数、立项人电话、项目类型、立项人、项目简介）；

个性化：（项目编号、对应项目、项目介绍、音频、展示图片）；

统计学：（图表编号、对应项目、图表名、图表类型、图标内容）；

函数：（函数编号、函数名、函数类型、表达式）；

投标：（项目编号、投标项目、投标人姓名、所属单位、联系地址、联系电话、负责人姓名、负责人电话）。

其中带下划线字体为该表的主键。

### **4.2.3 逻辑表格设计**

1. 用户表

用户表的逻辑表格设计如表4-1所示。

表4-1 用户表逻辑表格设计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | uId | 整型 | 10 | 是 | 用户编号 |
| 2 | uName | 字符串 | 20 | 否 | 用户名 |
| 3 | rName | 字符串 | 20 | 否 | 姓名 |
| 4 | rUnit | 字符串 | 20 | 否 | 隶属单位 |
| 5 | rPhone | 整型 | 11 | 否 | 联系电话 |
| 6 | email | 字符串 | 50 | 否 | 联系邮箱 |
| 7 | privilege | 字符串 | 4 | 否 | 权限 |
| 8 | password | 字符串 | 20 | 否 | 登录密码 |

1. 数据主题表

数据主题表的逻辑表格设计如表4-2所示。

表4-2 数据主题逻辑表格设计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | pId | 整型 | 10 | 是 | 项目编号 |
| 2 | pName | 字符串 | 20 | 否 | 项目名 |
| 3 | pType | 字符串 | 20 | 否 | 项目类型 |
| 4 | rName | 字符串 | 20 | 否 | 立项人 |
| 5 | rUnit | 字符串 | 20 | 否 | 对口单位 |
| 6 | rPhone | 整型 | 11 | 否 | 立项人电话 |
| 7 | funds | 浮点型 | 20 | 否 | 所需资金 |
| 8 | construction | 整型 | 4 | 否 | 施工人数 |
| 9 | introduction | 字符串 | 100 | 否 | 项目简介 |

1. 个性化表

个性化表的逻辑表格设计如表4-3所示。

表4-3 个性化逻辑表格设计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | pId | 整型 | 10 | 是 | 项目编号 |

续表4-3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | pName | 字符串 | 20 | 否 | 对应项目 |
| 3 | description | 字符串 | 20 | 否 | 项目介绍 |
| 4 | photo | 字符串 | 255 | 否 | 展示图片 |
| 5 | audio | 字符串 | 255 | 否 | 音频 |

1. 函数表

函数表的逻辑表格设计如表4-4所示。

表4-4 函数逻辑表格设计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | fId | 整型 | 10 | 是 | 函数编号 |
| 2 | fName | 字符串 | 20 | 否 | 函数名 |
| 3 | fType | 字符串 | 20 | 否 | 函数类型 |
| 4 | expression | 字符串 | 100 | 否 | 表达式 |

1. 统计学表

统计学表的逻辑表格设计如表4-5所示。

表4-5 统计学逻辑表格设计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | dId | 整型 | 10 | 是 | 图表编号 |
| 2 | dName | 字符串 | 20 | 否 | 图表名 |
| 3 | pName | 字符串 | 20 | 否 | 对应项目 |
| 4 | dType | 字符串 | 20 | 否 | 图表类型 |
| 5 | diagram | 字符串 | 255 | 否 | 图表内容 |

1. 投标表

投标表的逻辑表格设计如表4-6所示。

表4-6 投标逻辑表格设计表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 说明 |
| 1 | uId | 整型 | 10 | 是 | 项目编号 |
| 2 | uName | 字符串 | 20 | 否 | 投标项目 |
| 3 | iName | 字符串 | 20 | 否 | 投标人姓名 |
| 4 | iUnit | 字符串 | 20 | 否 | 所属单位 |
| 5 | iPhone | 整型 | 11 | 否 | 联系电话 |
| 6 | iAddress | 字符串 | 50 | 否 | 联系地址 |
| 7 | rName | 字符串 | 20 | 否 | 负责人姓名 |
| 8 | rPhone | 整型 | 11 | 否 | 负责人电话 |

## 4.3 物理结构设计

### **4.3.1 存取方式**

针对数据库系统中数据的结构，本系统采用mysql作为存储数据库，并且采用了索引顺序存储、数据分离和缓存优化技术。系统的数据库表存储采取顺序存储并对行字段建立对应索引，对经常查询的表进行数据分离，加快系统的查询速度，并且对于系统中的一些不经常更新的查询结果，将其存放在缓存数据库中，节省了重复查询的时间开销。

### **4.3.2 存储结构**

1. 用户表

用户表的物理存储结构如表4-7所示。

表4-7 用户表物理存储结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 字段长度 | 是否主键 | 字段含义 |
| 1 | uId | int | 10 | 是 | 用户编号 |
| 2 | uName | varchar | 20 | 否 | 用户名 |
| 3 | rName | varchar | 20 | 否 | 姓名 |
| 4 | rUnit | varchar | 20 | 否 | 隶属单位 |
| 5 | rPhone | int | 11 | 否 | 联系电话 |
| 6 | email | varchar | 50 | 否 | 联系邮箱 |
| 7 | privilege | bool | 4 | 否 | 权限 |

1. 数据主题表

数据主题表的物理存储结构如表4-8所示。

表4-8 数据主题表物理存储结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 类型 | 字段长度 | 是否主键 | 字段含义 |
| 1 | pId | int | 10 | 是 | 项目编号 |
| 2 | pName | varchar | 20 | 否 | 项目名 |
| 3 | pType | varchar | 20 | 否 | 项目类型 |
| 4 | rName | varchar | 20 | 否 | 立项人 |
| 5 | rUnit | varchar | 20 | 否 | 对口单位 |
| 6 | rPhone | int | 11 | 否 | 立项人电话 |
| 7 | funds | float | 20 | 否 | 所需资金 |
| 8 | construction | int | 4 | 否 | 施工人数 |
| 9 | introduction | varchar | 100 | 否 | 项目简介 |

1. 个性化表

个性化表的物理存储结构如表4-9所示。

表4-9 个性化表物理存储结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 字段含义 |
| 1 | pId | int | 10 | 是 | 项目编号 |
| 2 | pName | varchar | 20 | 否 | 对应项目 |
| 3 | description | varchar | 20 | 否 | 项目介绍 |
| 4 | photo | varchar | 255 | 否 | 展示图片 |
| 5 | audio | varchar | 255 | 否 | 音频 |

1. 函数表

函数表的物理存储结构如表4-10所示。

表4-10 函数表物理存储结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 字段含义 |
| 1 | fId | int | 10 | 是 | 函数编号 |
| 2 | fName | varchar | 20 | 否 | 函数名 |
| 3 | fType | varchar | 20 | 否 | 函数类型 |
| 4 | expression | varchar | 100 | 否 | 表达式 |

1. 统计学表

统计学表的物理存储结构如表4-11所示。

表4-11 统计学表物理存储结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 字段含义 |
| 1 | dId | int | 10 | 是 | 图表编号 |
| 2 | dName | varchar | 20 | 否 | 图表名 |
| 3 | pName | varchar | 20 | 否 | 对应项目 |
| 4 | dType | varchar | 20 | 否 | 图表类型 |
| 5 | diagram | varchar | 255 | 否 | 图表内容 |

1. 投标表

投标表的物理存储结构如表4-12所示。

表4-12 投标表物理存储结构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 数据类型 | 字段长度 | 是否主键 | 字段含义 |
| 1 | uId | int | 10 | 是 | 项目编号 |
| 2 | uName | varchar | 20 | 否 | 投标项目 |
| 3 | iName | varchar | 20 | 否 | 投标人姓名 |
| 4 | iUnit | varchar | 20 | 否 | 所属单位 |
| 5 | iPhone | int | 11 | 否 | 联系电话 |
| 6 | iAddress | varchar | 50 | 否 | 联系地址 |
| 7 | rName | varchar | 20 | 否 | 负责人姓名 |

# 5 界面设计

## 5.1 界面关系图

本系统界面关系图如图5-1所示。

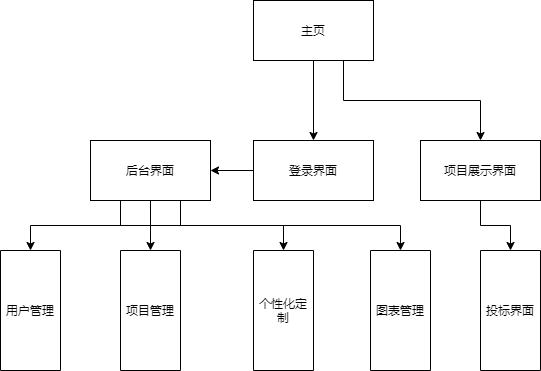


图5-1 系统功能结构图

用户在进入主页后，若拥有账号，即可使用登录界面进行登录，亦可直接在主页中的登录模块进行登录。若无账号，则可以进入注册界面进行注册。用户登录后即可进入后台，可在用户管理界面管理其个人信息，在项目管理界面既可提交项目，又可管理自己已提交的项目，在个性化定制界面针对已提交项目进行个性化定制操作。若用户为管理员，则可对以上内容（用户、项目、个性化定制）进行管理，此外还可进入图表管理界面管理由项目具体数据生成的图表。游客可在主页浏览当前已上传的项目，若对项目感兴趣，则可进入特定项目的界面，填写投标信息。

**5.2 界面设计成果**

### **5.2.1 主界面**



图5-2：系统首页

如图5-2所示，此处展示的是本系统的首页。主页处会展示最新的新闻动态，旁边有登录界面，若无账号则可以使用上方的红色“注册”按钮进行注册。上方快捷栏可到达新闻动态界面、项目展示界面、图表展示界面、投标界面，以及搜索界面。

### **5.2.2 子界面**

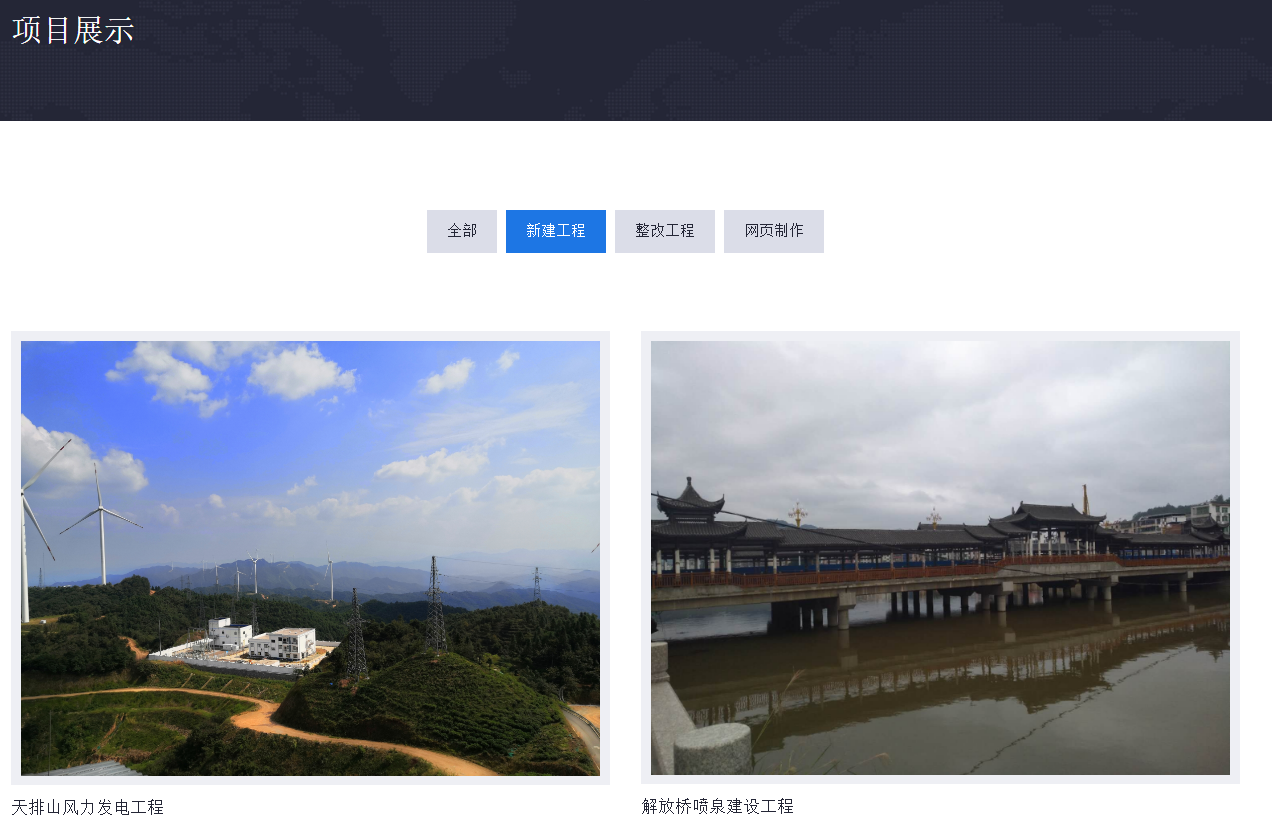


图5-3：项目展示界面

如图5-3所示，此处展示的是项目展示界面。已上传的项目会根据给出的分类“新建工程”“整改工程”“网页制作”三类进行分类展示，亦可通过“全部”查看当前系统内所有的项目。



图5-4：图表展示界面

如图5-4所示，此处展示的是图表展示界面。该界面会通过当前已上传项目的部分属性中的数据生成特定的扇形统计图。



图5-5：最近新闻界面

如图5-5所示，此处展示的是最近新闻界面。此界面会展示当前发改委发布的最新新闻及相关政策材料。

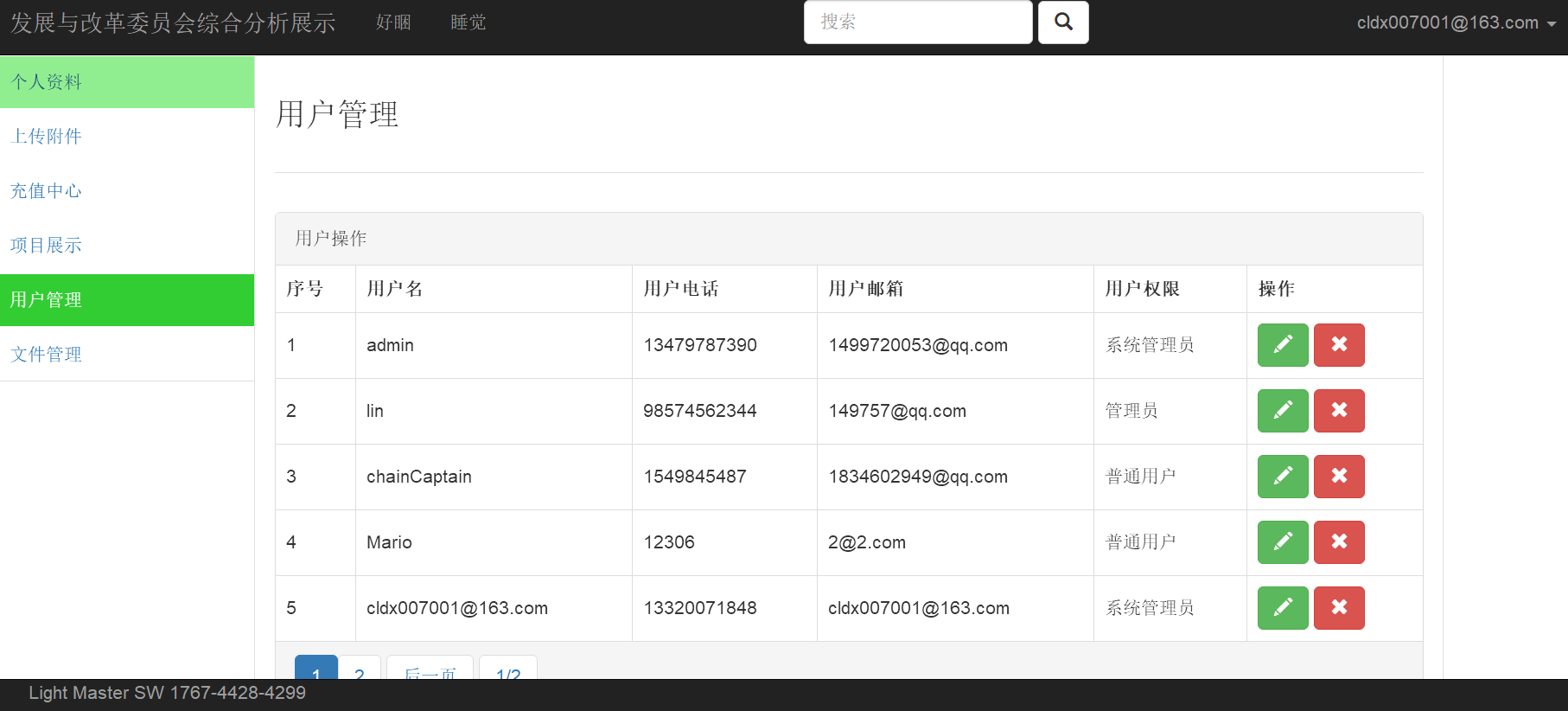


图5-5：用户管理界面

由于后台界面风格及功能类似，此处只演示后台用户管理界面，如图5-5所示。用户可通过此页面更改自身个人信息，若存在问题亦可删除之。

**6 详细设计**

## 6.1 系统主要功能模块介绍

本系统包括用户管理子系统、数据主题管理子系统，统计分析结果子系统，个性化定制展示子系统，展示综合联动交互子系统。

用户管理：包括用户信息管理模块、用户审核模块。

数据主题管理：包括数据主题导入模块、数据主题维护模块、数据主题审核模块。

统计分析结果：包括维度与统计值配置管理模块、立方体计算与结果缓存模块、同比/环比/趋势计算模块、可视化展现模块。

个性化定制展示：包括个性化内容导入模块、个性化内容整理模块。

展示综合联动：包括关键字搜索模块、结果展示模块、投标模块。

本系统总功能结构图如图6-1所示：

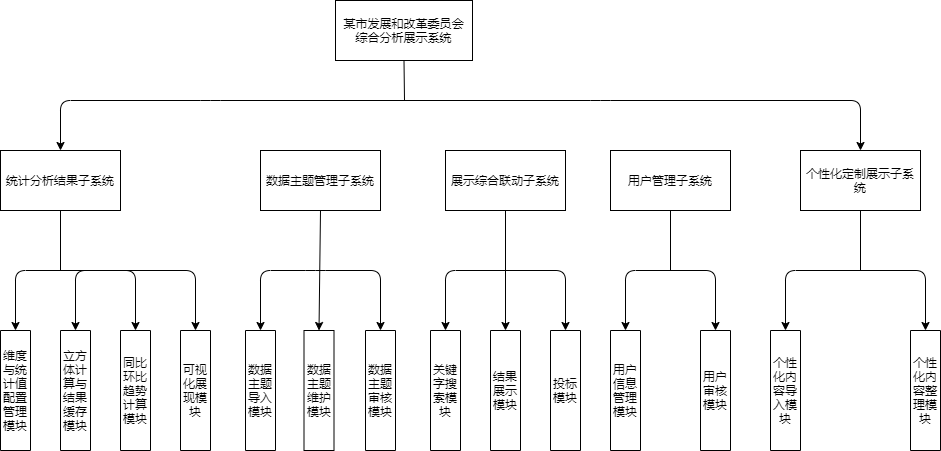


图6-1 系统功能结构图

## 6.2 用户管理模块设计

1. 功能结构设计

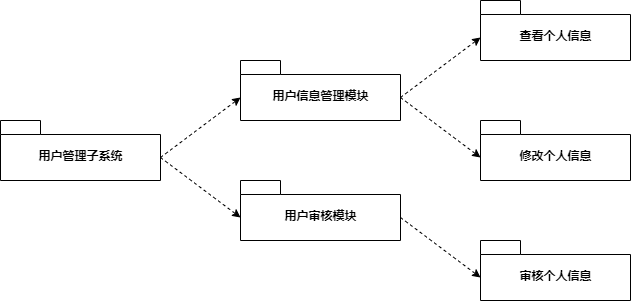


图6-2 用户管理子系统结构图

用户管理子系统包括用户信息管理模块和用户审核模块，其结构图如图6-2所示。

用户管理模块包括查看个人信息功能和修改个人信息功能，用户可在该模块中查看其目前已有的个人信息，若个人信息存在问题，亦可在此处进行修改，修改提交并待审核通过之后将更新，更新内容可在该模块内再次查看。

用户审核模块为管理员所使用的模块，其具有审核个人信息功能，管理员在收到普通用户于用户信息管理模块中提交的个人信息修改请求后，将对修改的个人信息具体内容进行审核，当审核无误后将通过审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。

1. 类图设计

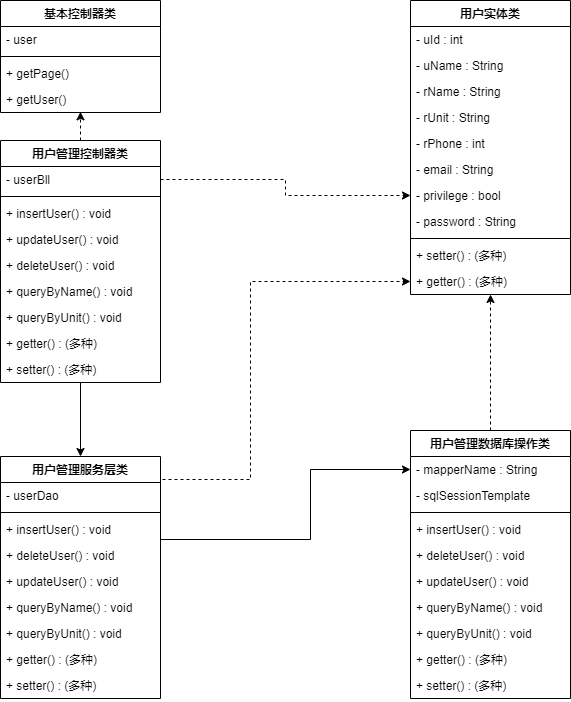
**

图6-3 用户管理类图

表6-1 用户管理功能描述表

|  |  |  |
| --- | --- | --- |
| **用户管理数据库操作类** | | |
| 类的描述：负责用户管理相关操作的数据库操作层函数封装 | | |
| 操作描述 | insertUser | 插入用户数据 |
| deleteUser | 删除用户数据 |
| updateUser | 更新用户数据 |
| queryByName | 根据用户名或姓名查找 |
| queryByUnit | 根据所属单位查找 |

续表6-1

|  |  |  |
| --- | --- | --- |
|  | getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **用户管理服务层类** | | |
| 类的描述：负责用户管理相关操作的服务层方法封装 | | |
| 操作描述 | insertUser | 插入用户数据 |
| deleteUser | 删除用户数据 |
| updateUser | 更新用户数据 |
| queryByName | 根据用户名或姓名查找 |
| queryByUnit | 根据所属单位查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **登录注册控制器类** | | |
| 类的描述：负责用户管理相关请求的处理和转发 | | |
| 操作描述 | insertUser | 插入用户数据 |
| deleteUser | 删除用户数据 |
| updateUser | 更新用户数据 |
| queryByName | 根据用户名或姓名查找 |
| queryByUnit | 根据所属单位查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **用户实体类** | | |
| 类的描述：用户的实体对象类 | | |
| 属性描述 | uId | 用户编号 |
| uName | 用户名 |
| rName | 用户真实姓名 |
| password | 密码 |
| rUnit | 所属单位 |
| privilege | 用户权限 |
| rPhone | 联系电话 |
| email | 联系邮箱 |
| 操作描述 | getter | 对用户属性的getter方法 |
| setter | 对用户属性的setter方法 |

1. 时序图设计

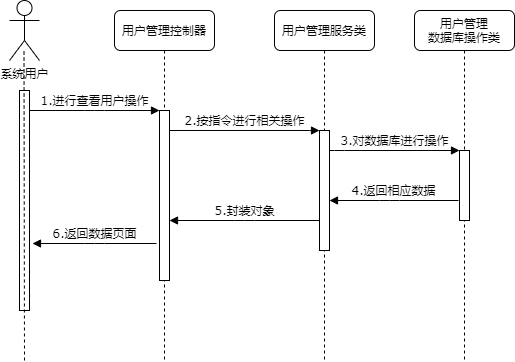


图6-4 用户管理时序图

1. 核心处理流程设计

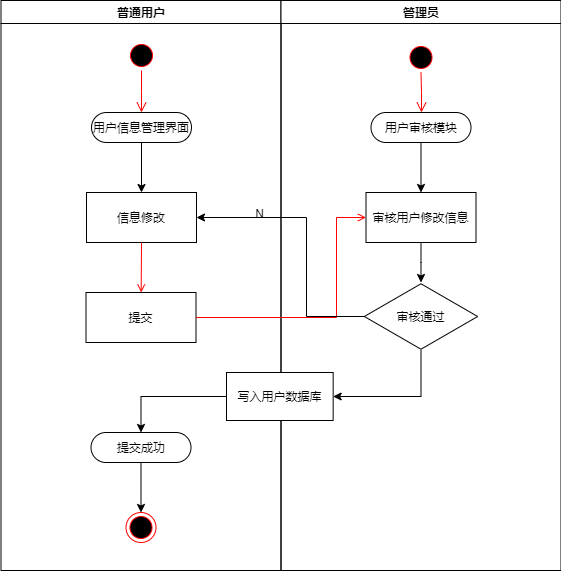
**

图6-5 用户管理子系统活动图

### **6.3 数据主题管理模块设计**

1. 功能结构设计

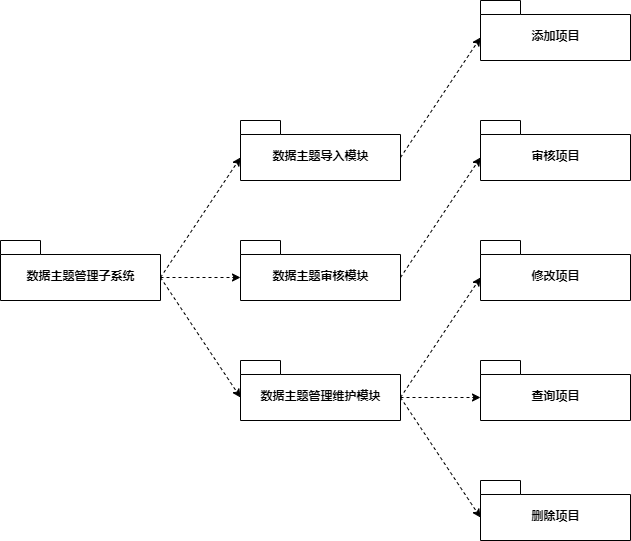


图6-6 数据主题管理子系统结构图

数据主题管理子系统包括数据主题导入模块、数据主题维护模块，以及数据主题审核模块，其结构图如图6-6所示。

数据主题导入模块包括添加项目功能。普通用户若需要立项，则可在该模块中提交自身提供的项目信息，待提交并待审核通过之后将成功上传，该上传内容可在数据主题维护模块内查看。

数据主题维护模块包括修改项目、删除项目以及查询项目功能。用户可在该模块中查看当前自身已提交的所有项目，若未找到，则可用模糊查询功能搜索之。若当前提交的项目存在一定的问题，需要修改其中一部分信息，则可进入特定项目的维护界面，对该项目进行信息维护工作，待提交并待审核通过之后将成功上传，该上传内容可在数据主题维护模块内查看。若当前存在失去时效性的项目，需要将其删除，则可在维护界面提交删除请求，待管理员审核通过后将成功删除。

数据主题审核模块为管理员所使用的模块，其具有审核项目信息功能，管理员在收到普通用户于数据主题导入模块及数据主题维护模块中提交的项目信息上传、修改请求后，将对上述符合条件的项目具体内容进行审核，当审核无误后将通过审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。此外，管理员在收到普通用户提交的项目删除请求后，将对该项目时效性等内容进行审核，审核无误后将通过删除审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。

1. 类图设计

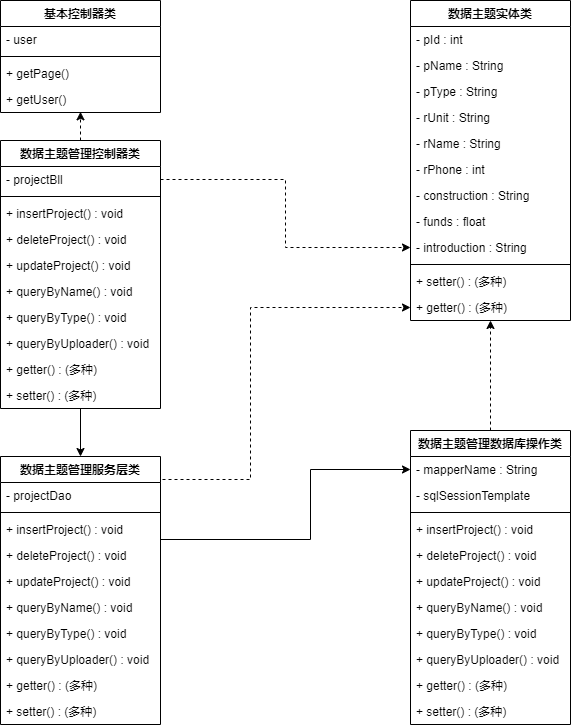


图6-7 数据主题类图

表6-2 数据主题管理功能描述表

|  |  |  |
| --- | --- | --- |
| **数据主题管理数据库操作类** | | |
| 类的描述：负责数据主题相关操作的数据库操作层函数封装 | | |
| 操作描述 | insertProject | 插入数据主题数据 |
| deleteProject | 删除数据主题数据 |
| updateProject | 更新数据主题数据 |
| queryByName | 根据项目名称查找 |
| queryByType | 根据所属类型查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| queryByUploader | 根据立项者查找 |
| **数据主题管理服务层类** | | |
| 类的描述：负责数据主题管理相关操作的服务层方法封装 | | |
| 操作描述 | insertProject | 插入数据主题数据 |
| deleteProject | 删除数据主题数据 |
| updateProject | 更新数据主题数据 |
| queryByName | 根据项目名称查找 |
| queryByType | 根据所属类型查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| queryByUploader | 根据立项者查找 |
| **数据主题管理控制器类** | | |
| 类的描述：负责数据主题管理相关请求的处理和转发 | | |
| 操作描述 | insertProject | 插入数据主题数据 |
| deleteProject | 删除数据主题数据 |
| updateProject | 更新数据主题数据 |
| queryByName | 根据项目名称查找 |
| queryByType | 根据所属类型查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| queryByUploader | 根据立项者查找 |
| **数据主题实体类** | | |
| 类的描述：数据主题的实体对象类 | | |
| 属性描述 | pId | 项目编号 |
| pName | 项目名 |
| pType | 项目类型 |
| rName | 立项人 |

续表6-2

|  |  |  |
| --- | --- | --- |
|  | rUnit | 对口单位 |
| rPhone | 立项人电话 |
| funds | 所需资金 |
| construction | 施工人数 |
| introduction | 项目简介 |
| 操作描述 | getter | 对数据主题属性的getter方法 |
| setter | 对数据主题属性的setter方法 |

1. 时序图设计

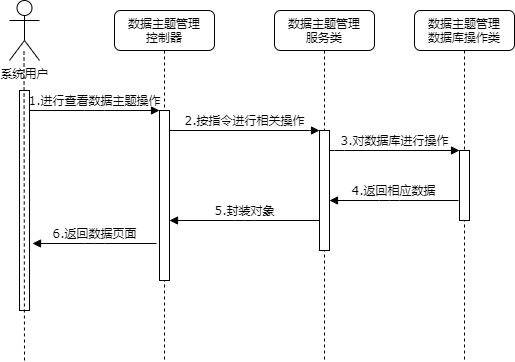
****

图6-8 数据主题管理时序图

1. 核心处理流程设计

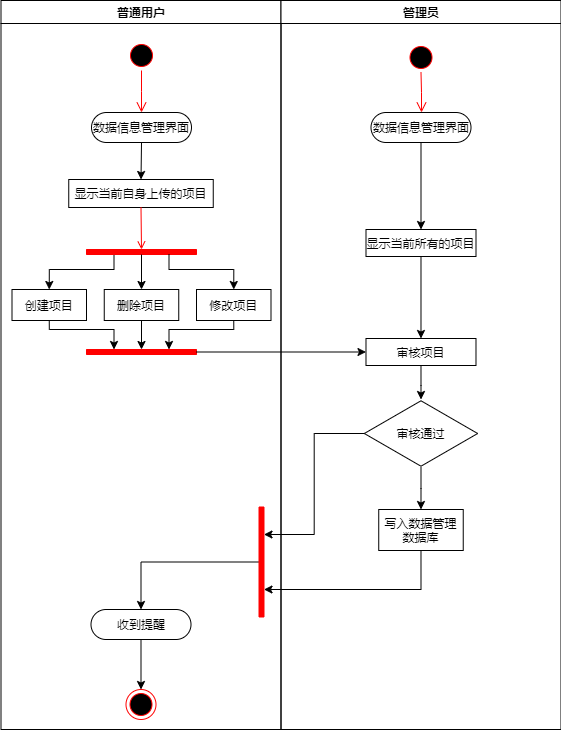


图6-9 数据主题管理子系统活动图

### **6.4 个性化定制模块设计**

1. 功能结构设计

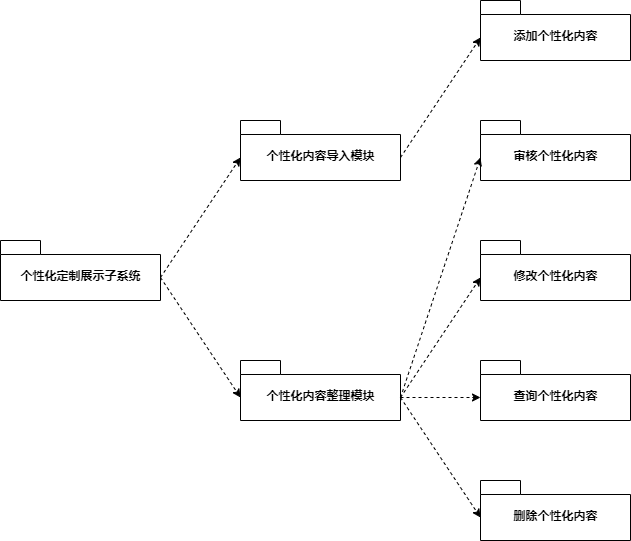


图6-10 个性化定制展示子系统结构图

个性化定制展示子系统包括个性化内容导入模块以及个性化内容整理模块，其结构图如图6-10所示。

个性化内容导入模块包括添加个性化内容功能。普通用户若需要在自身已提交的项目为基础之上进行个性化包装，则可在该项目对应的该模块中提交自身提供的个性化内容，待提交并待审核通过之后将成功上传，该上传内容可在个性化内容整理模块内查看。

个性化内容整理模块包括审核、修改个性化内容、删除个性化内容以及查询个性化内容功能，与数据主题维护模块功能大体相当。普通用户可在该模块中查看当前自身已提交的所有项目，若未找到，则可用模糊查询功能搜索之。若需要对当前特定项目进行个性化内容修改，则可进入特定项目的个性化定制界面，对该项目进行个性化定制操作，待提交并待审核通过之后将成功上传，该上传内容可在个性化内容整理模块内查看。若需要删除某个个性化内容，则可在整理界面提交删除请求，待管理员审核通过后将成功删除。若管理员进入该模块，则会在其页面中显示所有已上传个性化内容的项目，其具有审核个性化内容功能，管理员在收到普通用户于个性化内容导入模块及个性化内容整理模块中提交的项目信息上传、修改请求后，将对上述符合条件的项目具体内容进行审核，当审核无误后将通过审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。此外，管理员在收到普通用户提交的个性化内容删除请求后，将在审核无误后将通过删除审核，提交数据，并对修改者发出修改成功提醒，若审核未通过，则会对修改者发出修改失败提醒。

1. 类图设计

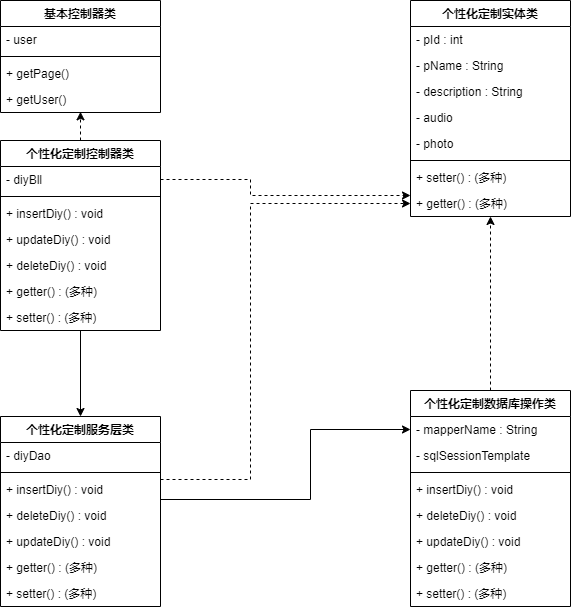


图6-11 个性化定制类图

表6-4 个性化定制展示功能描述表

|  |  |  |
| --- | --- | --- |
| **个性化定制数据库操作类** | | |
| 类的描述：负责个性化定制相关操作的数据库操作层函数封装 | | |
| 操作描述 | insertDiy | 添加个性化定制内容 |
| deleteDiy | 删除个性化定制内容 |

续表6-4

|  |  |  |
| --- | --- | --- |
|  | updateDiy | 更新个性化定制内容 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **个性化定制服务层类** | | |
| 类的描述：负责个性化定制相关操作的服务层方法封装 | | |
| 操作描述 | insertDiy | 添加个性化定制内容 |
| deleteDiy | 删除个性化定制内容 |
| updateDiy | 更新个性化定制内容 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **个性化定制控制器类** | | |
| 类的描述：负责个性化定制相关请求的处理和转发 | | |
| 操作描述 | insertDiy | 添加个性化定制内容 |
| deleteDiy | 删除个性化定制内容 |
| updateDiy | 更新个性化定制内容 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **个性化实体类** | | |
| 类的描述：个性化内容的实体对象类 | | |
| 属性描述 | pId | 项目编号 |
| pName | 对应项目 |
| description | 项目介绍 |
| photo | 展示图片 |
| audio | 音频 |
| 操作描述 | getter | 对个性化属性的getter方法 |
| setter | 对个性化属性的setter方法 |

1. 时序图设计

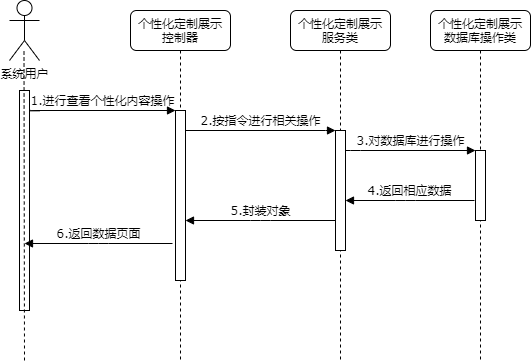


图6-12 个性化定制时序图

1. 核心处理流程设计

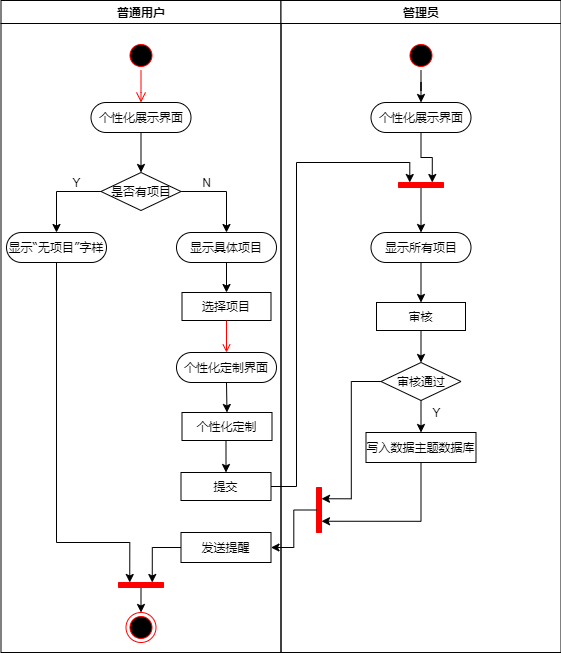


图6-13 个性化定制子系统活动图

### **6.5 统计分析结果模块设计**

1. 功能结构设计

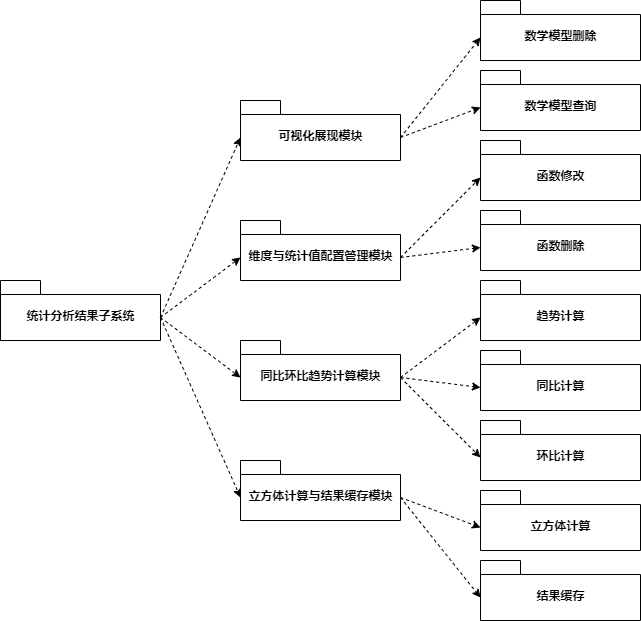


图6-14 统计分析结果子系统结构图

统计分析结果子系统包括维度与统计值配置管理模块、同比环比趋势计算模块、立方体计算与结果缓存模块以及可视化展现模块，其结构图如图6-14所示。

维度与与统计值配置管理模块拥有函数修改及函数删除功能。管理员可在此模块下对本系统所涉及到的函数进行管理，若某个函数表达式有误，则可以进入特定函数的修改界面进行修改操作。若某个函数表达式失去时效性，则可以删除该表达式。

立方体计算与结果缓存模块包括立方体计算及结果缓存功能。管理员在设置好新的函数之后，或普通用户提交新的项目之后，将由该模块对现有数据主题数值进行计算处理，并生成特定的立方体统计图，其中间结果将缓存至缓存区域中。

同比环比趋势计算模块，顾名思义，包括同比计算、环比计算以及趋势计算三个功能。管理员在设置好新的函数之后，或普通用户提交新的项目之后，将由该模块对现有数据主题数值进行计算处理，并生成特定的同比、环比、趋势数据。

可视化展现模块拥有数学模型查询及数学模型删除功能。经上述两个模块处理后所生成的数学图表将在该界面显示，并由管理员审核，若审核无误，则可令该数学图表发布至主页上。

1. 类图设计

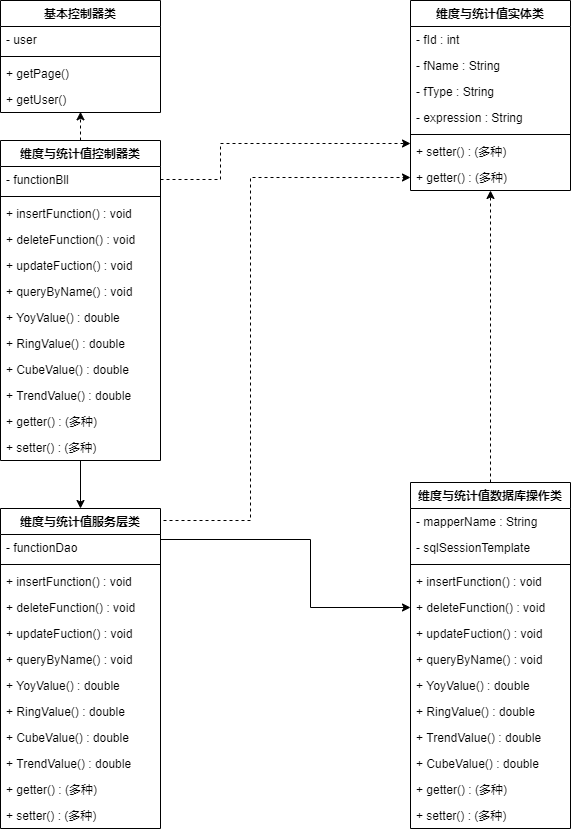


图6-15 函数管理类图

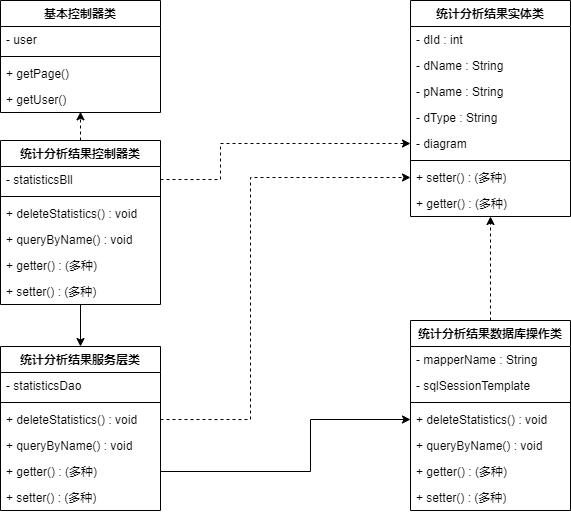


图6-16 统计分析结果类图

表6-5 统计分析结果功能描述表

|  |  |  |
| --- | --- | --- |
| **维度与统计值数据库操作类** | | |
| 类的描述：负责函数管理相关操作的数据库操作层函数封装 | | |
| 操作描述 | insertFunction | 插入函数 |
| deleteFunction | 删除函数 |
| updateFunction | 更新函数表达式 |
| queryByName | 根据函数名进行查找 |
| YoyValue | 同比计算 |
| RingValue | 环比计算 |
| TrendValue | 趋势计算 |
| CubeValue | 立方体计算 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **维度与统计值服务层类** | | |
| 类的描述：负责函数管理相关操作的服务层方法封装 | | |
| 操作描述 | insertFunction | 插入函数 |
| deleteFunction | 删除函数 |

续表6-5

|  |  |  |
| --- | --- | --- |
|  | updateFunction | 更新函数表达式 |
| queryByName | 根据函数名进行查找 |
| YoyValue | 同比计算 |
| RingValue | 环比计算 |
| TrendValue | 趋势计算 |
| CubeValue | 立方体计算 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **维度与统计值控制器类** | | |
| 类的描述：负责函数管理相关请求的处理和转发 | | |
| 操作描述 | insertFunction | 插入函数 |
| deleteFunction | 删除函数 |
| updateFunction | 更新函数表达式 |
| queryByName | 根据函数名进行查找 |
| YoyValue | 同比计算 |
| RingValue | 环比计算 |
| TrendValue | 趋势计算 |
| CubeValue | 立方体计算 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **函数实体类** | | |
| 类的描述：函数的实体对象类 | | |
| 属性描述 | fId | 函数编号 |
| fName | 函数名 |
| fType | 函数类型 |
| expression | 表达式 |
| 操作描述 | getter | 对个性化属性的getter方法 |
| setter | 对个性化属性的setter方法 |
| **统计分析结果数据库操作类** | | |
| 类的描述：负责统计分析相关操作的数据库操作层函数封装 | | |
| 操作描述 | deleteStatistics | 删除数学图表 |
| queryByName | 根据数学图表名称查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **统计分析结果服务层类** | | |
| 类的描述：负责统计分析相关操作的服务层方法封装 | | |

续表6-5

|  |  |  |
| --- | --- | --- |
| 操作描述 | deleteStatistics | 删除数学图表 |
| queryByName | 根据数学图表名称查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **统计分析结果控制器类** | | |
| 类的描述：负责统计分析相关请求的处理和转发 | | |
| 操作描述 | deleteStatistics | 删除数学图表 |
| queryByName | 根据数学图表名称查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **统计学实体类** | | |
| 类的描述：统计学的实体对象类 | | |
| 属性描述 | dId | 图表编号 |
| dName | 图表名 |
| pName | 对应项目 |
| dType | 图表类型 |
| diagram | 图表内容 |
| 操作描述 | getter | 对个性化属性的getter方法 |
| setter | 对个性化属性的setter方法 |

1. 时序图设计

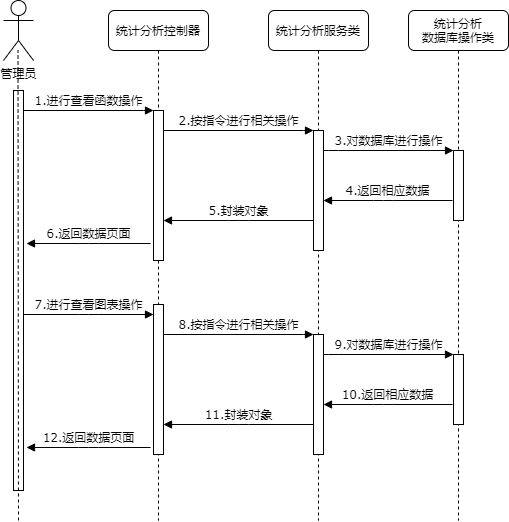


图6-17 统计分析结果时序图

1. 核心处理流程设计

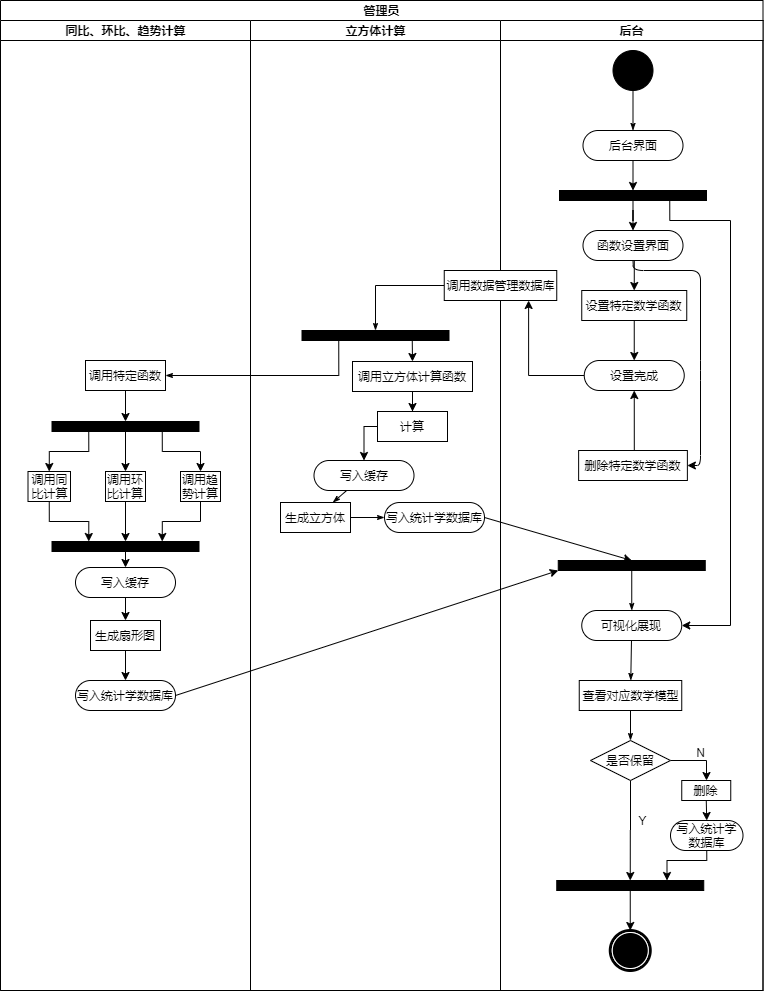


图6-18 统计分析结果子系统活动图

### **6.6 展示综合联动模块设计**

1. 功能结构设计

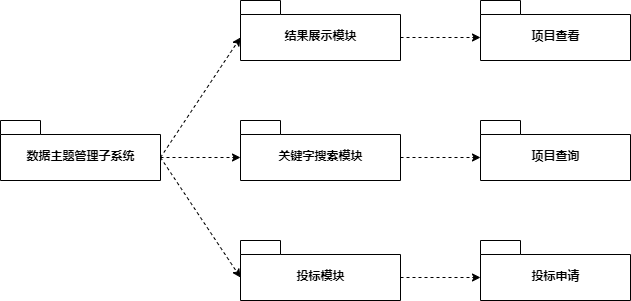


图6-19 展示综合联动子系统结构图

展示综合联动包括结果展示模块、关键字搜索模块，以及投标模块，其结构图如图6-19所示。

结果展示模块位于主页，拥有项目查看功能。用户可以在该模块查看特定项目的具体信息。

关键字搜索模块包括项目查询功能。若用户未在首页找到自己所需要的项目，则可通过该模块进行模糊查询，搜索自己所需要的项目，并查看之。

投标模块包括投标申请功能。若为游客权限的投标者找到了自己感兴趣的项目，需要投资，则需要在特定项目的投标界面中输入投标者自身的详细个人信息，并提交投标信息，待发改委工作人员审核并以电话形式反馈结果。

1. 类图设计

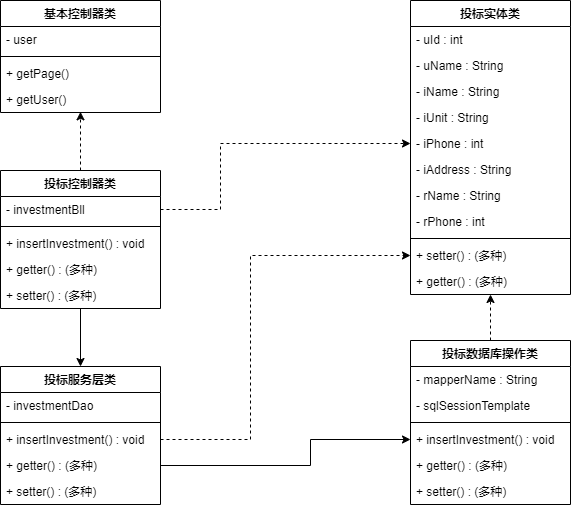


图6-20 投标类图

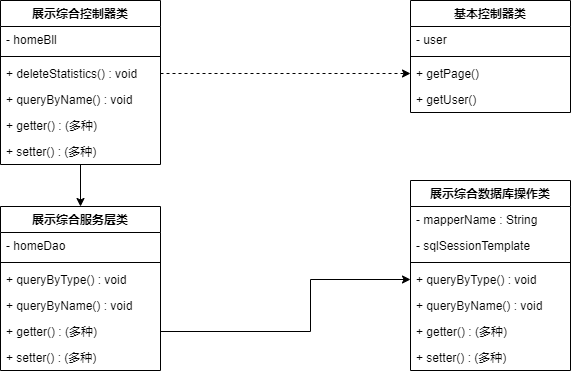


图6-21 展示综合类图

表6-6 展示综合联动功能描述表

|  |  |  |
| --- | --- | --- |
| **展示综合数据库操作类** | | |
| 类的描述：负责主页相关操作的数据库操作层函数封装 | | |
| 操作描述 | queryByType | 根据项目类型进行查找 |
| queryByName | 根据项目名进行查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **展示综合服务层类** | | |
| 类的描述：负责主页相关操作的服务层方法封装 | | |
| 操作描述 | queryByType | 根据项目类型进行查找 |
| queryByName | 根据项目名进行查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **个性化定制控制器类** | | |
| 类的描述：负责主页相关请求的处理和转发 | | |
| 操作描述 | queryByType | 根据项目类型进行查找 |
| queryByName | 根据项目名进行查找 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **投标实体类** | | |
| 类的描述：投标的实体对象类 | | |
| 属性描述 | uId | 项目编号 |
| uName | 投标项目 |
| iName | 投标人姓名 |
| iUnit | 所属单位 |
| iPhone | 联系电话 |
| iAddress | 联系地址 |
| rName | 负责人姓名 |
| 操作描述 | getter | 对个性化属性的getter方法 |
| setter | 对个性化属性的setter方法 |
| **投标数据库操作类** | | |
| 类的描述：负责投标相关操作的数据库操作层函数封装 | | |
| 操作描述 | insertInvestment | 添加投标申请 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |

**续**表6-6

|  |  |  |
| --- | --- | --- |
| **投标服务层类** | | |
| 类的描述：负责投标相关操作的服务层方法封装 | | |
| 操作描述 | insertInvestment | 添加投标申请 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |
| **个性化定制控制器类** | | |
| 类的描述：负责投标相关请求的处理和转发 | | |
| 操作描述 | insertInvestment | 添加投标申请 |
| getter | 对该属性的getter方法 |
| setter | 对该属性的setter方法 |

1. 顺序图设计

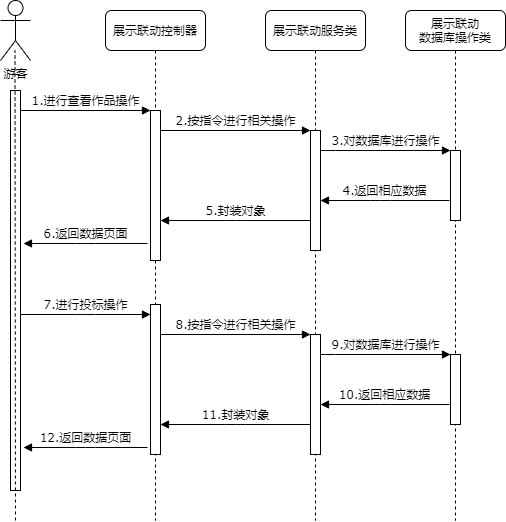


图6-22 展示综合联动时序图

1. 核心处理流程设计

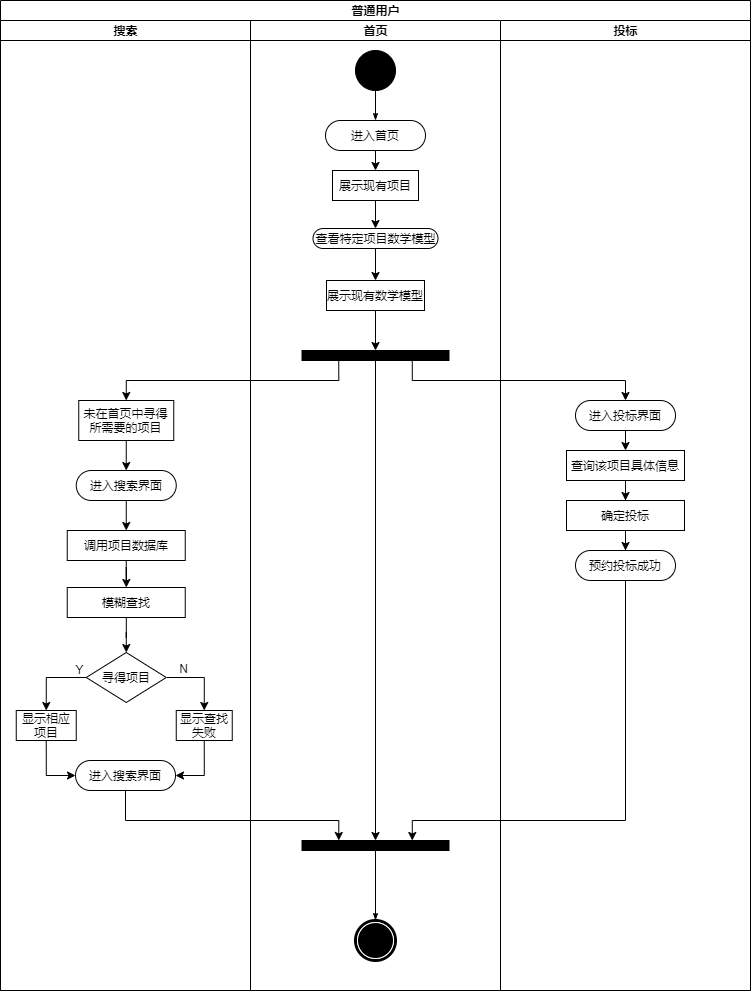


图6-23 展示综合联动子系统活动图

7 编码

## 7.1 代码实现与核心算法

本系统由SSM框架构架而成，故编码模式相当。

文件上传：用户在提交项目时所附有的文件上传功能，可供用户上传文件，其代码如下：

//修改文件名字

String uploadName=file.getOriginalFilename();

String saveName=learnFile.getId()+uploadName.substring(uploadName.lastIndexOf("."));

//设置保存名 和保存路径,文件类型

learnFile.setSave\_name(saveName);

learnFile.setSave\_path(filePath+"/"+userName+"/"+saveName);

learnFile.setFile\_type(uploadName.substring(uploadName.lastIndexOf(".")+1));

//文件上传

if(!file.isEmpty()){

File tempFile=new File(filePath+"/"+userName+"/"+saveName);

try {

//传输

file.transferTo(tempFile);

} catch (IOException e) {

e.printStackTrace();

System.err.println("传输失败");

return EnvironValues.FAIL;

}

}

//保存文件

fileDao.addFile(learnFile);

return EnvironValues.SUCCESS;

}

用户管理：以下为管理员权限的操作代码，其允许管理员指定用户类型，如将普通用户指定为管理员等：

var app = angular.module('myApp', []);

app.controller('userManageCtrl', function($scope,$http) {

$scope.isLoad=false;

//用户列表，当前页，页总数

$scope.userDate={'userList':[],'nowPage':1,'pageNum':null};

$scope.pageArray=[];

$scope.selectUser={};

//分页查询函数

$scope.queryData=queryUserList;

//删除用户

$scope.delUser=deleteUser;

//修改用户权限

$scope.mdfUser=modifyUser;

//信息初始,刷新之后就是第一页了

queryUserList($scope.userDate.nowPage);

$scope.isLoad=true;

//分页查询函数

function queryUserList(num) {

$http({

method: 'POST',

url: '/LearnRes/LR/userManageControl/queryUserList',

data:{'nowPage':num}

}).then(function success(response) {

// 请求成功执行代码

angular.copy(response.data,$scope.userDate);

$scope.pageArray=getPageArray($scope.userDate.nowPage,$scope.userDate.pageNum,5);

}, function error(response) {});

}

//删除用户

function deleteUser(delUser){

swal({title: "确定删除吗？",

text: "你将无法恢复该用户！",

type: "warning",

showCancelButton: true,

confirmButtonColor: "#DD6B55",

confirmButtonText: "确定删除！",

cancelButtonText: "取消删除！",

closeOnConfirm: false }, function(){

$http({

method: 'POST',

url: '/LearnRes/LR/userManageControl/deleteUser',

data:{'deleteId':delUser.user\_id}

}).then(function success(response) {

// 请求成功执行代码

if(response.data=='SUCCESS'){

swal({title:"删除！",

text:"用户已经删除",

type:"success"},function(){

//删除之后重新查询数据

queryUserList($scope.userDate.nowPage);

}

);

}

else if(response.data=='FAIL'){

swal("删除！", "用户删除失败。", "error");

}

}, function error(response) {});

});

}

//修改用户权限

function modifyUser(user){

angular.copy(user,$scope.selectUser);

$scope.selectUser.user\_power=String($scope.selectUser.user\_power);

$("#modifyPowerModal").modal('show');

}

//提交修改

$scope.commitModify=function() {

var power=parseInt($scope.selectUser.user\_power);

$http({

method: 'POST',

url: '/LearnRes/LR/userManageControl/updateUserPower',

data:{'user\_id':$scope.selectUser.user\_id,'user\_power':power

}

}).then(function success(response) {

// 请求成功执行代码

$("#modifyPowerModal").modal('hide');

if(response.data=='SUCCESS'){

swal({title:"修改！",

text:"修改权限成功",

type:"success"},function(){

//删除之后重新查询数据

queryUserList($scope.userDate.nowPage);

}

);

}

else if(response.data=='FAIL'){

swal("修改！", "修改权限失败。", "error");

}

}, function error(response) {

$("#modifyPowerModal").modal('hide');

});

}

});

app.filter("transformToPower",function () {

return function (text) {

var changed="";

switch (text.toString()) {

case '1':changed="系统管理员";break;

case '2':changed="管理员";break;

case '3':changed="普通用户";break;

default:changed="未知用户";break;

}

return changed;

}

});

项目管理：由于该模块功能实现代码与用户管理大同小异，此处不再展示复用率高的代码，以下为项目文件上传过程中系统识别格式的代码：

app.filter("transformToType",function () {

return function (text) {

var changed="";

switch (text) {

case 'png':changed="图片";break;

case 'jpg':changed="图片";break;

case 'mp4':changed="视频";break;

case 'avi':changed="视频";break;

case 'txt':changed="文本";break;

case 'docx':changed="文本";break;

default:changed="其他";break;

}

return changed;

}

});

XML：用于数据库处理的文件，此处以数据主题管理模块的部分对应代码为例：

<mapper namespace="lr.dao.UserMainDao">

<select id="getEntity" parameterType="java.lang.String" resultType="lr.entity.UserMain">

select \* from user\_main where user\_id=#{\_parameter}

</select>

<select id="getPassword" parameterType="java.lang.String" resultType="lr.entity.UserMain">

select user\_password,user\_id,user\_name from user\_main where user\_email=#{\_parameter}

</select>

<select id="getUserPower" parameterType="java.lang.String" resultType="int">

select user\_power from user\_main where user\_id=#{\_parameter}

</select>

<insert id="addUser" parameterType="lr.entity.UserMain">

insert into user\_main(user\_id,user\_name,user\_password,user\_power,user\_email,user\_phone)

values (#{user\_id},#{user\_name},#{user\_password},#{user\_power},#{user\_email},#{user\_phone})

</insert>

<select id="getUserList" resultType="lr.entity.UserMain">

select user\_id,user\_name,user\_power,user\_email,user\_phone from user\_main limit #{0},#{1}

</select>

<select id="check" parameterType="java.lang.String" resultType="lr.entity.UserMain">

select user\_id,user\_name,user\_power,user\_email,user\_phone from user\_main where user\_email=#{\_parameter}

</select>

<select id="getUserNumber" resultType="int">

select count(\*) from user\_main

</select>

<delete id="deleteUser" parameterType="java.lang.String">

delete from user\_main where user\_id=#{\_parameter}

</delete>

<update id="updateUserPower" parameterType="lr.entity.UserMain">

update user\_main set user\_power=#{user\_power} where user\_id=#{user\_id}

</update>

</mapper>

本系统使用Echarts来实现数学统计图的绘制，通过调用数据库特定属性的数据来生成相应的数学统计图。

下文以调用数据主题表中“项目类型”属性生成扇形统计图的代码为例。

var app = angular.module('myApp', []);

app.controller('projectShowCtrl', function($scope,$http) {

//用户列表，当前页，页总数

$scope.userDate={'one':{},'two':{},'third':{}};

// 基于准备好的dom，初始化echarts实例

var myChart = echarts.init(document.getElementById('main1'));

var myChart2 = echarts.init(document.getElementById('main2'));

var myChart3 = echarts.init(document.getElementById('main3'));

$http({

method: 'GET',

url: '/LearnRes/LR/UtilsControl/queryProject'

}).then(function success(response) {

// 请求成功执行代码

angular.copy(response.data,$scope.userDate);

$scope.showData();

}, function error(response) {});

$scope.showData=function() {

myChart.setOption({

series: [

{

name:'Test',

type: 'pie', // 设置图表类型为饼图

radius: '55%', // 饼图的半径，外半径为可视区尺寸（容器高宽中较小一项）的 55% 长度。

data: $scope.userDate.one.data

}

]

});

myChart2.setOption({

series: [

{

name: $scope.userDate.two.project\_name,

type: 'pie', // 设置图表类型为饼图

radius: '55%', // 饼图的半径，外半径为可视区尺寸（容器高宽中较小一项）的 55% 长度。

data:$scope.userDate.two.data

}

]

});

myChart3.setOption({

series: [

{

name: $scope.userDate.third.project\_name,

type: 'pie', // 设置图表类型为饼图

radius: '55%', // 饼图的半径，外半径为可视区尺寸（容器高宽中较小一项）的 55% 长度。

data: $scope.userDate.third.data} ] }); }});

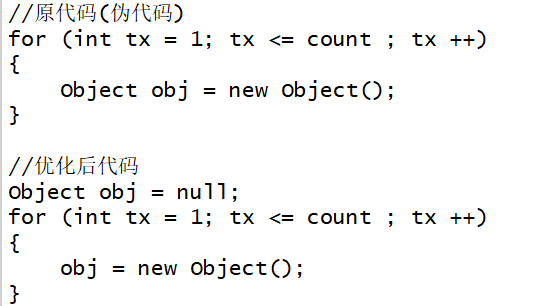
## 7.2 代码优化分析

### **7.2.1 代码优化目的**

所谓代码优化，即在不改变程序执行效果的同时更改实现该执行效果的代码，从而最终达到生成目标代码短（运行时间更短、占用空间更小），时空效率优化的结果。此外，由于部分程序员在编写代码时存在规范性不足的问题，故可通过代码优化来优化其代码可读性。

### **7.2.2 具体优化实例**

1. 考虑到实体类、控制层、数据操作层、服务层存在较多的共有代码，故分别对上述四种文件建立一个父类文件，可供剩余子类引用之。
2. 为优化可读性，故可制定一系列命名格式原则，具体表现在：使用“大驼峰”命名格式为类名命名（如WorkDao），使用“小驼峰”命名格式为方法、变量、参数等名称进行命名（如searchWork），令所有常量名称以大写字母进行表示（如TEST）。
3. 由于在用户数据库中的“隶属单位”、项目数据库中的“所属类型”等属性存在一定的重合性，故在数据库中将属性中的相同内容以数字形式统一表述，如在项目数据库的“所属类型”属性中，使用“3”指代“整改工程”、“2”指代“新建工程”等。
4. 如下列代码所示，由于原代码存在反复引用Object的情况，若此时count数值较大则极耗内存。经优化后，此时由于将创建对象语句置于循坏外，如此一来即可只引用该Object，每次new Object时可在仅耗一份内存的情况之下只需要指向不同的Object即可，节省空间。



8 测试

## 8.1 测试方案设计

### **8.1.1 测试策略**

(1) 测试目的

1. 软件测试是为了发现错误而执行程序的过程；
2. 发现错误并非软件测试的唯一目的；
3. 好的测试用例可发现当今尚未发现且存在的错误；
4. 成功的测试是发现了目前尚未发现且存在的错误测试；
5. 确保产品满足性能和效率的要求；
6. 确保产品完成了它所承诺或公布的功能。
7. 测试方法

当处于软件编码过程中，使用静态测试，通过编译软件给出的错误提示以及自身对代码的逻辑性分析进行代码修改工作。

当程序处于即将运行时，使用动态测试，首先检查软件是否可运行，其次在运行后针对特定功能输入相应的测试实例，待运行结束后检查其输出结果与预期结果之间的差异，从而检测本系统的编码正确性、程序有效性及可靠性，并可分析本系统的运行速率。

此外，当程序进行到模块阶段性开发结束时期，可使用黑盒测试，如此可忽视程序内部结构，只需要了解程序功能便可实行的测试。

### **8.1.2 测试进度安排**

编码阶段：使用静态测试，检查当前编写的代码是否存在问题，并可在试运行时使用动态测试，以检查程序是否出现错误。

模块阶段性开发结束时期：根据该软件的使用说明来制定测试用例并进行黑盒测试。

### **8.1.3 测试资源**

Apache JMeter 5.2.1

Google Chrome Ver. 57.0.2987.110

IntelliJ IDEA 2019.1.2 (Ultimate Edition)

由Apache组织开发的Apache JMeter是基于Java的压力测试工具，该工具主用于对软件做压力测试，可用于测试静态及动态资源，也可针对应用程序进行功能测试和回归测试，从而分析其具体性能。

### **8.1.4 关键测试点**

1. 数据库增删改查操作
2. 点击“添加”按键，检查是否跳出数据新增子界面；
3. 输入正确的数据，检查是否保存；
4. 输入非法数据，检查是否给出错误提示；
5. 当用户未输入某些必填数据时，检查是否出现错误提示；
6. 点击“删除”按键，检查对应数据是否已被删除；
7. 进行模糊查询时，检查是否出现对应结果。
8. 前后端联动
9. 于前端点击特定项目，检查是否跳出项目介绍页面；
10. 于前端点击特定项目并跳出项目介绍页面后，检查项目介绍页面的资料是否为数据库内数据；
11. 数据模型形成
12. 新增数据，检查是否生成数据模型；
13. 更改数据库某些数值性数据，检查生成的数据模型是否更新；
14. 修改生成数据模型的函数，检查生成的数据模型是否根据函数变化而往预期方向转变。
15. 各页面跳转
16. 点击特定跳转按钮，检查是否跳转至特定页面；
17. 点击特定跳转按钮，检查页面跳转之后是否正常显示。

## 8.2 测试用例构建

### **8.2.1 测试用例编写约定**

1. 测试用例编写原则
2. 系统性：针对系统业务流程，需要完整说明整个系统的业务需求、子系统数量以及各子系统之间的关系；对模块业务流程需要说明各子系统内部功能、重点功能以及各个子系统之间的关系；
3. 连贯性：针对系统业务流程，需要说明各个子系统之间如何进行拼接联动，若需要接口，需要检查各子系统之间是否存在正确的接口，若靠页面链接，则需要检查页面的链接是否正确；对模块业务流程要说明同级模块以及上下级模块构成子系统的形式，还需要检查其内部功能接口连贯性。
4. 全面性：应尽可能覆盖各种路径、尽可能覆盖各个业务点，并要考虑年份、月份交替时期的数据以及大数据量并发测试的准备。
5. 正确性：用于界面输入的数据应与测试文档所记录的数据相一致，此外预期结果也应该与测试数据发生的业务相一致。
6. 符合正常业务规则：测试数据需要符合用户实际工作中的业务流程，同时也要兼顾各种业务之间的变化，以及当前该业务行业的法律、法规等条例；
7. 人名、地名、电话号码等实际性强的属性应具有模拟功能，符合常规命名惯例；但不允许出现与实际存在的知名人士、文学作品中人物名等雷同情况；
8. 可操作性：测试用例中需要写清测试的操作步骤，以及不同的操作步骤相对应的测试结果。
9. 测试用例编写标准
10. 制订统一的测试用例模板进行测试，并约定测试用例模板的使用方法；
11. 根据项目实际情况编写测试案例编写手册，其中包括“案例编号规则”、“案例编写方法”、“案例编写内容”、“案例维护”等要素；此外，案例的编写应根据手册中约定的编写方法、内容等进行编写；
12. 步骤应当明确，输入、输出要素清晰，且与需求及存在的缺陷相对应；
13. 严格根据说明书及测试需求功能分析点进行，并要求覆盖全部需求功能点；
14. 注重测试用例的可复用性，即在以后定位类似系统的测试过程中可以重复使用该案例，从而减少测试用例设计的工作量。

### **8.2.2 测试用例设计**

1. 登录注册测试
2. 是否跳出登录注册页面；
3. 新用户是否可注册；
4. 拥有账号的用户是否可登录；
5. 当用户输入非法数据时是否出现错误提示。
6. 项目管理测试
7. 输入正确的数据之后是否保存；
8. 当用户输入非法数据时是否出现错误提示；
9. 当用户未输入某些必填数据时是否出现错误提示；
10. 点击“删除”后是否可删除该项目。
11. 扇形统计图生成测试
12. 输入正确的项目数据是否生成扇形统计图；
13. 输入正确的项目数据是否生成正确的扇形统计图；
14. 更改项目数据后是否一并更改扇形统计图。
15. 模糊查询测试
16. 输入所需要查询的项目全名后是否正确输出结果；
17. 输入所需要查询的项目部分名称后是否正确输出结果。
18. 用户管理测试
19. 输入正确的数据之后是否保存；
20. 当用户输入非法数据时是否出现错误提示；
21. 当用户未输入某些必填数据时是否出现错误提示；
22. 点击“删除”后是否可删除该项目；

### **8.2.3 关键测试用例**

表7-1 登录测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模块名称**：** | | 系统登录模块→用户登录 | |  |  | |
| 测试时间**：** | | 2020-3-3 | | 测试人**：** |  | |
| 操作序号 | | 01（用户登录） | | 错误等级 | 严重 □ | 一般 □ |
| 较严重 □ | 小错误 □ |
| 输入/动作 | | | 期望的输出/相应 | 实际情况 | 修改情况 | 复核 |
| 输入正确的系统地址 | http://localhost:8081/jxlwit/user/index | | 能正确的展示登录窗口 | 和预期输出相符! |  |  |
| 单击“登录”按钮 | 输入正确的用户名和密码 | | 正确登录系统 | 和预期输出相符! |  |  |
| 输入未设置权限的用户名 | | 系统给予正确提示 | 和预期输出相符! |  |  |
| 输入正确的用户名和密码 | | 无法登录系统 | 远程服务器崩溃 | 待远程服务器修复 | 正确登录系统 |
| 用户名不正确 | | 系统给予用户名不正确的提示 | 和预期输出相符! |  |  |
| 用户名为空 | | 系统给予用户名不能为空的提示 | 和预期输出相符! |  |  |
| 用户名为中文 | | 系统给予用户名不能为中文的提示 | 和预期输出相符! |  |  |
| 密码不正确 | | 系统给予密码不正确的提示 | 和预期输出相符! |  |  |
| 密码为空 | | 系统给予密码为空的提示 | 和预期输出相符! |  |  |
| 单击“取消”按钮 | | | 用户名和密码处全部清空 | 和预期输出相符! |  |  |

表7-2 项目管理测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模块名称**：** | | 用户管理模块→信息更改 | |  |  | |
| 测试时间**：** | | 2020-3-15 | | 测试人**：** |  | |
| 操作序号 | | 02（数据库操作） | | 错误等级 | 严重 □ | 一般 □ |
| 较严重 □ | 小错误 □ |
| 输入/动作 | | | 期望的输出/相应 | 实际情况 | 修改情况 | 复核 |
| 单击“提交”按钮 | 输入正确的项目数据 | | 保存该项目数据 | 和预期输出相符! |  |  |
| 未输入项目名 | | 系统给予“项目名不得为空”提示 | 和预期输出相符! |  |  |
| 未输入项目类型 | | 系统给予“项目类型不得为空”提示 | 和预期输出相符! |  |  |
| 未输入立项人 | | 系统给予“对口单位不得为空”提示 | 和预期输出相符! |  |  |
| 未填写对口单位 | | 系统给予“对口单位不得为空”提示 | 和预期输出相符! |  |  |
| 未填写项目简介 | | 保存数据 | 和预期输出相符! |  |  |
| 删除项目 | | 提示“已删除” | 和预期输出相符! |  |  |

表7-3 扇形统计图生成测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模块名称**：** | | 统计分析结果模块→图表生成 | |  |  | |
| 测试时间**：** | | 2020-3-22 | | 测试人**：** |  | |
| 操作序号 | | 03（数据库操作） | | 错误等级 | 严重 □ | 一般 □ |
| 较严重 □ | 小错误 □ |
| 输入/动作 | | | 期望的输出/相应 | 实际情况 | 修改情况 | 复核 |
| 单击“提交”按钮 | 输入正确的项目数据 | | 生成正确的扇形统计图 | 无图表生成 | 未调用生成扇形统计图的函数 | 下一步进行 |
| 输入正确的项目数据 | | 生成正确的扇形统计图 | 当前的图表与提交前图表一致，未更新 | 数据库中调用了错误的属性 | 下一步进行 |
| 输入正确的项目数据 | | 生成正确的扇形统计图 | 和预期输出相符! |  |  |
| 单击“删除”按钮 | | | 删除该图表并提示“删除成功” | 未删除图表 |  | 下一步进行 |
| 删除该图表并提示“删除成功” | 和预期输出相符! |  |  |

表7-4 模糊查询测试用例

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 模块名称**：** | | 用户管理模块→信息更改 | |  |  | | |
| 测试时间**：** | | 2020-3-22 | | 测试人**：** |  | | |
| 操作序号 | | 04（数据库操作） | | 错误等级 | 严重 □ | 一般 □ | |
| 较严重 □ | 小错误 □ |
| 输入/动作 | | | 期望的输出/相应 | 实际情况 | 修改情况 | 复核 |
| 输入需要查询的项目全名 | 单击“搜索”按钮 | | 显示出所需要查询的项目 | 无结果 | 查询语句编写出错 | 下一步进行 |
| 输入需要查询的项目全名 | 显示出所需要查询的项目 | 和预期输出相符! |  |  |
| 输入需要查询的项目关键字 | 显示出所需要查询的项目 | 无结果 | 查询语句编写出错 | 下一步进行 |
| 输入需要查询的项目关键字 | 显示出所需要查询的项目 | 和预期输出相符! |  |  |

表7-5 用户管理测试用例

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模块名称**：** | | 用户管理模块→信息更改 | |  |  | |
| 测试时间**：** | | 2020-3-26 | | 测试人**：** |  | |
| 操作序号 | | 05（数据库操作） | | 错误等级 | 严重 □ | 一般 □ |
| 较严重 □ | 小错误 □ |
| 输入/动作 | | | 期望的输出/相应 | 实际情况 | 修改情况 | 复核 |
| 单击“提交”按钮 | 输入正确的用户数据 | | 保存该项目数据 | 和预期输出相符! |  |  |
| 未输入用户名 | | 系统给予“用户名不得为空”提示 | 和预期输出相符! |  |  |
| 联系电话使用汉字表示 | | 系统给予“联系电话格式错误” | 和预期输出相符! |  |  |
| 未填写所属单位 | | 系统给予“所属单位不得为空”提示 | 和预期输出相符! |  |  |
| 未填写项目简介 | | 保存数据 | 和预期输出相符! |  |  |
| 未填写联系电话 | | 保存数据 | 和预期输出相符! |  |  |
| 删除项目 | | 提示“已删除”并给出提示 | 和预期输出相符! |  |  |

### **8.2.4 测试用例维护**

(1) 测试用例维护原因

维护测试用例，不仅可改善测试用例的可用性，一并提高了测试库的可信性，同时还能将一个基线测试用例库的效率和时效性保持在一个较高的级别上。测试用例维护方法大致分为以下几点：

1. 删除失去时效性的测试用例：由于需求的变更，可能导致先前所使用的测试用例不再具有时效性，应当删除。以变量界限为例，若在后期开发中改变了该变量界限，由此则会改变先前所定的边界值，如此一来原有针对边界值的测试即无法针对现有边界值进行测试，已失去时效性，故需删除；
2. 改进可控性差的测试用例：随着项目开发的进展，测试用例数量也会随之而涨，故存在一部分不易重复且难以控制其结果的测试用例，如此一来便会影响测试的效率，故需要针对此类测试用例进行改进，从而达到重复性及可控性均良好的要求。
3. 删除冗余测试用例：若存在两个及以上针对一组相同的输入和输出的测试用例，即说明以上测试用例为冗余的，这些冗余测试用例的存在会降低了回归测试的效率，因此需要定期进行整理测试用例库工作，并删除已存在的冗余测试用例。
4. 添加新的测试用例：若某个程序代码段、构件或关键的接口在现有的测试用例中并未参与过测试，则需要针对此类情况设计新的测试用例并使用该用例对其进行测试，再新开发的测试用例合并至基线测试包之内。

(2) 测试用例维护实例

以项目管理测试为例：

由于项目需要与立项者及立项者的各个人信息（隶属单位、联系方式）相绑定，故需要对项目管理测试用例作出如下调整：

1. 该项目是否有立项者；
2. 立项者姓名、所属单位、联系方式是否有误。

9 总结与展望

## 设计工作总结

本次制作的是发改委展示系统，涉及到一些投标的问题，目前对于本人而言其概念还比较陌生，不过万变不离其宗。主要的问题在于未接触过形成数学图表功能，在该功能处研究了一定的时间。此外，目前的数据库还存在一定的不稳定性，依然需要继续维护。

由于本人常制作Word文档供己使用，故在编写文档方面并未碰到诸如文字软件使用、排版等问题。实际上，真正的问题出现在项目本身。这次的项目主体有些许绕，有些东西换了一些说法，导致在早期编写诸如需求分析、系统设计之类的文档时出了比较多问题。好在参与了学校组织的校内培训，使得自己对软件开发所需要编写的相关文档有了一定的了解，这也使得自己可以以之前的文档作为本次毕业设计的参考。可以说，这次论文编写之所以可以在家进行，这与去年为含本人在内的一众学生进行培训的老师相关，正是因为这次培训，我们才有了一定的知识进行项目操作。

至于编码，考虑到本次项目与培训时编写的相似性，故可以用之前的项目进行参考。当然，正如上文所言，本人并未接触过形成数学图表的相关功能编写，所以在这个地方折腾了比较多的时间，具体应该是从，自己也不知道为什么会弄那么久。好在现在的项目开发工作与论文编写工作已接近尾声，这一切应该都是值得的。

## 未来工作展望

考虑到本次制作的是一个与政府相关的展示平台，本认为政府网站依然与数年之前类似，为缺少样式的简单界面外加悬浮广告的页面，忽略了近年以来由于国家的扶持以及资源提供问题，现在的政府网站确实做得比较好了，功能应有尽有，很难出现前几年因不明原因而无法连接的情况。

如此一来，编写政府相关平台并不像以往那样敷衍。结合最近政府网站得到的资源来看，当前无论是质方面还是量方面都比较优秀。软件产品质量的提升也代表着制造该软件产品的相关人员必须在各种方面提高自身素质，如熟悉政府机关工作流程、熟悉政府机关具体职能、了解现有政府网站的特点及不足，当然，更重要的还是需要让自己有足够的软件开发能力。

至于日后，由于本人近年以来主要研究的是网络拓扑图的绘制及测试，故可能不会在网站开发上下过多的功夫，不过整个计算机行业都是相通的，通过这次掌握的知识及开发经验必然会为日后的工作起到一定的作用。此外，由于自己先前也搭建过Discuz论坛平台，日后可以用当前学到的技术来优化之。

谢 辞

参考文献

1. 刘国钧，陈绍业，王凤翥. 图书馆目录[M]. 北京：高等教育出版社, 2018-4-1
2. 张海藩 等. 软件工程导论（第6版）[M]. 清华大学出版社, 2018-8-1
3. 思科产品与解决方案中心（中文版）[Z/OL]. https://www.cisco.com/cn
4. 萨师煊 等. 数据库系统概论（第5版）[M]. 高等教育出版社, 2017-9-1
5. 严蔚敏，吴伟民. 数据结构 [M]. 清华大学出版社, 2017-3-1
6. 杨开振 等. Java EE互联网轻量级框架整合开发[M]. 电子工业出版社, 2017-9-1
7. Nicholas C. Zakas 著， 李松峰 曹力 译. JavaScript高级程序设计（第3版）[M]. 人民邮电出版社, 2018-3-1
8. 福尔著. TCP/IP详解卷1:协议（英文版.第2版） [M]. 机械工业出版社, 2018-5-1
9. 阿霍. 编译原理（第2版）[M]. 机械工业出版社, 2019-1-1
10. [美]Nicholas S. Williams 著，王肖峰 译，java for Web Applications,Wiley Publishing,Inc [M]. 授权清华大学出版社出版，2017-6-3
11. （美）弗兰纳根　著，JavaScript权威指南（第八版） [M]，机械工业出版社，2016-04-01
12. （美）罗恩佩腾 著，张小松 曹跃等译，软件测试 [M]，机械工业出版社，2015-05-01
13. （美）凯S霍斯特曼 著，林琪等 译，Java核心技术 卷一 基础知识（原书第11版）[M]，机械工业出版社，2020-01-01
14. 王晓东 著，计算机算法设计与分析（第5版）[M]，电子工业出版社，2015-03-01
15. （英）Ben Forta 著 ，刘晓霞、钟鸣等 译，MySQL必知必会 [M]，人民邮电出版社，2018-01-01
16. MARK FEWSTER DOROTHY GRAHAM著 ，Software Test Automation Effective use of test execution tools [M]. ACM Press New York，2016-2-2
17. 晏小冬. 新余市家政服务业转型升级策略研究[D].广东:广东工业大学，2016.
18. 石正喜.MySQL数据库实用教程 [M]. 北京师范大学出版社，2014
19. Caliskan . Sevindik.Beginning Spring[M].北京:清华大学出版社，2015.
20. Dyck.Spring. MVC (A Tutorial series)[M].北京:人民邮电出版社，2015.
21. 省发展和改革委员会.省发展和改革委员会政策发布-本委规范性文件介绍 [Z/OL]. http://drc.jiangxi.gov.cn/col/col14652/index.html
22. 张素琴，吕映芝. 编译原理[M]. 北京, 清华大学出版社, 2016
23. 斯桃枝. 路由协议与交换技术 [M]. 北京：清华大学出版社，2018.1
24. 王长征，王兴宇. 计算机网络 [M]. 南昌：高校出版社，2018.6
25. 谢希仁. 计算机网络（第7版）[M]. 北京：电子工业出版社，2017.1
26. Tony Bautts, Terry Dawson. Linux Network Administrator's Guide [M]. Sebastopol, CA: O'Reilly Media,Inc., 2016.7
27. 蒋立源、康慕宁等，编译原理（第2版）[M]，西安：西北工业大学出版社, 2015
28. WordPress. 安装WordPress（中文版） [Z/OL].

https://codex.wordpress.org/zh-cn:%E5%AE%89%E8%A3%85\_WordPress

1. 蒋立源、康慕宁等，编译原理（第3版）[M]，西安：西北工业大学出版社, 2016
2. 何绍华，臧伟，孟学奇. Linux操作系统（第3版） [M]. 北京：人民邮电出版社，2017.8（2018.3重印）

附录A 外文翻译—原文部分

原文：Wikipedia - Unified Modeling Language

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1994–1995, with further development led by them through 1996.

In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005, UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard. Since then the standard has been periodically revised to cover the latest revision of UML.

UML offers a way to visualize a system's architectural blueprints in a diagram, including elements such as:

any activities (jobs);

individual components of the system;

and how they can interact with other software components;

how the system will run;

how entities interact with others (components and interfaces);

external user interface.

Although originally intended for object-oriented design documentation, UML has been extended to a larger set of design documentation (as listed above), and been found useful in many contexts.

Software development methods[edit]

UML is not a development method by itself; however, it was designed to be compatible with the leading object-oriented software development methods of its time, for example OMT, Booch method, Objectory and especially RUP that it was originally intended to be used with when work began at Rational Software.

Modeling

It is important to distinguish between the UML model and the set of diagrams of a system. A diagram is a partial graphic representation of a system's model. The set of diagrams need not completely cover the model and deleting a diagram does not change the model. The model may also contain documentation that drives the model elements and diagrams (such as written use cases).

UML diagrams represent two different views of a system model:

Static (or structural) view: emphasizes the static structure of the system using objects, attributes, operations and relationships. It includes class diagrams and composite structure diagrams.

Dynamic (or behavioral) view: emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects. This view includes sequence diagrams, activity diagrams and state machine diagrams.

UML models can be exchanged among UML tools by using the XML Metadata Interchange (XMI) format.

In UML, one of the key tools for behavior modeling is the use-case model, caused by OOSE. Use cases are a way of specifying required usages of a system. Typically, they are used to capture the requirements of a system, that is, what a system is supposed to do.

The Meta-Object Facility (MOF) is an Object Management Group (OMG) standard for model-driven engineering. Its purpose is to provide a type system for entities in the CORBA architecture and a set of interfaces through which those types can be created and manipulated. The official reference page may be found at OMG's website.

MOF was developed to provide a type system for use in the CORBA architecture, a set of schemas by which the structure, meaning and behaviour of objects could be defined, and a set of CORBA interfaces through which these schemas could be created, stored and manipulated.

MOF is designed as a four-layered architecture. It provides a meta-meta model at the top layer, called the M3 layer. This M3-model is the language used by MOF to build metamodels, called M2-models. The most prominent example of a Layer 2 MOF model is the UML metamodel, the model that describes the UML itself. These M2-models describe elements of the M1-layer, and thus M1-models. These would be, for example, models written in UML. The last layer is the M0-layer or data layer. It is used to describe real-world objects.

Beyond the M3-model, MOF describes the means to create and manipulate models and metamodels by defining CORBA interfaces that describe those operations. Because of the similarities between the MOF M3-model and UML structure models, MOF metamodels are usually modeled as UML class diagrams. A supporting standard of MOF is XMI, which defines an XML-based exchange format for models on the M3-, M2-, or M1-Layer.

MyBatis is a Java persistence framework that couples objects with stored procedures or SQL statements using an XML descriptor or annotations.

MyBatis is free software that is distributed under the Apache License 2.0.

MyBatis is a fork of iBATIS 3.0 and is maintained by a team that includes the original creators of iBATIS.

Unlike ORM frameworks, MyBatis does not map Java objects to database tables but Java methods to SQL statements.

MyBatis lets you use all your database functionality like stored procedures, views, queries of any complexity and vendor proprietary features. It is often a good choice for legacy or de-normalized databases or to obtain full control of SQL execution.

It simplifies coding compared to JDBC. SQL statements are executed with a single line.

MyBatis provides a mapping engine that maps SQL results to object trees in a declarative way.

SQL statements can be built dynamically by using a built-in language with XML-like syntax or with Apache Velocity using the Velocity integration plugin.

MyBatis integrates with Spring Framework and Google Guice. This feature allows one to build business code free of dependencies.

MyBatis supports declarative data caching. A statement can be marked as cacheable so any data retrieved from the database will be stored in a cache and future executions of that statement will retrieve the cached data instead hitting the database. MyBatis provides a default cache implementation based on a Java HashMap and default connectors for integrating with: OSCache, Ehcache, Hazelcast and Memcached. It provides an API to plug other cache implementations.

SQL statements are stored in XML files or annotations. Below depicts a MyBatis mapper, that consists of a Java interface with some MyBatis annotations:

MyBatis integrates with Spring Framework. This module allows MyBatis to participate in Spring transactions. It will also build MyBatis mappers and sessions and inject them into other beans.

The following sample shows a basic XML configuration that sets up a mapper and injects it into a "BlogService" bean.

MyBatis provides a code generator. MyBatis Generator will introspect a database table (or many tables) and generate MyBatis artifacts needed to perform CRUD operations (Create, Retrieve, Update, Delete).

An Eclipse plugin is available.

It will preserve any custom code in case of regeneration but only if you use the Eclipse plugin.

MyBatis Migrations is a Java command line tool that keeps track of database schema changes managing DDL files (known as migrations).

Migrations allows to query the current status of the database, apply schema changes and also undo them. It also helps to detect and solve concurrent database schema changes made by different developers.

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented toward relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

Sun Microsystems released JDBC as part of Java Development Kit (JDK) 1.1 on February 19, 1997.[1] Since then it has been part of the Java Platform, Standard Edition (Java SE).

The JDBC classes are contained in the Java package java.sql and javax.sql.

Starting with version 3.1, JDBC has been developed under the Java Community Process. JSR 54 specifies JDBC 3.0 (included in J2SE 1.4), JSR 114 specifies the JDBC Rowset additions, and JSR 221 is the specification of JDBC 4.0 (included in Java SE 6).[2]

JDBC 4.1, is specified by a maintenance release 1 of JSR 221[3] and is included in Java SE 7.[4]

JDBC 4.2, is specified by a maintenance release 2 of JSR 221[5] and is included in Java SE 8.[6]

The latest version, JDBC 4.3, is specified by a maintenance release 3 of JSR 221[7] and is included in Java SE 9.

JDBC ('Java Database Connectivity') allows multiple implementations to exist and be used by the same application. The API provides a mechanism for dynamically loading the correct Java packages and registering them with the JDBC Driver Manager. The Driver Manager is used as a connection factory for creating JDBC connections.

JDBC connections support creating and executing statements. These may be update statements such as SQL's CREATE, INSERT, UPDATE and DELETE, or they may be query statements such as SELECT. Additionally, stored procedures may be invoked through a JDBC connection. JDBC represents statements using one of the following classes:

Statement – the statement is sent to the database server each and every time.

PreparedStatement – the statement is cached and then the execution path is pre-determined on the database server allowing it to be executed multiple times in an efficient manner.

CallableStatement – used for executing stored procedures on the database.

Update statements such as INSERT, UPDATE and DELETE return an update count that indicates how many rows were affected in the database. These statements do not return any other information.

Query statements return a JDBC row result set. The row result set is used to walk over the result set. Individual columns in a row are retrieved either by name or by column number. There may be any number of rows in the result set. The row result set has metadata that describes the names of the columns and their types.

There is an extension to the basic JDBC API in the javax.sql.

JDBC connections are often managed via a connection pool rather than obtained directly from the driver.

A JDBC driver is a software component enabling a Java application to interact with a database. JDBC drivers are analogous to ODBC drivers, ADO.NET data providers, and OLE DB providers.

To connect with individual databases, JDBC (the Java Database Connectivity API) requires drivers for each database. The JDBC driver gives out the connection to the database and implements the protocol for transferring the query and result between client and database.

JDBC technology drivers fit into one of four categories.

JDBC-ODBC bridge

Native-API driver

Network-Protocol driver (Middleware driver)

Database-Protocol driver (Pure Java driver) or thin driver.

附录B 外文翻译—译文部分

原文：维基百科——统一建模语言（UML）

统一建模语言（UML）是软件工程领域中的一种通用的，开发性的建模语言，旨在提供一种可视化系统设计的标准方法。

UML的创建最初是出于对不同的符号系统和软件设计方法进行标准化的愿望。它是由Rational Software的Grady Booch，Ivar Jacobson和James Rumbaugh于1994–1995年开发的，并在1996年之前进一步进行了开发。

1997年，对象管理组（OMG）将UML用作标准，此后一直由该组织进行管理。 2005年，UML也被国际标准化组织（ISO）发布为批准的ISO标准。从那时起，该标准就进行了定期修订，以涵盖UML的最新修订。

UML提供了一种在图表中可视化系统架构蓝图的方法，其中包括以下元素：

任何活动（工作）；

系统的各个组件；

以及它们如何与其他软件组件进行交互；

系统将如何运行；

实体如何与他人互动（组件和接口）；

外部用户界面。

尽管UML最初旨在用于面向对象的设计文档，但已扩展到更大的一组设计文档（如上所述），并且在许多情况下都非常有用。

软件开发方法[编辑]

UML本身并不是一种开发方法。但是，它被设计为与当时领先的面向对象的软件开发方法兼容，例如OMT，Booch方法，Objectory以及特别是RUP，这些方法最初是在Rational Software开始工作时就使用的。

造型

区分UML模型和系统图集很重要。图表是系统模型的部分图形表示。这组图不必完全覆盖模型，删除图也不会更改模型。该模型还可能包含驱动模型元素和图的文档（例如书面用例）。

UML图表示系统模型的两个不同视图：

静态（或结构）视图：使用对象，属性，操作和关系强调系统的静态结构。它包括类图和复合结构图。

动态（或行为）视图：通过显示对象之间的协作以及对象内部状态的更改来强调系统的动态行为。该视图包括顺序图，活动图和状态机图。

通过使用XML元数据交换（XMI）格式，可以在UML工具之间交换UML模型。

在UML中，行为建模的关键工具之一是由OOSE引起的用例模型。用例是指定系统所需用法的一种方式。通常，它们用于捕获系统需求，即系统应该执行的操作。

元对象工具（MOF）是用于模型驱动工程的对象管理组（OMG）标准。其目的是为CORBA体系结构中的实体提供类型系统，以及为一组接口创建和操纵这些类型的接口。官方参考页可以在OMG的网站上找到。

开发MOF的目的是提供一种用于CORBA体系结构的类型系统，一组可以定义对象的结构，含义和行为的方案，以及一组可以创建，存储和操纵这些方案的CORBA接口。 。

MOF被设计为四层体系结构。它在顶层（称为M3层）提供了元元模型。此M3模型是MOF用于构建元模型的语言，称为M2模型。第2层MOF模型的最突出示例是UML元模型，该模型描述了UML本身。这些M2模型描述了M1层的元素，因此也描述了M1模型。例如，这些将是用UML编写的模型。最后一层是M0层或数据层。它用于描述现实世界的对象。

除了M3模型之外，MOF还通过定义描述这些操作的CORBA接口来描述创建和操纵模型和元模型的方法。由于MOF M3模型和UML结构模型之间的相似性，因此MOF元模型通常被建模为UML类图。 MOF的支持标准是XMI，它为M3-，M2-或M1-Layer上的模型定义了基于XML的交换格式。

MyBatis是一个Java持久化框架，它通过XML描述符或注解把对象与存储过程或SQL语句关联起来。

MyBatis是在Apache许可证 2.0下分发的自由软件，是iBATIS 3.0的分支版本。其维护团队也包含iBATIS的初创成员。

与其他的对象关系映射框架不同，MyBatis并没有将Java对象与数据库表关联起来，而是将Java方法与SQL语句关联。MyBatis允许用户充分利用数据库的各种功能，例如存储过程、视图、各种复杂的查询以及某数据库的专有特性。如果要对遗留数据库、不规范的数据库进行操作，或者要完全控制SQL的执行，MyBatis是一个不错的选择。

与JDBC相比，MyBatis简化了相关代码：SQL语句在一行代码中就能执行。MyBatis提供了一个映射引擎，声明式的把SQL语句执行结果与对象树映射起来。通过使用一种内建的类XML表达式语言，或者使用Apache Velocity集成的插件，SQL语句可以被动态的生成。

MyBatis与Spring Framework和Google Guice集成，这使开发者免于依赖性问题。

MyBatis支持声明式数据缓存（declarative data caching）。当一条SQL语句被标记为“可缓存”后，首次执行它时从数据库获取的所有数据会被存储在一段高速缓存中，今后执行这条语句时就会从高速缓存中读取结果，而不是再次命中数据库。MyBatis提供了基于 Java HashMap 的默认缓存实现，以及用于与OSCache、Ehcache、Hazelcast和Memcached连接的默认连接器。MyBatis还提供API供其他缓存实现使用。

Java数据库连接，（Java Database Connectivity，简称JDBC）是Java语言中用来规范客户端程序如何来访问数据库的应用程序接口，提供了诸如查询和更新数据库中数据的方法。JDBC也是Sun Microsystems的商标。JDBC是面向关系型数据库的。

在J2SE中，提供了一个称之为JDBC-ODBC桥（JDBC-ODBC Bridge）的API。通过ODBC，JDBC-ODBC桥驱动程序可以访问所有支持ODBC的关系型数据库。与JDBC API不同的是，这个驱动程序并不是由Java代码而是由机器代码（native code）编写，并且不是开放源代码的。

附录C 软件使用说明书

本系统分为用户管理子系统、数据主题管理子系统、个性化定制展示子系统、统计分析结果子系统、可视化展现子系统。

用户管理子系统包括查看个人信息功能、修改个人信息功能，以及审核个人信息功能。

数据主题管理子系统包括添加项目功能、修改项目、删除项目、查询项目，以及审核项目信息功能。

个性化定制展示子系统包括添加个性化内容、审核、修改个性化内容、删除个性化内容以及查询个性化内容功能。

统计分析结果子系统函数修改、函数删除、同比计算、环比计算、立方体计算、趋势计算、数学模型查询及数学模型删除功能。

展示综合联动子系统包括项目查看、项目查询以及投标申请功能。



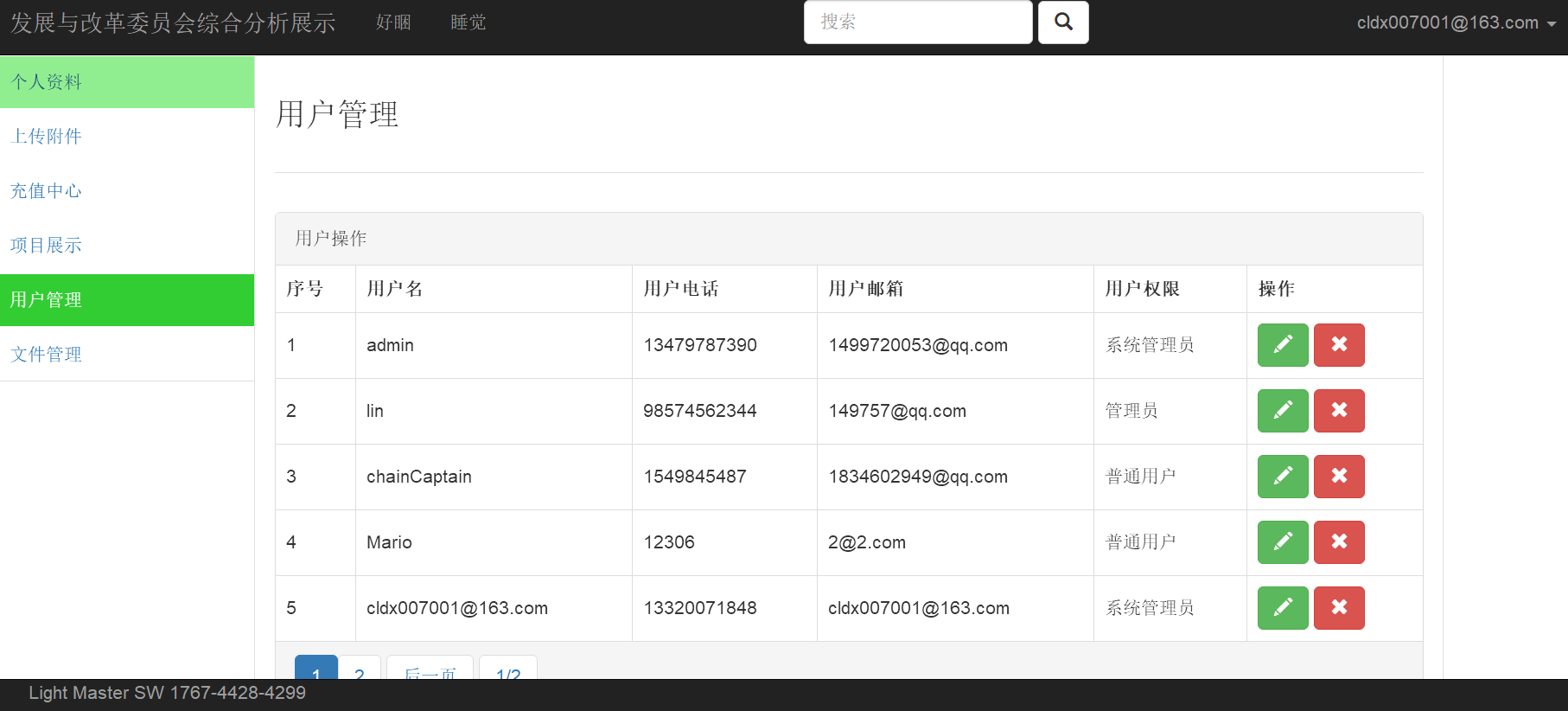
用户进入本系统后，首先将进入主页。若当前用户拥有账号，则可以在右侧位置进行登录；若当前用户需要账号，则可以点击右上角处进行注册。用户登录之后，则会自动进入后台界面。

主页上方有快捷栏，分别可进入新闻动态界面、项目展示界面、图表展示界面、投标界面，以及搜索界面。

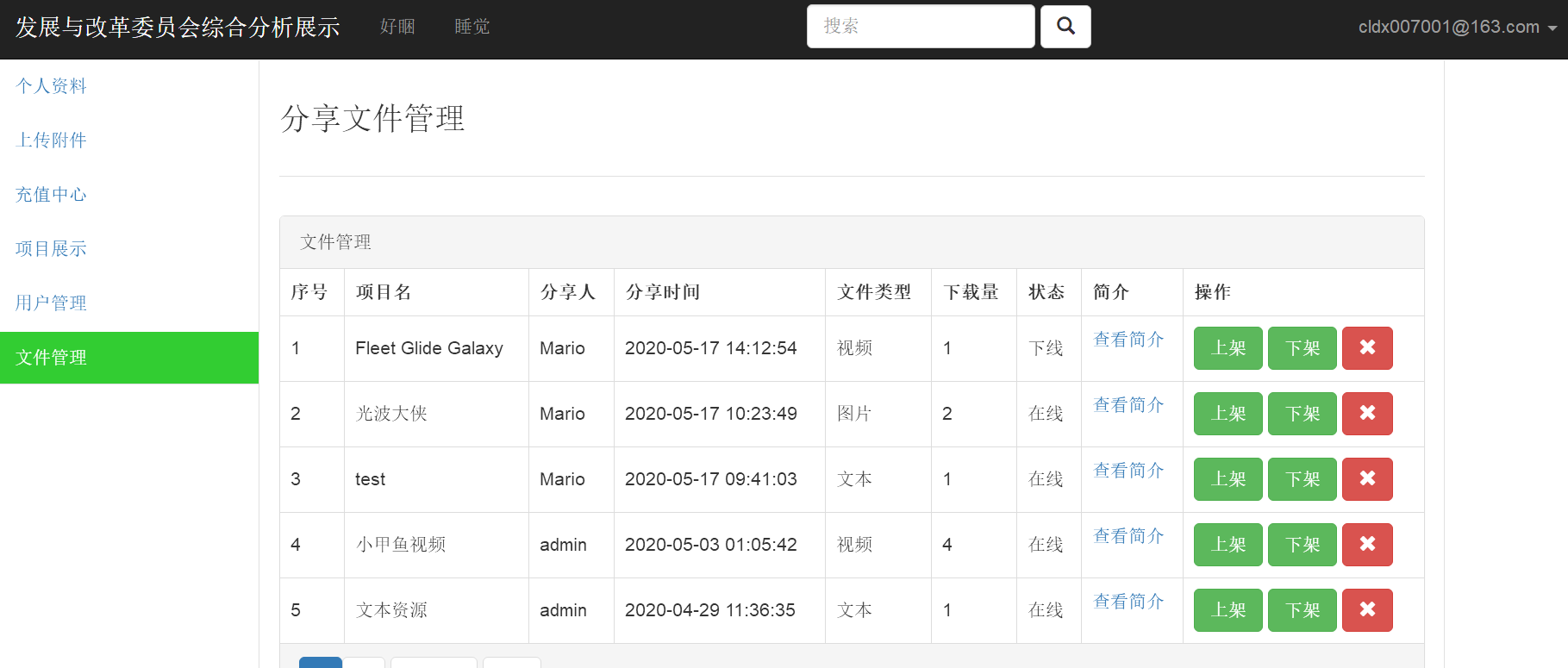
用户登录之后，则会进入后台，可更改其个人信息，亦可提交及修改其项目的相关信息，如下图所示：

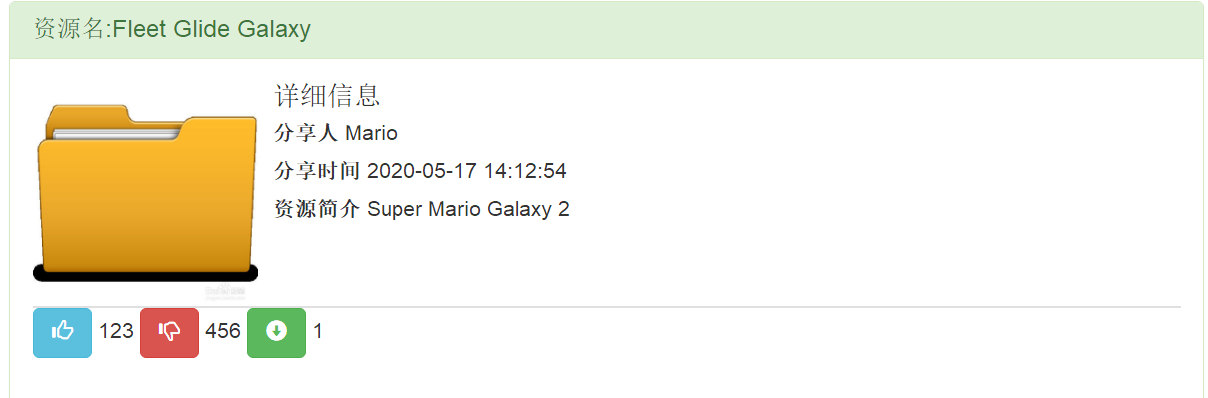






管理员可以审核普通用户提交项目时上传的附件，若项目无问题即可上架，亦可下架。





附录D 主要源代码

实体类Project：

package entity;

import kesun.entity.AbsBusinessObject;

import kesun.entity.AbsSuperObject;

public class Project extends AbsSuperObject {

private int pId;

private String pName;

private String pType;

private String rName;

private String rUnit;

private int rPhone;

private String introduction;

private int construction;

private float funds;

public int getpId() {

return pId;

}

public void setpId(int pId) {

this.pId = pId;

}

public String getpName() {

return pName;

}

public void setpName(String pName) {

this.pName = pName;

}

public String getpType() {

return pType;

}

public void setpType(String pType) {

this.pType = pType;

}

public String getrName() {

return rName;

}

public void setrName(String rName) {

this.rName = rName;

}

public String getrUnit() {

return rUnit;

}

public void setrUnit(String rUnit) {

this.rUnit = rUnit;

}

public int getrPhone() {

return rPhone;

}

public void setrPhone(int rPhone) {

this.rPhone = rPhone;

}

public String getIntroduction() {

return introduction;

}

public void setIntroduction(String introduction) {

this.introduction = introduction;

}

public int getConstruction() {

return construction;

}

public void setConstruction(int construction) {

this.construction = construction;

}

public float getFunds() {

return funds;

}

public void setFunds(float funds) {

this.funds = funds;

}

}

服务层：

package bll.attchment.impl;

import bll.KesunSuperService;

import bll.attchment.IProjectService;

import dao.IDoData;

import entity.Project;

import kesun.entity.system.User;

import org.apache.poi.ss.usermodel.Row;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

@Service("projectservice")

public class ProjectServiceImpl extends KesunSuperService implements IProjectService {

@Autowired

private dao.inter.attachment.IProjectDao projectDao;

public ProjectServiceImpl() {

setModel(new Project());

}

@Override

public String getLoadoutExcelFileName() {

return null;

}

@Override

public IDoData getDAO() {

return projectDao;

}

@Override

public List getSaveAll(List<Row> source, User worker) {

return null;

}

@Override

public String getLoadoutExcelTemplateName() {

return null;

}

}

控制层：

package controller.attachment.impl;

import bll.KesunSuperService;

import controller.KesunBusinessController;

import controller.attachment.IProjectController;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import javax.annotation.Resource;

@Controller

@RequestMapping("Project")

public class ProjectController extends KesunBusinessController implements IProjectController {

@Resource(name="projectservice")

private bll.attchment.impl.ProjectServiceImpl bll;

@Override

public KesunSuperService getService() {

return bll;

}

@RequestMapping("index")

public String index(){

return "project/project";

}

@RequestMapping("project")

public String project(){

return "project/projectmanage";

}

}

数据操作层：

package dao;

import org.apache.ibatis.annotations.Param;

import org.springframework.stereotype.Repository;

import java.util.List;

import java.util.Map;

/\*\*

\* 常规数据操作接口

\* Created by wph-pc on 2017/10/9.

\*/

public interface IDoData {

/\*新增操作\*/

int save(Object obj);

/\*批量新增操作\*/

int batchSave(List objs);

/\*批量修改操作\*/

int batchUpdate(List objs);

/\*更新操作\*/

int update(Object obj);

int update1(Object obj);

/\*对象状态更新操作\*/

int changeStatus(Object obj);

/\*删除操作\*/

int delete(Object obj);

/\*批量删除操作\*/

int batchDelete(List objs);

/\*查找当前对象\*/

Object getMe(Object obj);

/\*查找对象ID的数据是否在使用中\*/

Boolean objectInUse(String id);

/\*查找符合条件values的对象,以List的Map形式返回\*/

List<Map<String,Object>> findForMap(Map values);

/\*快速查找符合条件values的对象,以List的Map对象形式返回\*/

List<Map<String,Object>> fastFind(Map values);

/\*查找符合条件values的对象,以List的对象形式返回\*/

List<?> find(Map values);

List<?> find1(Map values);

List<?> find2(Map values);

/\*查找符合条件values的数据\*/

List<Map<String,Object>> findByPage(Map conValues);

/\*查找符合条件values的行数\*/

int getRowsCount(Map values);

}

XML：

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="dao.inter.attachment.IProjectDao">

<resultMap id="BaseResultMap" type="entity.Project">

<result column="id" property="id"/>

<result column="name" property="name"/>

<result column="status" property="status"/>

<association property="theme\_id" javaType="entity.Theme">

<id column="theme\_id" property="id"></id>

<result column="themeName" property="name"/>

</association>

<association property="user" javaType="kesun.entity.system.User">

<id column="user" property="id"></id>

<result column="userName" property="name"/>

</association>

<association property="actor" javaType="kesun.entity.system.Actor">

<id column="actor" property="id"></id>

<result column="actorName" property="name"/>

</association>

<association property="org" javaType="kesun.entity.system.Organization">

<id column="org" property="id"></id>

<result column="orgName" property="name"/>

</association>

</resultMap>

<sql id="Base\_Column\_List">

id,name,status,theme\_id

</sql>

<sql id="colums">

<trim suffix="" suffixOverrides=",">

<if test="id != null">

id,

</if>

<if test="name != null">

name,

</if>

<if test="status != null">

status,

</if>

<if test="theme\_id != null">

theme\_id,

</if>

<if test="user != null">

user,

</if>

<if test="actor !=null">

actor,

</if>

<if test="org != null">

org,

</if>

</trim>

</sql>

<sql id="columsValue">

<trim suffix="" suffixOverrides=",">

<if test="id != null">

#{id},

</if>

<if test="name != null">

#{name},

</if>

<if test="status != null">

#{status},

</if>

<if test="theme\_id != null">

#{theme\_id.id},

</if>

<if test="user != null">

#{user.id},

</if>

<if test="actor != null">

#{actor.id},

</if>

<if test="org != null">

#{org.id},

</if>

</trim>

</sql>

<sql id="conditionByPower">

<where>

<trim prefix="" suffixOverrides="and">

org=#{org}

</trim>

</where>

</sql>

<!--单个对象新增-->

<insert id="save" parameterType="entity.Project">

insert into project (

<include refid="colums"/>

) values (

<include refid="columsValue"/>)

</insert>

<!--批量新增-->

<insert id="batchSave" parameterType="java.util.List">

<foreach collection="list" item="item" index="index">

insert into project (

id,name,status，user,actor,org

) values

(

#{item.id},#{item.name},#{item.status},#{item.user},#{item.actor},#{item.org}

);

</foreach>

</insert>

<!--单个对象更新-->

<update id="update" parameterType="entity.Project">

update project set

<trim suffix="" suffixOverrides=",">

<if test="id != null">

id=#{id},

</if>

<if test="name != null">

name=#{name},

</if>

<if test="status != null">

status=#{status},

</if>

<if test="theme\_id != null">

theme\_id=#{theme\_id.id},

</if>

<if test="user != null">

user=#{user.id},

</if>

<if test="actor != null">

actor=#{actor.id},

</if>

<if test="org != null">

org=#{org.id},

</if>

</trim>

where id=#{oldId,jdbcType=VARCHAR}

</update>

<!--对象批量更新-->

<update id="batchUpdate" parameterType="java.util.List">

<foreach collection="list" item="item" index="index" open="" close="" separator=";">

update project set

<trim suffix="" suffixOverrides=",">

<if test="id != null">

id=#{id},

</if>

<if test="name != null">

name=#{name},

</if>

<if test="status != null">

status=#{status},

</if>

<if test="theme\_id != null">

theme\_id=#{theme\_id.id},

</if>

<if test="user != null">

user=#{item.user},

</if>

<if test="actor != null">

actor=#{item.actor},

</if>

<if test="org != null">

org=#{item.org},

</if>

</trim>

where id=#{item.oldId,jdbcType=VARCHAR}

</foreach>

</update>

<!--单个对象删除-->

<delete id="delete" parameterType="java.lang.String">

delete from project where id = #{id,jdbcType=VARCHAR}

</delete>

<!--批量删除-->

<delete id="batchDelete" parameterType="java.util.List">

delete from project

<where>

id in

<foreach collection="list" item="item" separator="," open="(" close=")">#{item,jdbcType=VARCHAR}</foreach>

</where>

</delete>

<!--单个对象查找-->

<select id="getMe" parameterType="java.lang.String" resultMap="BaseResultMap">

SELECT

a.\*,b.name as themeName

FROM project a LEFT JOIN theme b ON a.theme\_id = b.id WHERE a.id = #{id,jdbcType=VARCHAR};

</select>

<!--查找，以Map返回-->

<select id="findForMap" parameterType="java.util.Map" resultType="java.util.Map">

select

<include refid="Base\_Column\_List"/>

from project

<include refid="conditionByPower"></include>

order by name ASC

</select>

<!--快速查找，以Map返回-->

<select id="fastFind" parameterType="java.util.Map" resultType="java.util.Map">

select id,name

from project

<include refid="conditionByPower"></include>

order by name ASC

</select>

<!--查询，以对象集合返回-->

<select id="find" parameterType="java.util.Map" resultMap="BaseResultMap">

select a.\*,b.name as themeName from project a,theme b where a.theme\_id = b.id;

</select>

<select id="getRowsCount" parameterType="java.util.Map" resultType="java.lang.Integer">

select count(1) as total from project

<include refid="conditionByPower"></include>

</select>

<!--分页查找-->

<select id="findByPage" parameterType="java.util.Map" resultType="java.util.Map">

select

<include refid="Base\_Column\_List"/>

from project

<include refid="conditionByPower"></include>

order by name ASC

<if test="start !=0 or limit!=0">

limit #{start},#{limit}

</if>

</select>

<!--判断数据是否在使用中-->

<select id="objectInUse" parameterType="java.lang.String" resultType="java.lang.Boolean">

<![CDATA[ select count(id) from project where name=#{id} ]]>

</select>

</mapper>