# 目 录

1. [绪 论 1](#_bookmark0)
   1. [研究的背景及意义 1](#_bookmark1)
      1. [选题的背景 1](#_bookmark2)
      2. [国内外研究现状 1](#_bookmark3)
      3. [研究的意义 2](#_bookmark4)
   2. [系统目标 3](#_bookmark5)
2. [需求分析 5](#_bookmark6)
   1. [业务需求 5](#_bookmark7)
      1. [主要业务流程 5](#_bookmark8)
   2. [功能需求 8](#_bookmark9)
      1. [角色分析 8](#_bookmark10)
      2. [业务功能 8](#_bookmark11)
   3. [非功能需求 12](#_bookmark12)
      1. [环境需求 12](#_bookmark13)
      2. [性能需求 13](#_bookmark14)
      3. [安全需求 13](#_bookmark15)
3. [总体设计 14](#_bookmark16)
   1. [系统设计的原则 14](#_bookmark17)
   2. [系统体系结构设计 14](#_bookmark18)
   3. [系统功能结构设计 14](#_bookmark19)
4. [数据库设计 16](#_bookmark20)
   1. [概念结构设计 16](#_bookmark21)
      1. [设计思路 16](#_bookmark22)

[4.1.2 E-R 图 16](#_bookmark23)

* 1. [逻辑结构设计 18](#_bookmark24)
     1. [设计思路 18](#_bookmark25)
     2. [逻辑模型 18](#_bookmark26)
  2. [物理结构设计 20](#_bookmark27)
     1. [存取方式 20](#_bookmark28)
     2. [存储结构 20](#_bookmark29)

1. [界面设计 22](#_bookmark30)
   1. [界面关系图或工作流图 22](#_bookmark31)
   2. [界面设计成果 22](#_bookmark32)
      1. [主界面 22](#_bookmark33)
      2. [子界面 23](#_bookmark34)
2. [详细设计 29](#_bookmark35)

##### [系统主要功能模块介绍 29](#_bookmark36)

##### [项目问题管理模块 29](#_bookmark37)

##### [功能结构设计 29](#_bookmark38)

##### [类图设计 30](#_bookmark39)

##### [顺序图设计 31](#_bookmark40)

##### [核心处理流程设计 31](#_bookmark41)

##### [项目问题统计模块 32](#_bookmark42)

##### [功能结构设计 32](#_bookmark43)

##### [类图设计 32](#_bookmark44)

##### [顺序图设计 33](#_bookmark45)

##### [核心处理流程设计 33](#_bookmark46)

##### [前期工作统计模块 34](#_bookmark47)

##### [功能结构设计 34](#_bookmark48)

##### [类图设计 34](#_bookmark49)

##### [顺序图设计 35](#_bookmark50)

##### [核心处理流程设计 35](#_bookmark51)

##### [项目总体统计模块 36](#_bookmark52)

##### [功能结构设计 36](#_bookmark53)

##### [类图设计 36](#_bookmark54)

##### [顺序图设计 37](#_bookmark55)

##### [核心处理流程设计 38](#_bookmark56)

##### [问题督办统计模块 38](#_bookmark57)

##### [功能结构设计 38](#_bookmark58)

##### [类图设计 39](#_bookmark59)

##### [顺序图设计 39](#_bookmark60)

##### [核心处理流程设计 40](#_bookmark61)

[**7 编码** 41](#_bookmark62)

##### [代码实现与核心算法 41](#_bookmark63)

##### [项目问题管理模块核心代码 41](#_bookmark64)

##### [项目问题统计模块核心代码 42](#_bookmark65)

##### [前期工作统计模块核心代码 44](#_bookmark66)

##### [项目总体统计模块核心代码 45](#_bookmark67)

##### [代码优化分析 47](#_bookmark68)

[**8 测试** 48](#_bookmark69)

##### [测试方案设计 48](#_bookmark70)

##### [测试策略 48](#_bookmark71)

##### [测试进度安排 48](#_bookmark72)

##### [测试资源 48](#_bookmark73)

##### [关键测试点 48](#_bookmark74)

##### [测试用例构建 49](#_bookmark75)

##### [测试用例编写约定 49](#_bookmark76)

##### [测试用例设计 49](#_bookmark77)

##### [关键测试用例 49](#_bookmark78)

##### [测试用例维护 50](#_bookmark79)

1. [**总结与展望** 51](#_bookmark80)

##### [设计工作总结 51](#_bookmark81)

##### [未来工作展望 51](#_bookmark82)

[**谢 辞** 52](#_bookmark83)

[**参考文献** 53](#_bookmark84)

[**附录 A 外文翻译—原文部分** 55](#_bookmark85)

[**附录 B 外文翻译**—**译文部分** 60](#_bookmark86)

[**附录 C 软件使用说明书** 65](#_bookmark87)

[**附录 D 主要源代码** 68](#_bookmark88)

# 1 绪 论

#### 研究的背景及意义

##### 选题的背景

首先，中央政治局会议提出了明确全面做好“六稳”工作的要求，而政府对投资工作也提出要求要着重抓好“六个统筹”，再根据经济工作会议中关于“稳字当头、稳中求 进”的工作要求，应切实完成项目投资的规模、项目资金来源渠道和项目投资收益的统筹安排，以此加快推进平稳增长、促进有效改革、调整完善结构、普惠大众民生、预防未来风险、保持持续稳定的工作。所以应该着力使用重点项目为载体来有效实施重大战略，应该将重点项目作为平台来与国家对接、获得市上的支持，从而可以加强投资工作的整体性、系统性以及协同性，保证投资保持稳定增长，加快经济高质量发展。同时对于要统筹储备此地区预算内的投资、各类专项建设基金等各类政府投资项目，必须尽快推出地方政府投资项目纳入政府投资项目储备库，才能够有效发挥合理投资在稳定经济增长方面的决定性作用，这同时也是对投资项目管理有了新要求和明确了新方向。

##### 其次，依据国家发展改革委的全局规划，为了响应积极推进“五位一体”的国家总体布局和有效推进“四个全面”的国家战略布局，有效完成“七大战略行动计划”和“五大攻坚战”，引导和稳定预期，促进经济社会持续稳定发展，结合实际情况，打算依靠电子政务外网和互联网，来建设一个固定资产投资的项目在线服务监管平台，创新投资项目管理，形成以信息流和数据流为核心新型服务监管体系，提升发展改革委的项目服务监管水平和服务决策支撑能力。

再是目前，发改委已经使用了一批信息化系统，包括网审平台、固定资产投资项目在线服务监管平台、重庆市重点项目管理系统等，有效的加强了对政府投资项目的审批监管， 取得了显著的成效。但是同时也存在项目信息重复录入、缺乏对资源的调度配置和聚集、缺少数据统筹利用、缺乏对投资运行情况的综合分析和研判等问题。故急需建设统一的重大重点项目管理系统，与市级纵向、县级级横向相关系统大融合，实现所有项目进系统大汇集、大管理，基于大数据实现大决策，推动放管服工作上新台阶，为发改委转型晋级提供足够的信息化撑持。

##### 国内外研究现状

目前，“互联网+”能够将互联网的有效成果和经济社会的各个范畴进行深度融合， 可以有效推动技术的快速进步、多方位效率的明显提高和组织的积极革新，增强我国实体经济的创新力和生产力，更加构成了普遍的基于互联网的根底设施和创新因素的经济社会发展新形态。因为现在寰球正处在新一轮的科技革命及产业改革当中，所以互联网联合各范畴融合发展无疑是有着十分美好的前景和不可限量的潜力，如今它是不可抵挡的时代风潮，正在产生对全世界国家的经济社会发展的巨大影响。由此，在我国互联网已经占据了

1

##### 些许发展优势的情况下，要踊跃打造万众创新的氛围，在实现公共及服务方面，已经获得进步，对经济发展新常态的发展，我国起到一定的引领作用。逐渐形成经济发展的新能量， 对有效实现中国经济快速健康增长来说十分重要。

在我国的“互联网+”行动中，有着开放共享、融合创新、变革转型、引领跨越、安 全有序的基本原则，不仅有效发扬了我国互联网的规模和运用优势，而且保持将改革革新和市场需要作为导向，在互联网和经济社会等诸多领域交融中，鼎力拓宽其广度和深度。

##### 因为，“互联网＋”重要有两方面：第一个方面是在互联网技术和平台的基础上，传统行业积极开展自我创新，不断提高行业总体效率；第二个方面是在互联网的帮助下，不同产业进行融合，进而发展出新产品和新业态。

以此同时，美国和德国等发达国家的“互联网+”应用已经切实存在于他们的日常生活之中，甚至被用来研究和发展的，在作为他们创新战略的关键。美国利用“互联网+” 使得在之前的“万物互联”状态走上“万物智能”的状态。比如在你起床之前，厨房的咖啡已经自动准备好了；在你出门的时候，车子已经自动行驶到大门前面了；当家里的牛奶已经喝完了的时候，智能的冰箱就会为你自动续订牛奶…除了美国的发展，欧盟也确定好了 14 点行动计划，这个计划是为了加速物联网发展而制定的。与此同时，日本也积极投入到物联网的建设中去了，比如在 u-japan 计划当中，日本就把物联网设为四项重点战略领域之一。不仅如此，高科技农业的发展也是日本官方政府持续关注的焦点。“智能农 业”，就是他们利用机器人和信息技术的优秀成果，并且争取发展节省劳力的高质量农业。在澳大利亚，他们利用“互联网+”技术来搭建平台，用以在农业的信息服务，同时积极 开发诸多智能应用软件，方便农民使用并免费供给。在此基础之上，澳大利亚还开发了定位系统，可以全覆盖的。这样农户就能够通过利用 GPS 来精确耕作，同时对作物的布局进行合理的安排，在灌溉、灾害防治等方面也有许多积极作用。在新西兰，宽带基础设施一直是全力发展着的，为了能够将其用在覆盖郊区及偏远地区，而在生态农业旅游方面，同样也将信息化融入其中。

##### 研究的意义

首先，在互联网技术、互联网产业、互联网运用和跨界交融等方面，我国目前已经取得了积极进展，所以已经拥有坚实基础，可以加快推进“互联网+”发展。不过传统企业高效运用互联网的意识和能力还是不足，需要不断学习并提高。同时互联网企业对传统产业理解也进行得不够深刻，以及新业态发展在体制机制面前必然会遇到的阻碍、在交叉行业的融合型人才目前严重匮乏等问题是依然存在的。本次实现互联网+固定资产投资服务问题督办与统计分析监管平台，是一次积极的发展，在快速推进互联网与各个范畴深化融合方面，也是在稳定增长、促进改革、调整结构、普惠民生和预防风险方面，充沛地发挥“互联网+”的关键影响的一次积极尝试。

##### 其次，为了适应全球的“互联网+”发展趋向，并且在其中拔得头筹，那么我国目前

的互联网的诸多长处就一定要得到充分发挥，必须积极革新政府的办事形式，夯实网络发展根基，打造出安全可靠的网络环境，用以有效提高公共服务的总体水平。所以此次研发的互联网+固定资产投资服务问题督办与统计分析监管平台正是将企业事务办理与政府服务有力结合起来，使得政府各部门之间和企业的相关事项能够更加快速地得到解决。

##### 再是，设计与实现互联网+固定资产投资服务问题督办与统计分析监管平台是主动作为，在看待“互联网+”时，以长远发展的眼光来看待它，并且大力地在实践中去研究深入。彼此分享在“互联网+”有效融合运用方面的成功经验，加速“互联网+”新业态、新经济发展。在此同时，互联网+固定资产投资服务问题督办与统计分析监管平台，作为一个联系了现实的需要，进而打造了适合当地的“互联网+”行动落实计划，真正做到了因地制宜，能够推进“互联网+”行动的进程。

互联网+固定资产投资服务问题督办和统计分析监管平台使用“互联网+监管”工作， 合理的将互联网+技术应用在固定资产投资项的管理中,项目监管和统计的高效和精准化 程度，得到了很大的提高，使管理的工作效率和服务质量都能够得到进一步提升。

#### 系统目标

##### 对于此互联网+固定资产投资服务问题督办与统计分析监管平台来说，主要实现以下的系统目标，首先实现简洁的用户注册与登录界面，可供不同角色注册与登录，在登录后可进入该角色的系统首页界面。

在发改部门登录系统进入发改部门系统首页之后，在问题督办模块有问题待处理、问题已处理和问题结项三个功能界面。在问题待处理界面内，展示了行业主管部门与法人申请后的问题，发改部门可实现对这些问题的选择处理功能；在问题已处理界面中，展示了在问题待处理界面内选择了处理的问题，说明此问题已处理；在问题结项界面展示了行业主管部门与法人结项后的问题。发改部门在统计分析模块有项目总体统计、前期工作统计和问题督办统计三个功能界面。在项目总体统计界面中，使用图表展示了项目总数及类型、项目建设信息、项目投资情况；在前期工作统计界面中，使用图表展示了行业主管部门的前期工作信息情况；在问题督办统计界面中，使用图表展示了项目问题的情况。

##### 在法人登录系统进入法人系统首页之后，在问题督办模块有问题填报、问题申请、问题处理、问题结项四个功能界面。法人在问题填报界面处，可以进行项目问题的填报，其中项目问题按项目信息和项目调度类别来分类选择填报不同类型的项目问题，在此处需要填写项目问题的具体信息。法人填报完问题之后，可在问题申请处看到自己填报的项目问题，在此界面法人可点击申请按钮，则项目问题状态会由已填报改变为已申请。而已申请的项目问题会出现在发改部门的问题待处理界面，由发改部门处理后，在问题处理界面项目问题的状态会有已申请改编为已处理，则法人可将项目问题进行结项。在项目结项界面， 会展示法人结项的项目问题，在点击了项目名称时可以查看到项目问题的详细信息。法人

在统计分析模块有项目总体统计、项目问题统计两个功能界面。

在行业主管部门登录系统进入行业主管部门系统首页之后，在问题督办模块有问题填报、问题申请、问题处理、问题结项四个功能界面。行业主管部门在问题填报界面处，可以进行项目问题的填报，填写项目问题的具体信息。行业主管部门填报完问题之后，可在问题申请处看到自己填报的项目问题，在此界面行业主管部门可点击申请按钮，则项目问题状态会由已填报改变为已申请。而已申请的项目问题会出现在发改部门的问题待处理界面，由发改部门处理后，在问题处理界面项目问题的状态会有已申请改编为已处理，则行业主管部门可将项目问题进行结项。在项目结项界面，会展示行业主管部门结项的项目问题，在点击了项目名称时可以查看到项目问题的详细信息。行业主管部门在统计分析模块有前期工作统计功能界面。在前期工作统计界面中，使用图表展示了前期工作事项的情况。

# 需求分析

#### 业务需求

##### 本系统的业务需求描述如下：

1. 发改部门可以对法人和行业主管部门申请的问题进行处理，在处理之后反馈给法人和行业主管部门。发改部门可展示全部已处理问题，并按问题类型查看。发改部门还可以展示法人和行业主管部门的所有已结项的问题，并按问题类型查看。发改部门可对全部问题进行监控、督促。

##### 发改部门可以对项目总体情况、前期工作情况、项目问题情况进行统计分析。

1. 法人可以进行问题填报，包括项目信息与项目调度类型。填报之后选择申请至发改部门，再由发改部门处理反馈之后再进行问题结项。

##### 法人可以对项目总体情况、项目问题统计情况进行统计分析。

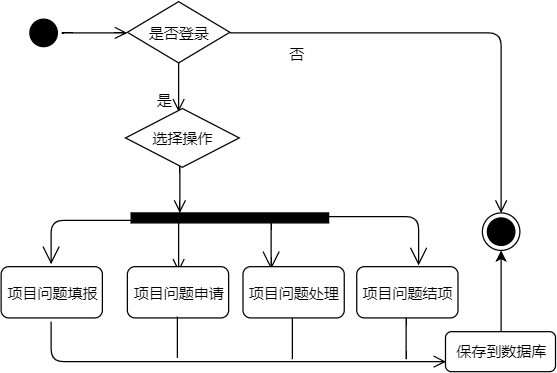
1. 行业主管部门可以填报前期工作问题，填报完成后申请至发改部门，由发改部门处理之后可将问题进行结项。

##### 行业主管部门可以对前期工作问题情况、前期工作信息情况进行统计分析。

##### 主要业务流程

##### 项目问题管理模块活动图

项目（法人）单位在登录后可以进行以下操作：项目问题填报、项目问题申请、项目问题结项。发改部门在登录后可以对项目问题进行处理。项目问题填报模块，其活动图如图 2-1 所示：

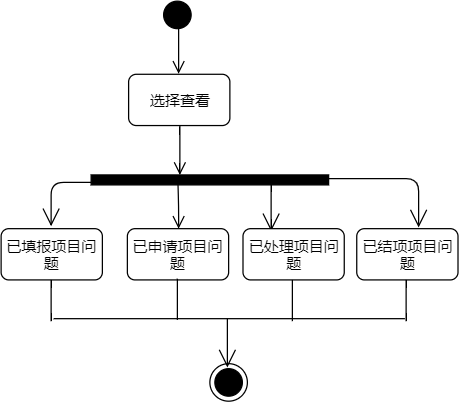


##### 项目问题统计模块活动图

图 2-1 项目问题管理模块活动图

##### 项目（法人）单位可以对已填报项目问题、已申请项目问题、已处理项目问题、已结

项项目问题进行查看，发改部门可以对已申请项目问题、已处理项目问题、已结项问题进行查看。项目问题统计模块，其活动图如图 2-2 所示：



##### 前期工作统计模块活动图

图 2-2 项目问题统计模块活动图

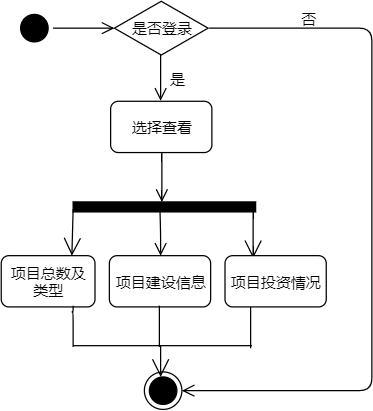
##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\T($D0A0VV[FK)%8XSW]BBDH.png行业主管部门登录后可以进行前期工作问题填报、前期工作问题申请、前期工作问题结项这些关于前期工作问题的操作，同时可以展示前期工作信息。发改部门可以进行前期工作问题处理的操作，同时可以对前期工作信息进行展示。前期工作统计模块活动图如下图 2-3 所示：

1. 项目总体统计模块活动图

图 2-3 前期工作统计模块活动图

##### 项目（法人）单位、发改部门登录后可以查看项目总数及项目类型、项目建设信息、

项目投资情况。项目总体统计模块，其活动图如图 2-4 所示：



##### 问题督办统计模块活动图

图 2-4 项目总体统计模块活动图

##### 发改部门登录后可以查看项目问题总数及类型统计、项目问题状态统计。问题督办统计模块，其活动图如图 2-5 所示：

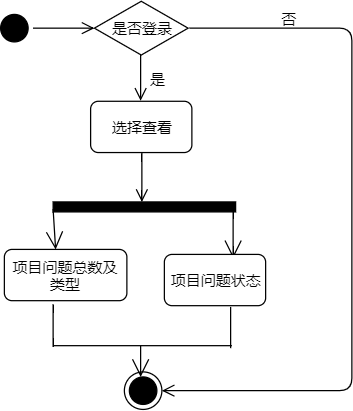


图 2-5 问题督办统计模块活动图

#### 功能需求

##### 角色分析

以下从互联网+固定资产投资服务问题督办与统计分析监管平台系统的实际需求分 析，系统涉及到的角色如下表 2-1 所示：

表 2-1 系统角色表

|  |  |
| --- | --- |
| **角色** | **职责或功能** |
| 发改部门 | 处理项目问题、展示已处理的项目问题、展示已结项的项目问题、统计分析项目总数情况、统计分析项目建设信息情况、统计分析项目投资情况、统计分析  前期工作情况、统计分析项目问题情况 |
| 法人 | 项目问题填报、项目问题申请、项目问题结项、统计分析项目总体情况 |
| 行业主管部门 | 前期工作项目问题填报、前期工作项目问题申请、前期工作项目问题结项、统  计分析前期工作、统计分析前期工作问题情况 |

##### 业务功能

通过对此系统的需求进行分析之后可以发现，互联网+固定资产投资服务问题督办与统计分析监管平台可以拆分为以下几个子模块系统：项目问题管理模块、项目问题统计模块、前期工作统计模块、项目总体统计模块、问题督办统计模块。系统的角色涉及到发改部门、项目（法人）单位和行业主管部门。在考虑不同角色在系统中要有不同的功能操作权限，设计了其功能需求总体用例图如下图 2-6 所示：

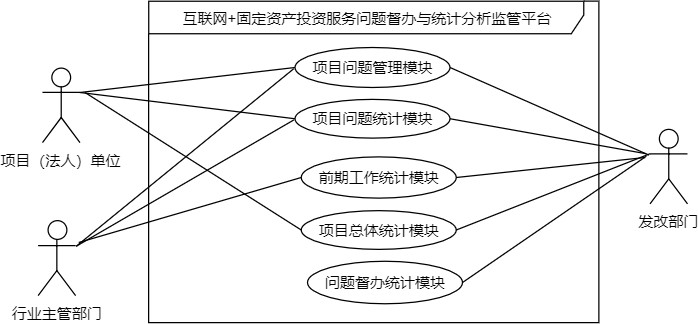


图 2-6 系统总体用例图

##### 项目问题管理模块用例图，如图 2-7 所示：

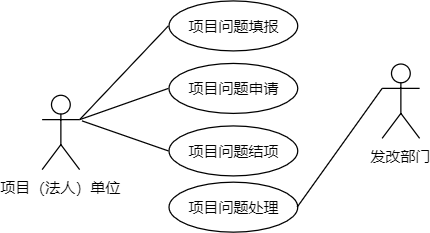


图 2-7 项目问题管理模块用例图

##### 此模块的项目问题填报用例分析表，如表 2-2 所示：

表 2-2 项目问题填报用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 项目问题填报 |
| 标识符\* | CF31 |
| 用例描述 | 描述项目（法人）单位填报项目信息问题的过程 |
| 参与者表 | 项目（法人）单位 |
| 优先级 | 1 |
| 前置条件 | 项目（法人）单位已经成功登录系统，进入项目（法人）单位系  统首页界面 |
| 后置条件 | 跳转进入问题申请页面 |
| 基本操作流 | 1. 项目（法人）单位输入邮箱与密码，点击登录； 2. 项目（法人）单位进入系统首页页面后点击问题督办下的问题填报； 3. 选择问题类型为项目信息，填写问题内容后点击提交。 |

##### 项目问题统计模块用例图如下图 2-8 所示：

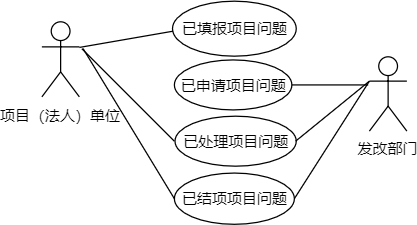


图 2-8 项目问题统计模块用例图

##### 此模块的已填报项目问题用例分析表如下表 2-3 所示：

表 2-3 已填报项目问题用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 已填报项目问题 |
| 标识符\* | CF14 |
| 用例描述 | 描述项目（法人）单位查看已填报项目问题的过程 |
| 参与者表 | 项目（法人）单位 |
| 优先级 | 1 |
| 前置条件 | 项目（法人）单位已经成功登录系统，进入首页界面 |
| 后置条件 | 相关操作 |
| 基本操作流 | 1. 项目（法人）单位输入邮箱与密码，点击登录； 2. 项目（法人）单位进入系统首页页面后点击问题督办的问题申请； 3. 在问题申请界面即可展示所有已填报项目问题 |

##### 前期工作统计模块用例图如下图 2-9 所示：

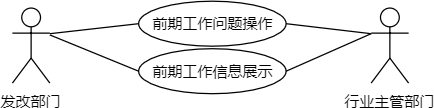


图 2-9 前期工作统计模块用例图

##### 此模块的前期工作信息展示用例分析表如下表 2-4 所示：

表 2-4 前期工作信息展示用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 前期工作信息展示 |
| 标识符\* | CF14 |
| 用例描述 | 描述发改部门、行业主管部门处理项目问题的过程 |
| 参与者表 | 发改部门、行业主管部门 |
| 优先级 | 1 |
| 前置条件 | 发改部门、行业主管部门已经成功登录系统，分别进入发改部门、  行业主管部门系统首页界面 |
| 后置条件 | 相关操作 |
| 基本操作流 | 1. 发改部门、行业主管部门输入邮箱与密码，点击登录； 2. 发改部门、行业主管部门进入系统首页页面后选中统计分析下的前期工作统计界面，即展示前期工作相关信息。 |

##### 项目总体统计模块用例图如下图 2-10 所示：

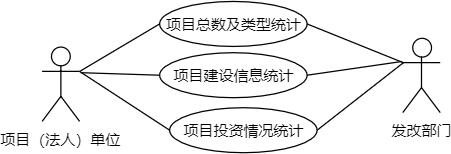


图 2-10 项目总体统计模块用例图

##### 此模块的项目总数及类型统计用例分析表如下表 2-5 所示：

表 2-5 项目总数及类型统计用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 项目总数及类型统计 |
| 标识符\* | CF14 |
| 用例描述 | 描述发改部门、项目（法人）单位对项目具体情况进行统计的过  程 |
| 参与者表 | 发改部门 |
| 优先级 | 1 |
| 前置条件 | 发改部门、项目（法人）单位已经成功登录系统，分别进入发改  部门、项目（法人）单位系统首页界面 |
| 后置条件 | 相关操作 |
| 基本操作流 | 1. 发改部门、项目（法人）单位输入邮箱与密码，点击登录； 2. 发改部门、项目（法人）单位进入系统首页后点击进入统计分析的项目总体统计页面； 3. 发改部门在项目总体统计可查看项目总数及类型统计。 |

##### 问题督办统计模块用例图如下图 2-11 所示：

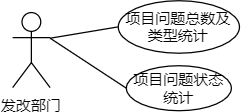


图 2-11 问题督办统计模块用例图

##### 此模块的项目问题状态统计用例分析表如下表 2-6 所示：

表 2-6 项目问题状态统计用例描述

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 用例名称 | 项目问题状态统计 |

续表 2-6

|  |  |
| --- | --- |
| 描述项 | 说明 |
| 标识符\* | CF14 |
| 用例描述 | 描述发改部门将项目问题状态情况进行统计的过程 |
| 参与者表 | 发改部门 |
| 优先级 | 1 |
| 前置条件 | 发改部门已经成功登录系统，进入系统首页界面 |
| 后置条件 | 后续相关操作 |
| 基本操作流 | 1. 发改部门输入邮箱与密码，点击登录； 2. 发改部门进入系统首页后点击进入统计分析的问题督办统计页面； 3. 发改部门在问题督办统计页面可查看项目问题状态相关统计数   据。 |

#### 非功能需求

##### 环境需求

此系统是基于 Java 技术与 Spring Boot 框架进行开发的，此系统能够在 Windows 操作系统上进行部署并运行。与此同时，此系统还是使用开源框架和技术实现的，在运行环境的配置要求上并不高，对于系统的后期维护，也是不需要更多的人力成本的，所以此系统有很好的经济可行性。

##### 此系统时基于 B/S 架构，由服务器端和客户端构成的，所以不同部分的具体运行环境如下表 2-7 所示：

表 2-7 运行环境配置要求表

|  |  |
| --- | --- |
| 服务器端运行环境 | |
| 项目 | 配置要求 |
| 系统所使用数据库 | Microsoft SQL Server 2008 SP1 |
| 操作系统 | Windows XP 及以上版本或Linux 或 MacOS |
| 应用服务器 | Apache Tomcat 8.5.16 |
| JDK | 1.8 |
| 开发框架 | Springboot 1.5.6、SpringMVC、Mybatis、BootStrap 等 |
| 客户端运行环境 | |
| 项目 | 配置要求 |
| 操作系统 | Windows XP 及以上版本或 Linux 或 MacOS |

##### 续表 2-7

|  |  |
| --- | --- |
| 客户端运行环境 | |
| 浏览器 | IE6 以上版本或火狐、谷歌浏览器 |
| 分辨率 | 最佳分辨率 1920🞨1080 像素 |

##### 性能需求

##### 为保证本系统能够长期保持正常稳定的运行状态，在性能需求方面应该满足以下几点：

1. 处理能力

##### 系统的处理能力重点考虑系统能够承载的最大并发数，需要按照实际情况进行规划。

1. 扩展能力

##### 在对系统进行设计时，适当考虑到之后会添加的功能模块，留下扩展接口，使系统具有持续扩展功能、适度变化的能力。

1. 响应时间

##### 为了使用户能有效快速地进行查询的服务，系统应该能够有效地响应查询请求。

##### 安全需求

##### 数据传输安全

传输的数据会采用加密算法加密，让数据即使在被泄露或者截获之后，其数据内容都无法被识别，有效确保数据安全。

##### 系统容错性

针对系统中的用户，在其出现错误时能够及时制止其错误的行为，会出现信息提示， 同时对已出现的错误能够有效追踪出错原因。

##### 网络安全

针对系统所部署的外部环境，有相应的网络安全防护措施，让操作系统和应用系统都能够相对处在安全良好的网络环境下，有效降低遭受外部网络威胁的概率。

# 总体设计

#### 系统设计的原则

##### 在充分考虑系统总体规划目标和系统应用环境之后，本系统设计将严格遵循以下的设计原则：

1. 适合性原则。在系统设计时，要尽力满足客户需求，设计出满足客户需求的系统，使得开发方与客户方都能利益最大化，而不需要一味追求最先进的系统，不必过度设计， 要能够懂得抓住重点。

##### 灵活性原则。鉴于如今软件需求变化及其迅速，本系统设计时应采用模块化结构设计， 对于后期系统进行改进、扩充有较大的帮助，并且系统可以处于不断完善的过程中， 不断扩充和完善。

1. 用户友好性原则。系统设计应该以人为本，要有良好的以用户为中心的交互设计。在研究清楚角色模型，数据的整合与呈现方式，用户行为与上下文方面多花点心思，使得新手用户可以无障碍的较快学会该系统的使用。
2. 可维护性原则。可维护性直接体现在系统结构的良好性和文档体系的完整性。能够在文档的基础上进行维护，在系统结构处出发，将重构做为核心，最大限度地重用现存系统。

#### 系统体系结构设计

在此系统设计中，运用的是 B/S 结构。除了使用了 b/s 架构，本系统还使用了 Spring Boot 框架 。因为 Spring Boot 框架会创建独立的 Spring 应用，并且也不用部署 WAR 文件，同时使用 starter 依赖可使配置构建与管理依赖变得简单化，一定条件下能够自动装配第三方类库， 也不必进行 XML 配置，所以 Spring Boot 的诸多优势极大地简化了开发，提高了开发效率。

开发中，本系统还使用到了 ECharts 这一使用 javaScript 实现的开源可视化库。ECharts 可以提供常规的折线图、柱状图、饼图等用于统计的盒形图，也提供地理数据可视化的地图等用于关系数据可视化的关系图，它也支持图与图之间的混搭。使用 Echart 技术可以展示直观，交互丰富的数据化图表。

#### 系统功能结构设计

##### 本系统一共分为五个模块，包括项目问题管理模块、项目问题统计模块、前期工作统计模块、项目总体统计模块、问题督办统计模块。项目问题管理模块又分为项目问题填报、项目问题申请、项目问题处理、项目问题结项四个子模块；本系统重要模块有项目问题管理模块、项目总体统计模块和问题督办统计模块。项目总体统计模块又分为项目总数及类型统计、项目建设信息统计、项目投资情况统计三个子模块；问题督办统计模块又分为项目问题总数及类型统计、项目问题状态统计两个子模块。

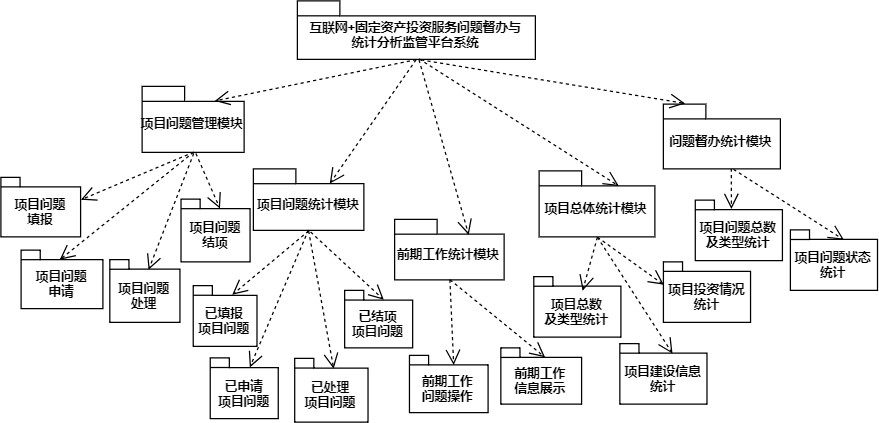
本系统的系统总包图，如图 3-1 所示：

图 3-1 系统总包图

# 数据库设计

#### 概念结构设计

##### 设计思路

因为概念设计是从现实生活中抽象出切实的实体，在明确实体的属性的时候也应该要确定实体之间的关系，由此得到数据库最原始的数据形式，并且可得到用户视图。在进行数据库的概念设计时，我们常常会采用 E-R 方法，由部分再到整体，最后进行优化，就能够得到我们需要的 E-R 模式，也就是概念模式。

##### E-R 图

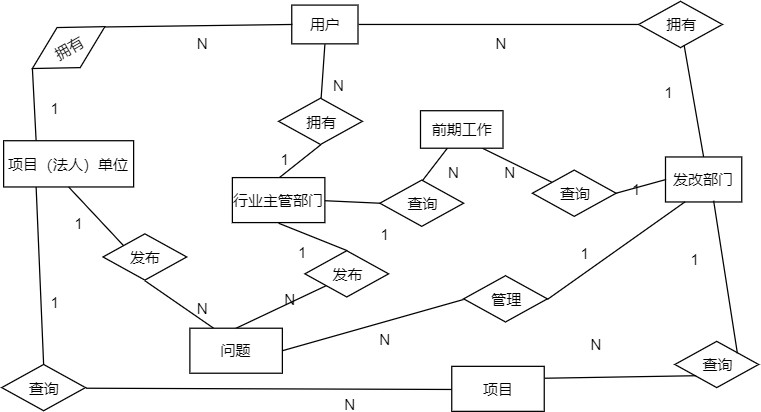
1. 本系统实体关系图，如下图 4-1 所示。

图 4-1 系统实体关系图

##### 用户实体图，如下图 4-2 所示。

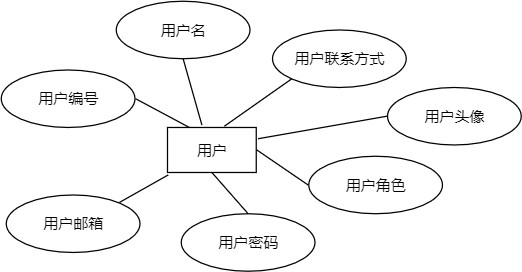


图 4-2 用户实体图

##### 问题的实体图，如图 4-3 所示。

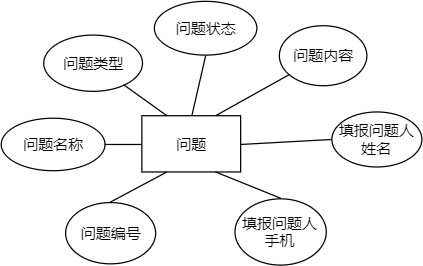


图 4-3 问题实体图

##### 前期工作实体图，如图 4-4 所示。

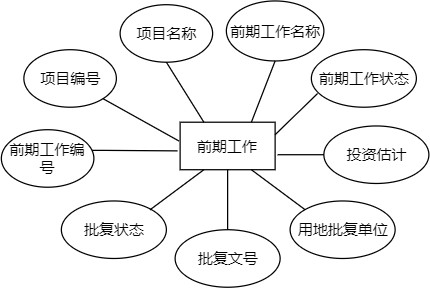


图 4-4 前期工作实体图

##### 项目实体图，如图 4-5 所示。

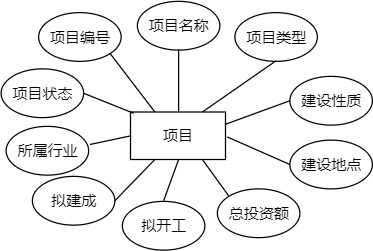


图 4-5 项目实体图

#### 逻辑结构设计

##### 设计思路

本系统的数据库逻辑结构设计过程主要有以下三个步骤：

##### 先将之前设计好了的概念结构转化为一般的关系、网状、层次模型；

1. 再将上一步转换而来的关系、网状、层次模型转化为特定 DBMS 支持下的数据模型；

##### 对数据模型进行优化。

##### 逻辑模型

##### 由关系 E-R 图，可以得到的关系模型如下所示：

1. 用户表：（用户编号、用户名、用户联系方式、用户头像、用户角色、用户邮箱、用户密码）

##### 问题表：（问题编号、问题名称、问题类型、问题状态、问题内容、填报问题人姓名、填报问题人手机）

1. 前期工作表：（前期工作编号、项目编号、项目名称、前期工作名称、前期工作状态、投资估计、用地批复单位、批复文号、批复状态）

##### 项目表：（项目编号、项目状态、项目名称、项目类型、建设性质、建设地点、总投资额、拟开工、拟建成、所属行业）

根据以上的关系模型最终设计了 6 张数据表来处理本文所涉及的模块功能，如下表 4-1

##### 所示：

表 4-1 系统涉及数据表

|  |  |
| --- | --- |
| 表名 | 功能说明 |
| user | 用户表 |
| ques | 问题表 |
| proinfo | 前期工作表 |
| projectinfo | 项目表 |

##### 用户表

用户表中字段包括用户编号、用户名、用户联系方式、用户头像、用户角色、用户邮箱、用户密码。用户表的结构设计，如表 4-2 所示：

表 4-2 用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 备注 |
| id | 整型 | 20 | 主键 | 用户编号 |
| username | 字符串型 | 255 | 否 | 用户名 |
| password | 字符串型 | 255 | 否 | 密码(加密后) |
| tel | 字符串型 | 255 | 否 | 用户联系方式 |
| email | 字符串型 | 255 | 否 | 用户邮箱 |

续表 4-2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 备注 |
| role | 字符串型 | 255 | 否 | 用户邮箱 |
| headImg | 字符串型 | 255 | 否 | 用户头像 |

##### 问题表

问题表中字段包括问题编号、问题名称、问题状态、问题内容、填报问题人姓名、填报问题人手机。问题表的结构设计，如表 4-3 所示：

表 4-3 问题表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 备注 |
| qid | 整型 | 11 | 主键 | 问题编号 |
| qname | 字符串型 | 50 | 否 | 问题名称 |
| qstatus | 字符串型 | 50 | 否 | 问题状态 |
| qtype | 字符串型 | 50 | 否 | 问题类型 |
| qcontent | 字符串型 | 50 | 否 | 问题内容 |
| qperson | 字符串型 | 50 | 否 | 填报问题人姓名 |
| qtel | 字符串型 | 50 | 否 | 填报问题人手机 |

##### 前期工作表

前期工作问题表中字段包括前期工作编号、项目编号、项目名称、前期工作名称、前期工作状态、投资估计、用地批复单位、批复文号、批复状态。前期工作表的结构设计， 如表 4-4 所示：

表 4-4 前期工作表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 备注 |
| proid | 整型 | 11 | 主键 | 前期工作编号 |
| pid | 整型 | 50 | 否 | 项目编号 |
| projectname | 字符串型 | 50 | 否 | 项目名称 |
| proname | 字符串型 | 50 | 否 | 前期工作名称 |
| prostatus | 字符串型 | 50 | 否 | 前期工作状态 |
| investabout | 字符串型 | 255 | 否 | 投资估计 |
| pioland | 字符串型 | 50 | 否 | 用地批复单位 |
| pifnland | 字符串型 | 50 | 否 | 批复文号 |
| psland | 字符串型 | 50 | 否 | 批复状态 |

##### 项目表

项目表中字段包括项目编号、项目状态、项目名称、项目类型、建设性质、建设地点、总投资额、拟开工、拟建成、所属行业。项目表的结构设计，如表 4-5 所示：

表 4-5 项目表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 备注 |
| pid | 整型 | 50 | 主键 | 项目编号 |
| projectstatus | 字符串型 | 255 | 否 | 项目状态 |

续表 4-5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键否 | 备注 |
| projectname | 字符串型 | 50 | 否 | 项目名称 |
| projecttype | 字符串型 | 255 | 否 | 项目类型 |
| buildnature | 字符串型 | 255 | 否 | 建设性质 |
| buildplacce | 字符串型 | 255 | 否 | 建设地点 |
| investmenttotal | 字符串型 | 255 | 否 | 总投资额 |
| nstarttime | date | 255 | 否 | 拟开工 |
| nendtime | date | 255 | 否 | 拟建成 |
| industry | 字符串型 | 255 | 否 | 所属行业 |

#### 物理结构设计

##### 存取方式

现在的 DBMS 可以提供许多存取方法，例如索引法以及 HASH 法。在本文用的是索引法。因为在数据库中，利用索引可以使得系统速度查询到表中的数据，能够极大地减少数据的查询时间，提高系统效率和用户体验感。

##### 在数据库关系图中为表定义主键将自动创建主键[索引](https://baike.baidu.com/item/%E7%B4%A2%E5%BC%95)，用户表中设置用户编号为索引，项目信息问题表设置项目信息问题编号作为索引，项目调度问题表设置项目调度问题编号作为索引，前期工作问题表设置前期工作问题编号作为索引，前期工作表中设置前期工作编号为索引，项目表中设置项目编号为索引。

##### 存储结构

##### 由于在确定数据的存放位置与存储结构的时候，我们都应该同时想到存取的时间、存储空间的利用率以及维护代价，这三方面因素对系统的影响。要提高系统的性能，我们应该依据应用情况将数据的易变部分、稳定部分与经常存取的部分和存取频率较低的部分分开存储。

根据逻辑结构设计得到的结果，将逻辑模型转为 MySQL 数据库表，设计对应的物理模型表，具体化数据类型。具体信息如下所示：

##### 用户表中用户编号、用户名、用户联系方式、用户头像、用户角色、用户邮箱、用户密码的数据类型可依次确定为 int, varchar, varchar, varchar, varchar，varchar, varchar；

1. 其中问题表中的问题编号、问题名称、问题状态、问题类型、问题内容、填报问题人姓名、填报问题人手机的数据类型可依次确定为 int, varchar, varchar, varchar, varchar, varchar，varchar；

##### 其中前期工作表中的前期工作编号、项目编号、项目名称、前期工作名称、前期工作状态、投资估计、用地批复单位、批复文号、批复状态的数据类型可依次确定为 int, int, varchar, varchar, varchar, varchar, varchar, varchar, varchar；

1. 其中项目表中的项目编号、项目状态、项目名称、项目类型、建设性质、建设地点、总投资额、拟开工、拟建成、所属行业的数据类型可依次确定为 int, varchar, varchar, varchar, varchar, varchar, varchar，date，date，varchar；

# 界面设计

#### 界面关系图或工作流图

##### IE1F2`U4OBUU`81QT)I}0[Q对本系统的界面进行设计中，设计使用左导航栏便于用户选择不同界面交互。各个角色的界面关系图如下图 5-1 所示：

图 5-1 系统界面关系图

#### 界面设计成果

##### 主界面

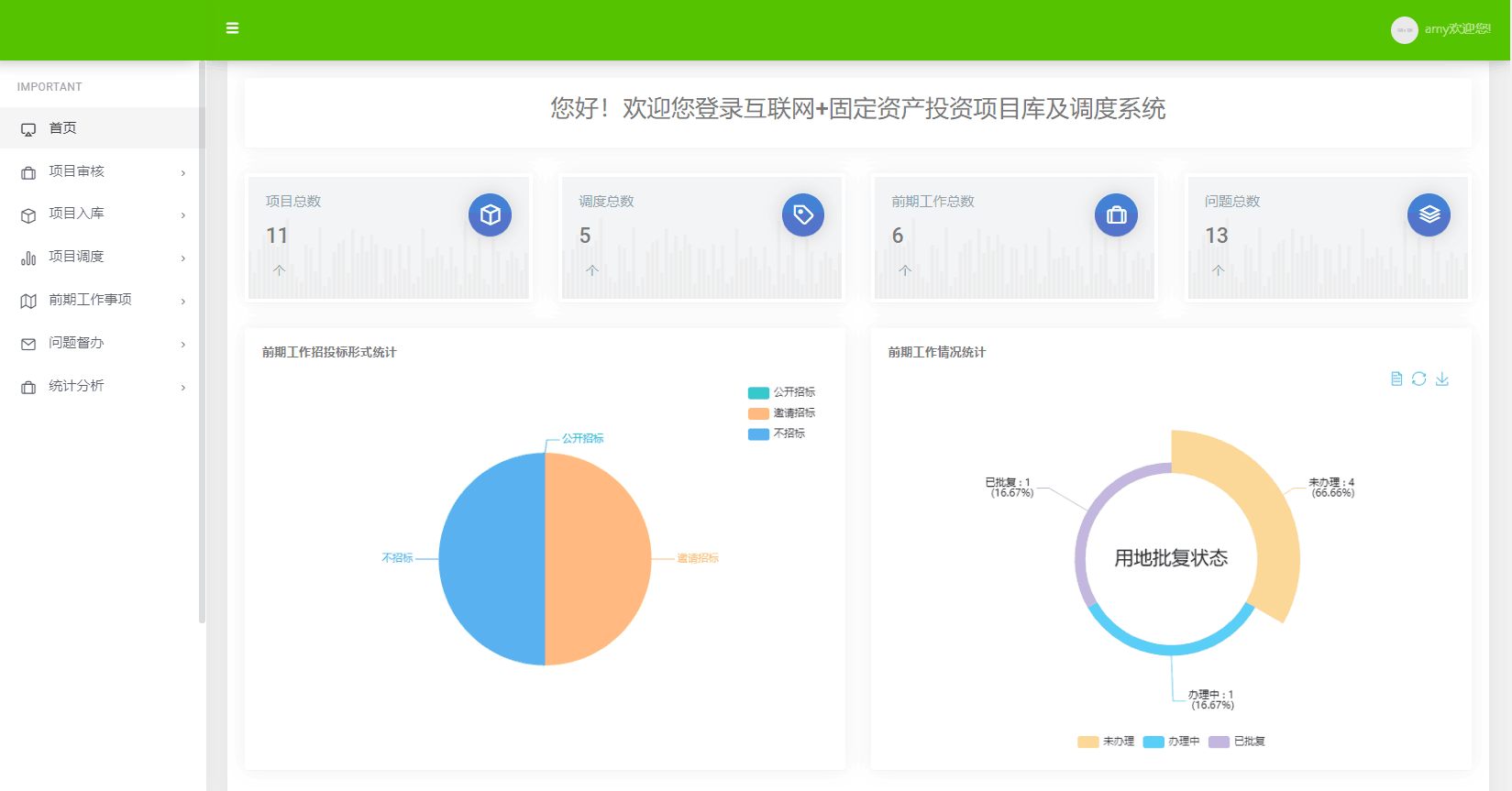
1. 发改部门主界面。发改部门登录系统后，默认进入发改部门主界面，如下图 5-2 所示：

图 5-2 发改部门主界面图

##### 项目（法人）单位首页页面。项目（法人）单位登录系统后，系统默认进入项目（法

人）单位主界面，如下图 5-3 所示：



图 5-3 项目（法人）单位主界面图

##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\P%2)13MR]$K]WTMH3S$MJ}2.png行业主管部门首页页面。行业主管部门登录系统后，系统默认进入行业主管部门主界面，如下图 5-4 所示：

##### 子界面

图 5-4 行业主管部门主界面图

##### 注册界面。该页面是系统的注册页面，用户可以在此界面进行注册。在用户进行注册的时候需要填写完整邮箱、用户名和密码信息，同时还需要选择自己对应的角色，角色包括项目（法人）单位、发改部门、各行业主管部门。填写好注册信息后即可点击

注册，如下图 5-5 所示：

图 5-5 登录注册界面图

##### 登录界面。该页面是登录界面，用户注册成功后可在此登录界面进行登录。用户在输入框中输入自己的邮箱及密码后，点击登录按钮选择登录系统，如下图 5-6 所示：



图 5-6 登录注册界面图

##### 项目（法人）单位问题填报界面。该页面是项目（法人）单位登录系统后，选择问题督办下的问题填报，进行项目填报的界面，如下图 5-7 所示：

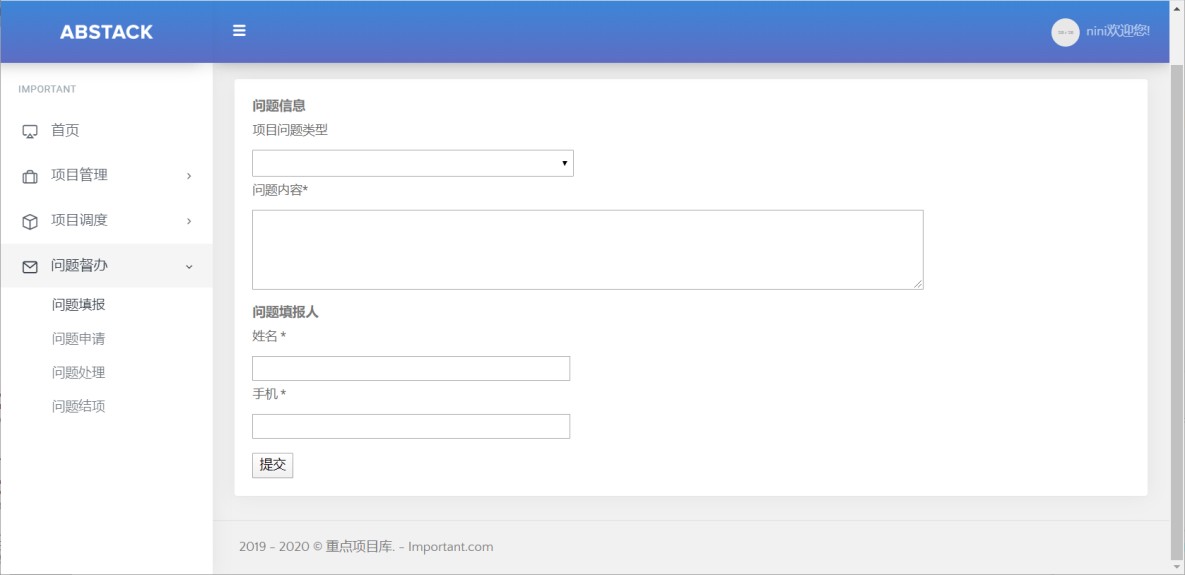


图 5-7 法人问题填报页面

##### 行业主管部门问题处理界面。该页面是行业主管部门登录系统后，选择已填报的项目问题申请至发改部门进行处理后,展示已处理项目问题的界面，如下图 5-8 所示：

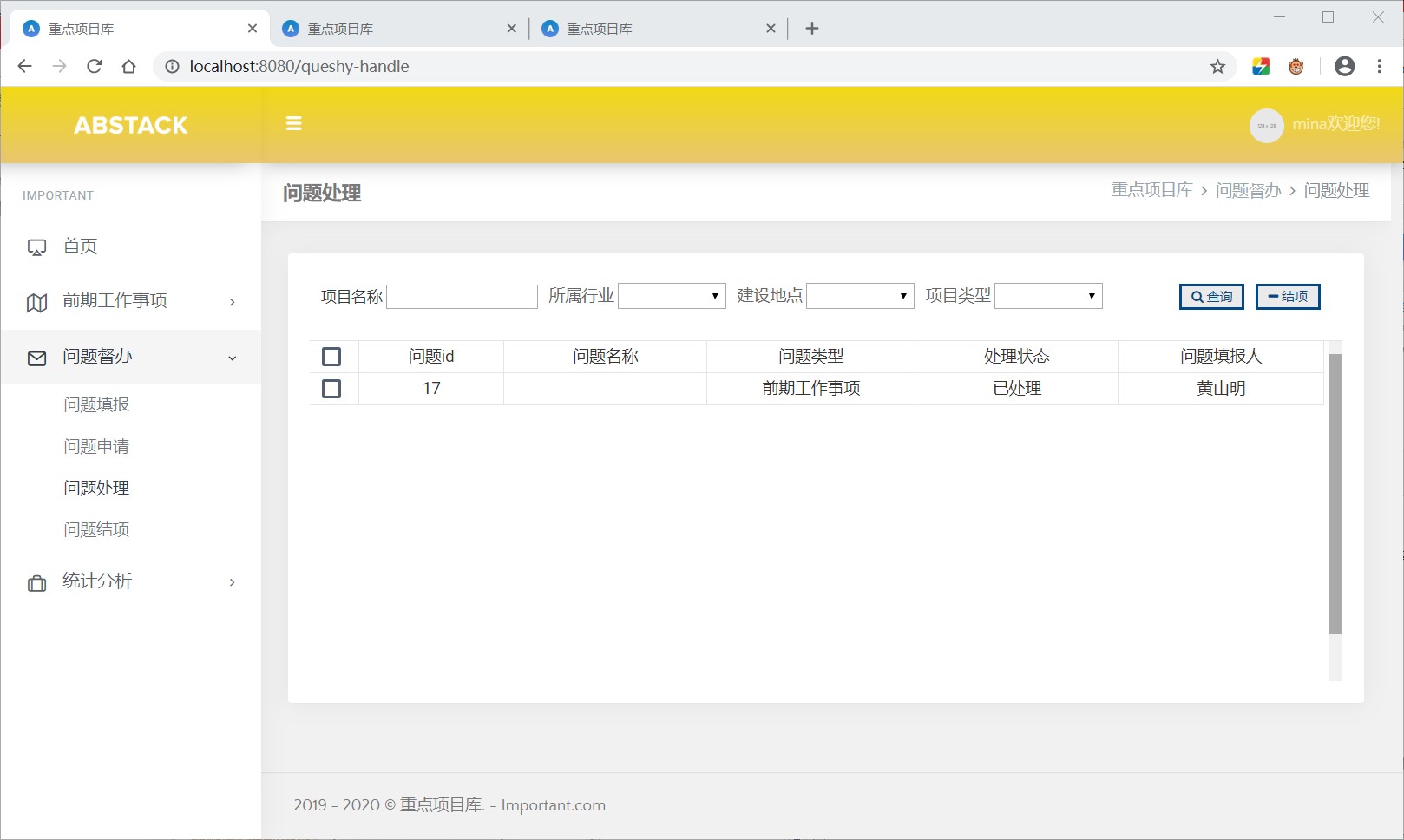


图 5-8 行业主管部门问题处理页面

##### 行业主管部门问题结项界面。该页面是行业主管部门登录系统后，选择已处理的项目问题进行结项的界面，如下图 5-9 所示：

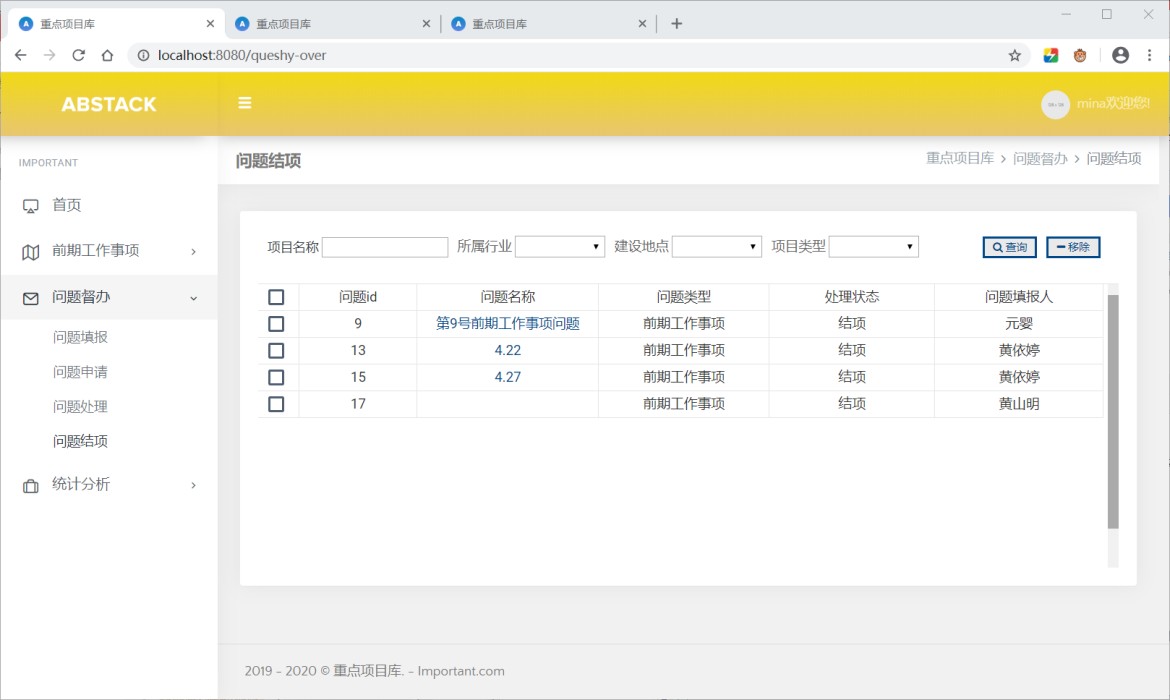


图 5-9 行业主管部门问题结项页面

##### 发改部门问题待处理界面。该页面是发改部门登录系统后，对项目（法人）单位和行业主管部门申请后的项目问题进行处理的界面，如下图 5-10 所示：

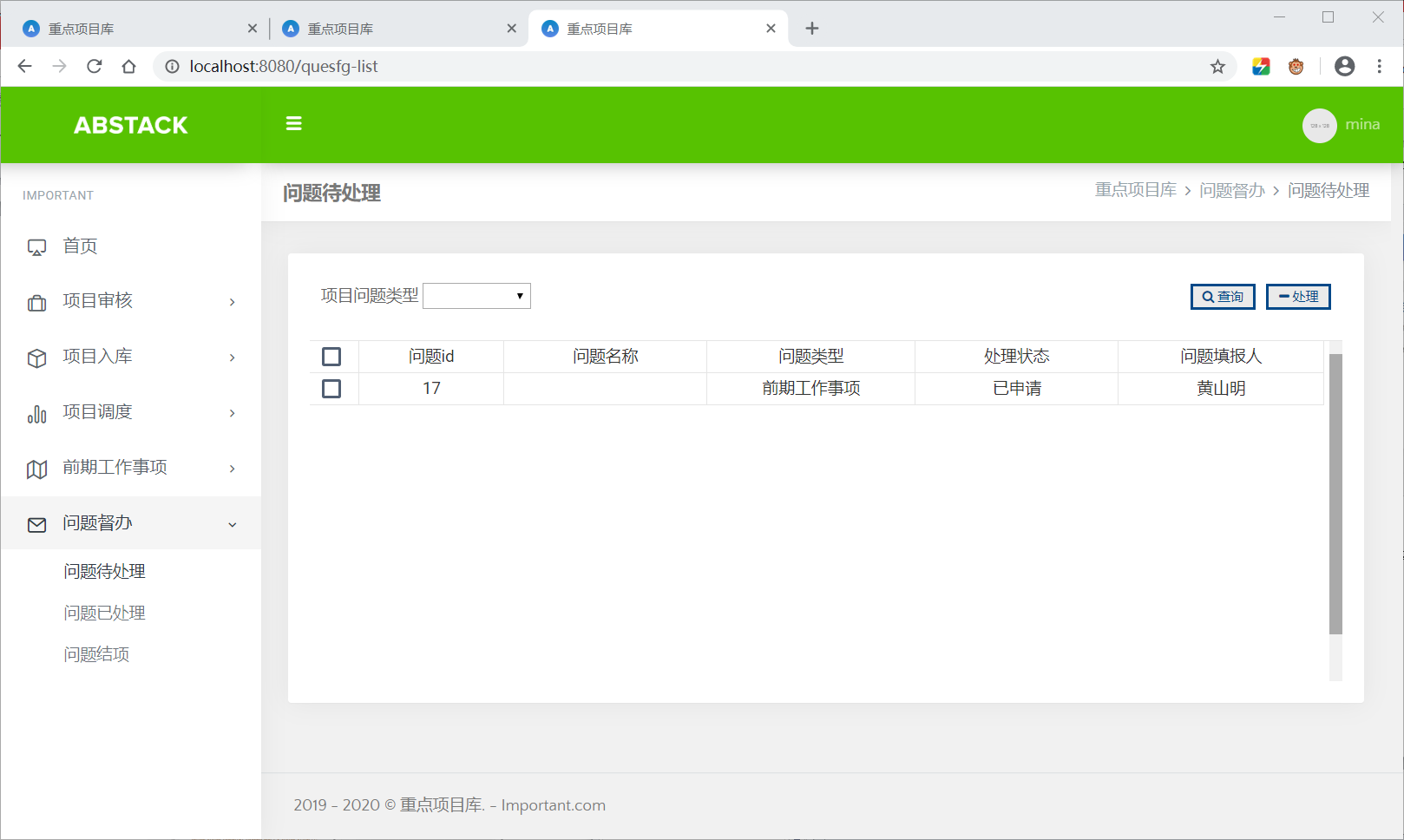


图 5-10 发改部门问题待处理页面

##### 发改部门已处理界面。该页面是发改部门登录系统后，对项目（法人）单位和行业主管部门申请后的项目问题处理后展示的界面，如下图 5-11 所示：

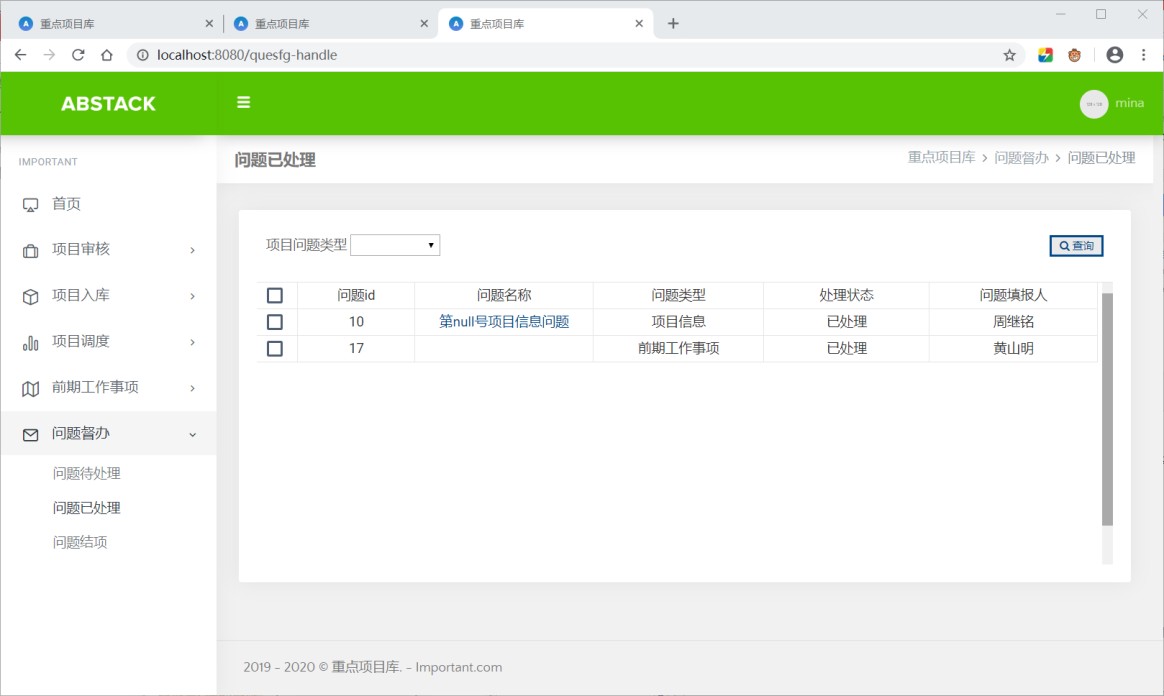


图 5-11 发改部门问题已处理页面

##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\$GHM_J6M5}2V_%8XL355QBU.png发改部门结项界面。该页面是发改部门登录系统后，对所有项目（法人）单位、行业主管部门已结项的项目问题进行展示的界面，如下图 5-12 所示：

图 5-12 发改部门问题结项页面

##### 发改部门查看项目总体统计页面。该页面是发改部门登录系统后，在统计分析下的项目总体统计下展示了项目建设地点分布情况，如下图 5-13 所示：

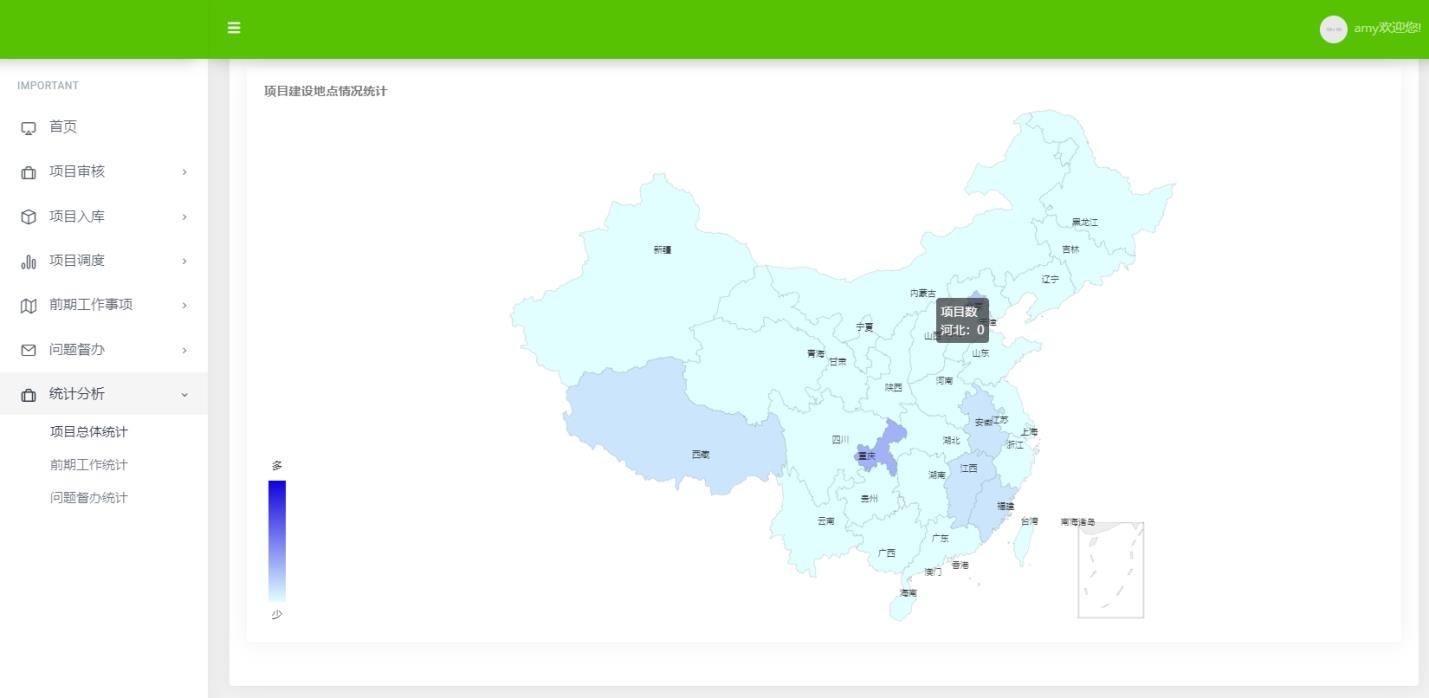


图 5-13 发改部门建设地点界面

##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\MIW0Z`%_7M0MHN%A54DBDL6.png项目（法人）单位查看项目问题统计界面。该页面是项目（法人）单位登录系统后， 在统计分析下的项目问题统计下展示了项目问题情况，如下图 5-14 所示：

图 5-14 项目(法人)单位项目问题统计界面

##### 

# 详细设计

#### 系统主要功能模块介绍

##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\W~(OY]CJXS8C`TJ9E}1C5YL.png本系统是由项目问题管理模块、项目问题统计模块、前期工作统计模块、项目总体统计模块、问题督办统计模块共五个模块，本系统的模块结构图，如下图 6-1 所示：

图 6-1 系统模块结构图

##### 每个模块的功能说明表如下表 6-1 所示：

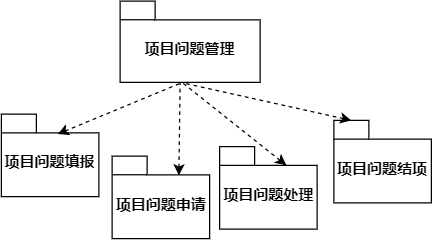
表 6-1 各模块功能说明表

|  |  |
| --- | --- |
| 模块 | 功能 |
| 项目问题管理模块 | 对项目问题进行填报、申请、处理、结项的操作和管理 |
| 项目问题统计模块 | 对项目问题已填报、已申请、已处理、已结项的操作和管理 |
| 前期工作统计模块 | 对前期工作问题的操作管理和前期工作的统计展示 |
| 项目总体统计模块 | 对项目数目、类型、建设信息、投资情况进行统计展示 |
| 问题督办统计模块 | 对项目问题数目、类型、状态的统计展示 |

#### 项目问题管理模块

##### 功能结构设计

项目问题管理模块的功能是完成项目（法人）单位对项目问题填报、项目问题申请、项目问题结项的操作，以及发改部门对项目问题处理的操作。本模块由项目信息问题填报、项目问题申请、项目问题处理、项目问题结项四个子模块组成。项目问题管理模块的包图如下图 6-2 所示：



##### 类图设计

图 6-2 项目问题管理模块包图

##### 在项目问题管理模块类图中，主要涉及五个类的关联，其中问题控制器类继承基本控制器类，依赖于问题服务层类和问题类，问题服务层类依赖于问题数据库操作类和回答类， 问题数据库操作类依赖于问题实体类。项目问题管理模块的类图如下图 6-3 所示：

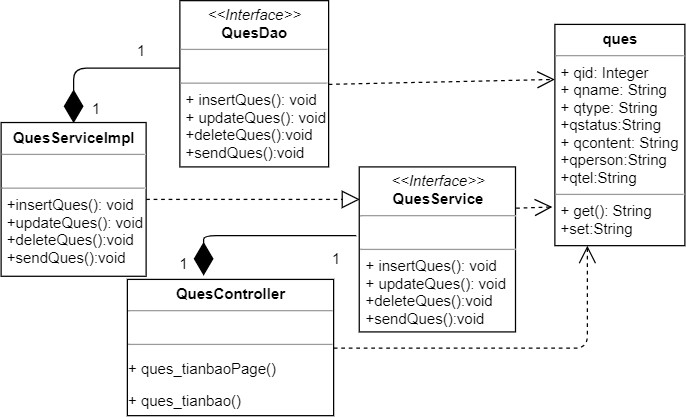


图 6-3 项目问题管理模块类图

##### 项目问题填报模块涉及的各个类的描述如下表所示，其中问题控制器类描述如下表 6-2

所示：

表 6-2 问题控制类描述

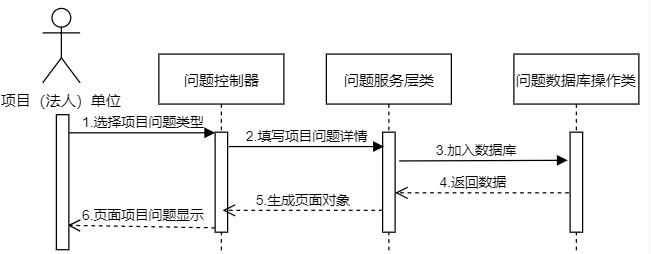
|  |  |  |
| --- | --- | --- |
| 问题控制类 | | |
| 类的描述：问题填报的控制类，负责请求与转发 | | |
| 操作描述 | ques\_tianbaoPage() | 项目填报 |
| ques\_tianbao() | 项目填报后提交 |

##### 问题实体类描述表如下表 6-3 所示：

表 6-3 项目信息问题实体类描述

|  |  |  |
| --- | --- | --- |
| 问题实体类 | | |
| 类的描述：问题的实体对象类 | | |
| 属性描述 | qid | 问题编号 |
| qname | 问题名称 |
| Qtype | 问题类型 |
| Qstatus | 问题状态 |
| qcontent | 问题内容 |
| qperson | 问题填报人姓名 |
| qptel | 问题填报人手机 |
| 操作描述 | setter/getter() | set 和 get 方法 |

##### 顺序图设计

在项目问题填报的时序图中，角色在进行填报时，会首先调用服务层类和数据库操作类进行新增操作的处理，将新增的信息封装到页面对象，并最终显示至填报完成界面。项目问题填报子模块的时序图如下图 6-4 所示：

##### 核心处理流程设计

图 6-4 项目问题填报时序图

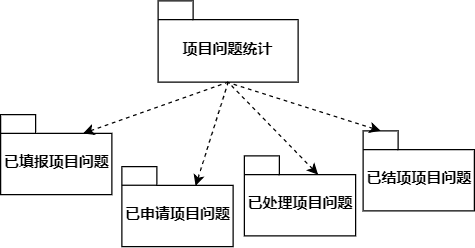
##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\H`FK[`XD8@04A1W1X(QB4JU.png项目问题管理模块主要活动图如下图 6-5 所示：

图 6-5 项目问题管理模块活动图

#### 项目问题统计模块

##### 功能结构设计

项目问题统计模块的主要功能是项目（法人）单位可以查看已填报项目问题、已申请项目问题、已处理项目问题、已结项项目问题。项目问题统计模块包图如下图 6-6 所示：



##### 类图设计

图 6-6 项目问题统计模块包图

##### 在项目问题统计模块类图中，主要涉及五个类的关联，其中问题控制器类继承基本控制器类，依赖于问题服务层类和问题类，问题服务层类依赖于问题数据库操作类和回答类， 问题数据库操作类依赖于问题实体类。

项目问题统计模块的类图如下图 6-7 所示：

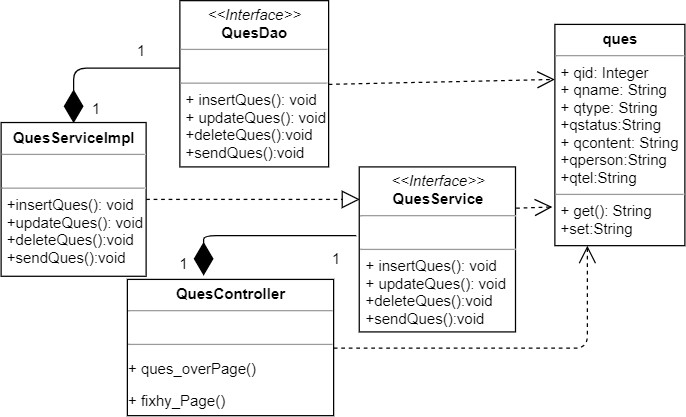


图 6-7 项目问题统计模块类图

##### 问题控制层类描述表如下表 6-4 所示：

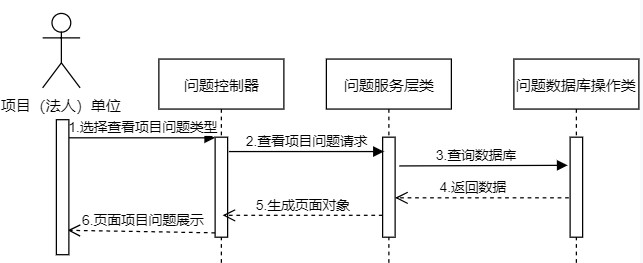
表 6-4 问题控制层类描述

|  |  |  |
| --- | --- | --- |
| 问题控制类 | | |
| 类的描述：问题的控制类，负责请求与转发 | | |
| 操作描述 | ques\_overPage() | 问题办理清单 |
| fixhy\_Page（） | 问题展示 |

##### 顺序图设计

在项目问题统计的时序图中，项目（法人）单位在选择查看项目问题类型时，会首先调用服务层类和数据库操作类进行查询操作的处理，将查询到的信息封装到页面对象，并最终显示至查询结果界面。

##### 项目问题统计时序图如下图 6-8 所示：



* + 1. 核心处理流程设计

图 6-8 项目问题统计时序图

##### 项目问题统计子模块模块活动图如下图 6-9 所示：

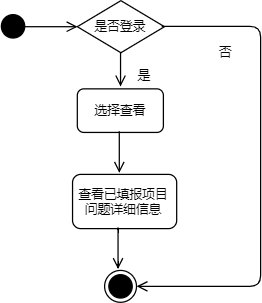
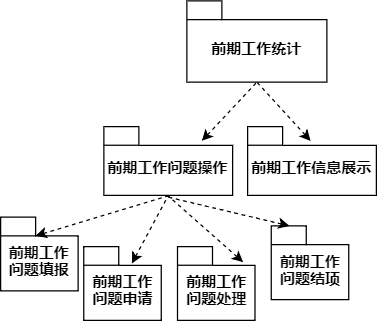


图 6-9 项目问题统计活动图

#### 前期工作统计模块

##### 功能结构设计

该模块主要面对发改部门、行业主管部门。已登录成功的发改部门、行业主管部门能够进行查看前期工作统计情况。其中行业主管部门可以对前期工作问题进行填报、申请、结项。发改部门可以对已申请的前期工作问题进行处理后反馈给行业主管部门，也可以查看前期工作详细信息的统计。前期工作统计模块包图如下图 6-10 所示：

##### 类图设计

图 6-10 前期工作统计包图

##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\5JR1_MQK[78MVH}P6{YC}_7.png在前期工作统计模块类图中，主要涉及五个类的关联，其中前期工作控制器类继承基本控制器类，依赖于前期工作服务层类和前期工作类，前期工作服务层类依赖于前期工作数据库操作类和回答类，前期工作数据库操作类依赖于前期工作实体类。前期工作统计模块其类的关系如图 6-11 所示：

图 6-11 前期工作统计模块类图

##### 前期工作统计模块涉及的各个类的描述如下表所示，前期工作实体类描述表如下表6-5

所示：

表 6-5 前期工作实体类描述

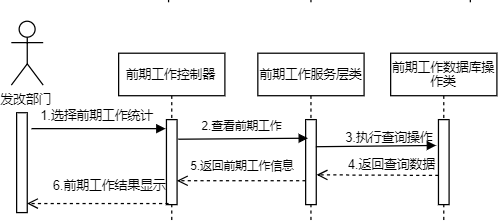
|  |  |  |
| --- | --- | --- |
| 前期工作实体类 | | |
| 类的描述：前期工作的实体对象类 | | |
| 属性描述 | proid | 前期工作编号 |
| pid | 项目编号 |
| projectname | 项目名称 |
| proname | 前期工作名称 |
| prostatus | 前期工作状态 |
| pioland | 用地批复单位 |
| pifnland | 批复文号 |
| psland | 批复状态 |
| 操作描述 | setter/getter() | set 和 get 方法 |

##### 前期工作控制器类描述表如下表 6-6 所示：

表 6-6 前期工作控制器类描述

|  |  |  |
| --- | --- | --- |
| 前期工作控制器类 | | |
| 类的描述：前期工作的控制器类，负责请求处理和转发 | | |
| 操作描述 | gz\_fixPage | 前期工作事项展示 |
| qq\_handlePage | 发改部门处理前期工作事项 |

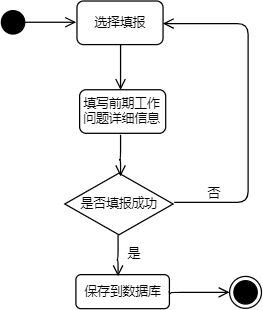
##### 顺序图设计

在前期工作统计的时序图中，发改部门在进行查询前期工作信息统计时，会首先调用服务层类和数据库操作类进行查询，之后返回数据，并最终渲染至查询界面进行项目信息显示。前期工作统计时序图如下图 6-12 所示：

##### 核心处理流程设计

图 6-12 前期工作统计时序图

前期工作问题填报子模块活动如下图 6-13 所示：

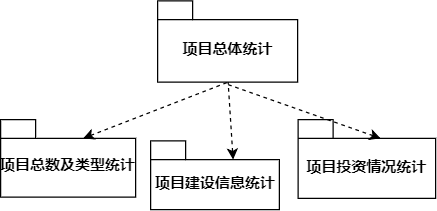


#### 项目总体统计模块

图 6-13 前期工作问题填报时序图

##### 功能结构设计

该模块主要面对发改部门、项目（法人）单位。已登录成功的发改部门、项目（法人） 单位能够进行查看项目统计情况。项目总体统计模块包图如下图 6-14 所示：



##### 类图设计

图 6-14 项目总体统计包图

##### 在项目总体统计模块类图中，主要涉及五个类的关联，其中项目控制器类继承基本控制器类，依赖于项目服务层类和项目信息类，项目服务层类依赖于项目数据库操作类和回答类，项目数据库操作类依赖于项目实体类。项目总体统计模块类的关系如图 6-15 所示：

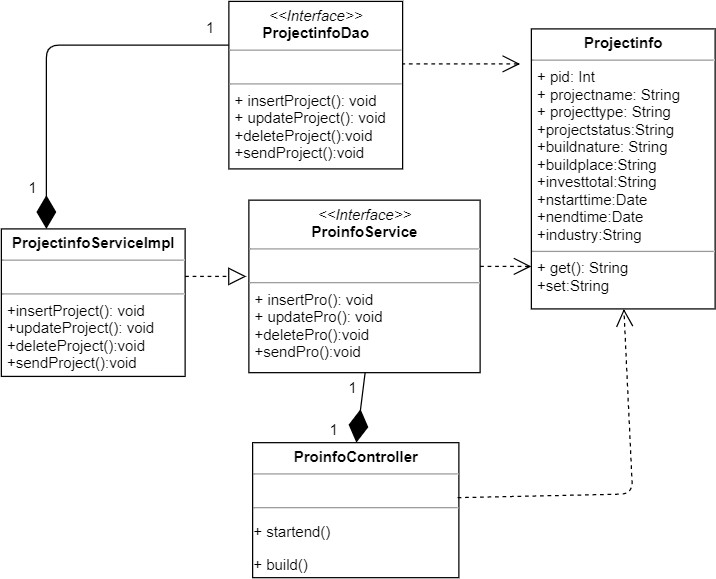


图 6-15 项目总体统计模块类图

##### 项目实体类描述表如下表 6-7 所示：

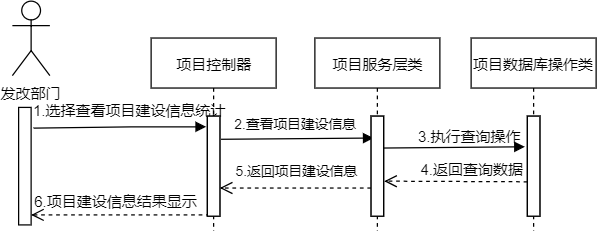
表 6-7 项目实体类描述

|  |  |  |
| --- | --- | --- |
| 项目实体类 | | |
| 类的描述：项目的实体对象类 | | |
| 属性描述 | pid | 项目名称 |
| pname | 项目类型 |
| buildnature | 建设性质 |
| buildplace | 建设地点 |
| industry | 所属行业 |
| investmenttotal | 总投资额 |
| nstarttime | 拟开工 |
| nendtime | 拟竣工 |
| 操作描述 | setter/getter() | set 和 get 方法 |

##### 顺序图设计

在项目总体统计的时序图中，管理员在进行查询项目建设信息时，会首先调用服务层类和数据库操作类进行查询，之后返回数据，并最终渲染至查询界面进行项目建设信息显

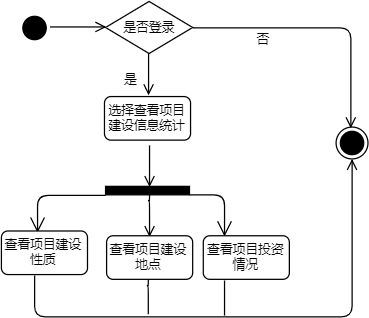
##### 示。项目总体统计时序图如下图 6-16 所示：



* + 1. 核心处理流程设计

图 6-16 项目总体统计时序图

项目建设信息统计子模块活动图如下图 6-17 所示：



#### 问题督办统计模块

图 6-17 项目建设信息统计时序图

##### 功能结构设计

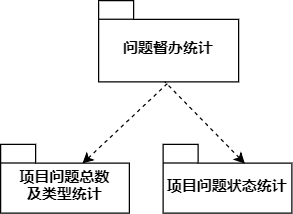
问题督办统计模块的主要功能是发改部门可以对所有项目问题进行统计，统计项目问题总数及类型、项目问题状态。问题督办统计模块包图如下图 6-18 所示：

图 6-18 问题督办统计模块包图

##### 类图设计

在问题督办统计模块类图中，主要涉及五个类的关联，其中问题控制器类继承基本控制器类，依赖于问题服务层类和问题类，问题服务层类依赖于问题数据库操作类和回答类， 问题数据库操作类依赖于问题实体类。问题督办统计模块的类图如下图 6-19 所示：

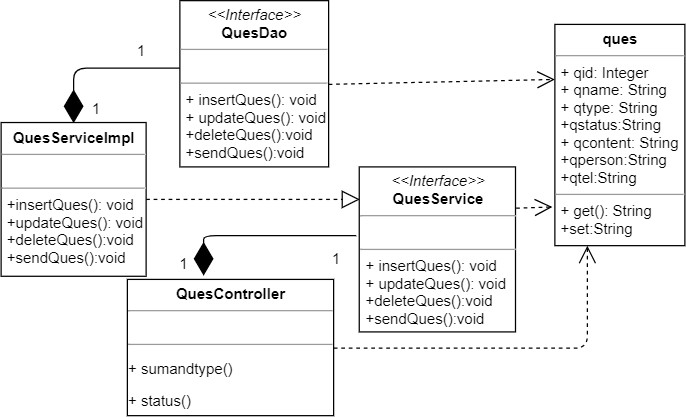


图 6-19 项目问题处理模块类图

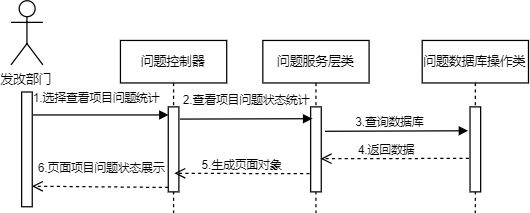
##### 问题控制层类描述表如下表 6-8 所示：

表 6-8 问题控制层类描述

|  |  |  |
| --- | --- | --- |
| 问题控制器类 | | |
| 类的描述：问题的控制器类，负责请求处理和转发 | | |
| 操作描述 | sumandtype() | 问题总数及类型统计 |
| status() | 问题状态统计 |

##### 顺序图设计

在项目督办统计的时序图中，发改部门在进行查看项目问题统计时，会首先调用服务层类和数据库操类进行查询操作的处理，将新增到的信息封装到页面对象，并最终显示至处理界面。问题督办统计模块时序图如下图 6-20 所示：



##### 核心处理流程设计

图 6-20 问题督办统计处理时序图

##### 项目问题总数及类型统计活动图如下图 6-21 所示：

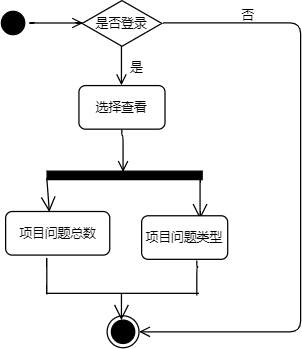


图 6-21 项目问题总数及类型统计活动图

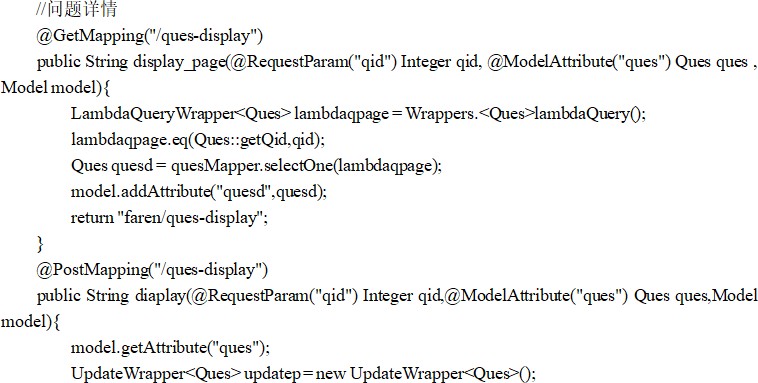
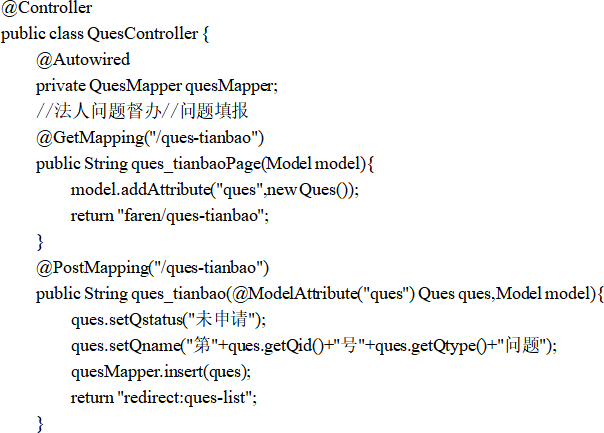
# 编码

#### 代码实现与核心算法

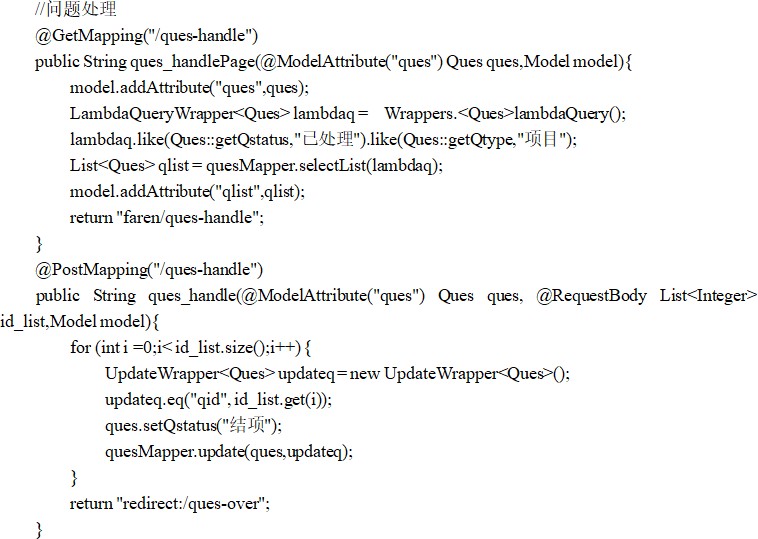
##### 项目问题管理模块核心代码

项目问题管理模块主要是对项目问题相关的请求进行处理，控制访问问题相关页面的跳转和数据查询功能，并且控制对问题的增删改查功能。

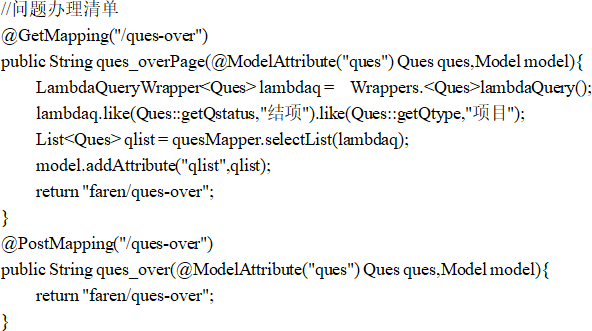
##### 在控制层，利用 controller 来完成业务逻辑，使用 Model 进行传值，通过映射来找到页面，实现项目（法人）单位进行项目问题填报和查看问题详情功能的代码如下：





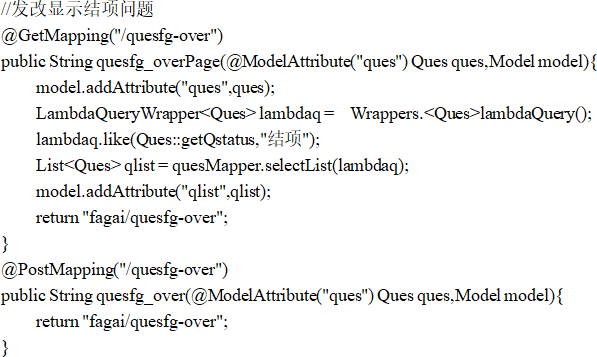
在控制层，利用 controller 来完成业务逻辑，使用 Model 进行传值，通过映射来找到页面，实现对处理项目问题的功能代码如下：

##### 项目问题统计模块核心代码

在控制层，利用 controller 来完成业务逻辑，使用 Model 进行传值，通过映射来找到页面，实现问题办理后的项目问题进行展示的代码，展示问题办理清单的代码如下。

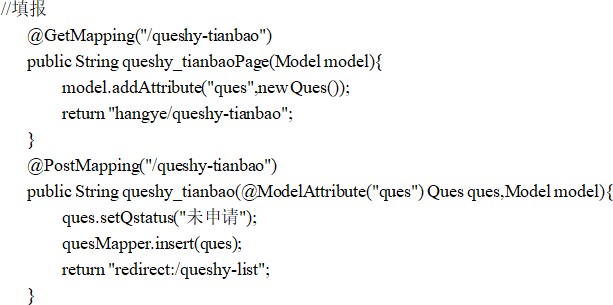
##### 在控制层，利用 controller 来完成业务逻辑，使用 Model 进行传值，通过映射来找到页面，实现已申请项目问题展示功能的代码如下：



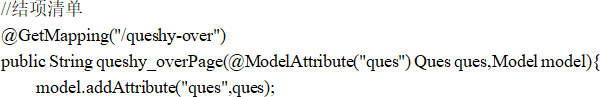
在控制层，利用 controller 来完成业务逻辑，使用 Model 进行传值，通过映射来找到页面，实现已结项问题展示的功能代码如下：

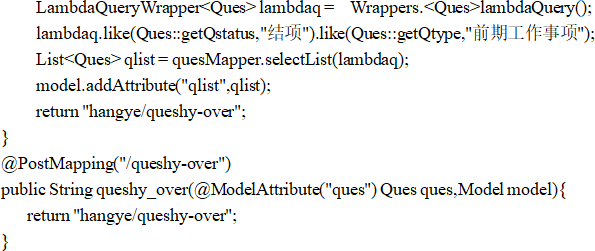
##### 前期工作统计模块核心代码

在控制层，利用 controller 来完成业务逻辑，使用 Model 进行传值，通过映射来找到页面，实现前期工作问题的填报功能代码如下：



##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\U7_NBB38U~8QO@A%0RHA9@S.png在控制层，利用 controller 来完成业务逻辑，使用 Model 进行传值，通过映射来找到页面，实现对前期工作问题进行申请的功能和对前期工作问题进行结项的功能代码如下：

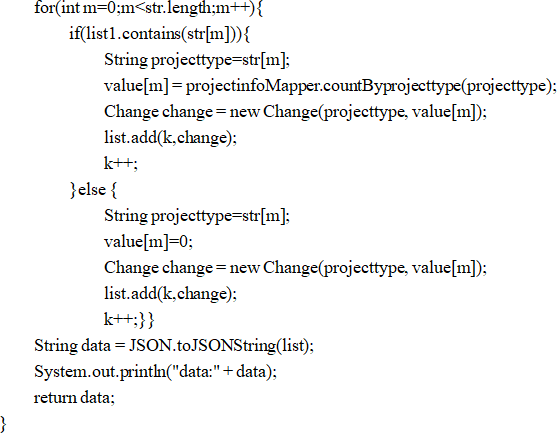


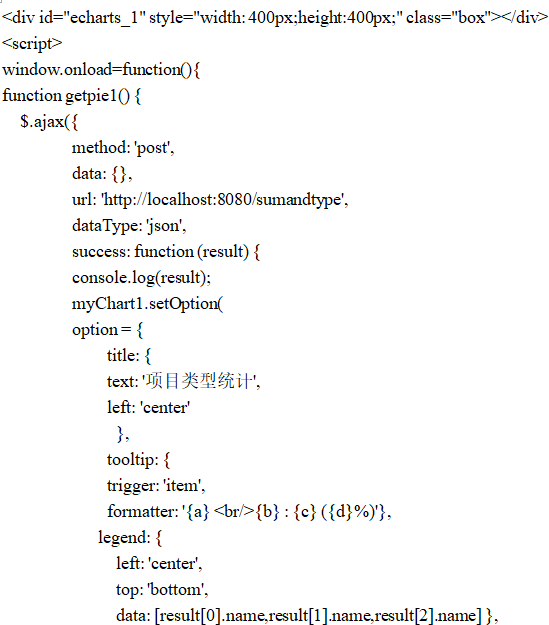


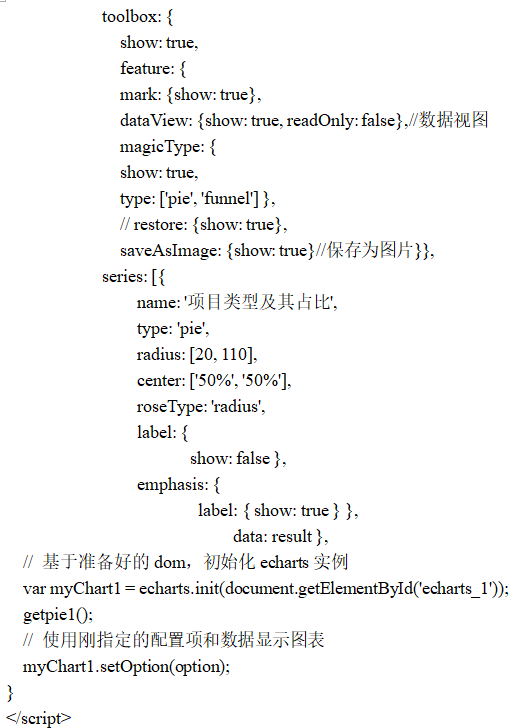
* + 1. 项目总体统计模块核心代码

##### 利用 Echarts 技术与 ajax 技术共同将项目总数及项目类型的数据从数据库中获取，并准备传到前台进行展示的代码如下：





利用 Echarts 技术与 ajax 技术共同在前台渲染出饼图展示项目类型及其占比的代码如下：



#### 代码优化分析

##### 对于代码优化，它的目的是让最终生成的代码可以更加轻松地实现功能，从而提高时空效率。但是在代码优化过程中，要注意优化是适度地优化，如果过度地优化会使得程序的可读性会降低，同时后期维护人员对系统的维护也更加困难。所以在优化代码中应该注意以下几点：

1. 明确优化目标。在一开始，应该清楚地了解自己想要达到的效果，明确如何就能够满足需求，同时清楚和其他优化有关的问题，在整个优化过程中都做好要坚持以上的准备。

##### 选择合适的优化指标。选择合适的指标来衡量自己的优化效果，这样才能避免过度优化，并且可以按照这些指标来做好优化工作计划和评估优化工作。

1. 不应该过早优化。因为如果项目早期就急着优化，会影响运行或者阅读。对于优化， 要将代码优化独立于开发来做。
2. 优化并非万能。因为不能优化一切，有时候也没有办法在同一时刻去优化两个方面。有舍就有的，在速度得到优化的时候，可能就会给资源利用带来阻碍。所以在这样的情况下自己就需要权衡一下，哪个更符合优化目标。

# 测试

#### 测试方案设计

##### 测试策略

在本系统的开发过程中，单元测试与集成测试伴随其中。其中单元测试是对系统的每个单元进行测试，它需要的分析数据比较少，针对性比较强，通过单元测试可以确定问题的范围再进行排查纠错。而集成测试与单元测试有所不同，集成测试是建立在单元测试的基础之上进行的，对组装好的子系统进行测试。而系统测试则是建立在[集成测试](https://baike.baidu.com/item/%E9%9B%86%E6%88%90%E6%B5%8B%E8%AF%95)之后，在现实的运行环境之下对计算机系统进行严格有效的测试，用来发现软件中隐含的问题。最后进行交付测试，交付测试是为了保证用户对所交付的系统满意。

##### 测试进度安排

基于对系统的分析和初步选择的测试策略，本系统的测试进度安排如下表 8-1 所示：

表 8-1 测试进度安排表

|  |  |
| --- | --- |
| 测试阶段 | 测试任务 |
| 单元测试 | 对软件中的最小可测试单元进行检查和验证 |
| 集成测试 | 根据设计的要求，把所有的模块都组合成为子系统或系统，进行集成测试。 |
| 系统测试 | 对整个系统的测试，检验它是否有不符合系统说明书的地方 |
| 交付测试 | 保证用户对所交付的系统满意 |

##### 测试资源

所需测试资源如下表 8-2 所示：

表 8-2 测试资源表

|  |  |
| --- | --- |
| 资源类型 | 资源描述 |
| 计算机硬件 | Intel I5 4G 内存 500G 硬盘 |
| 操作系统 | Windows 10 1920🞨1080 分辨率 |
| JDK | JDK 1.8 |
| 浏览器 | Chrome 浏览器 |
| 开发环境 | IDEA2018 |
| 服务器 | Tomcat9.0 |

##### 关键测试点

本系统中进行测试时的关键测试点列举如下：

##### 注册界面，能否注册成功；注册成功后选择登录是否能进入该角色系统首页；

1. 在项目问题填报时，是否所有信息填写完成后才能提交，否则提示请填写完整；

##### 在项目问题申请后，是否会将数据传送至发改部门的问题待处理界面；

1. 添加项目问题之后，问题督办统计界面关于项目问题的统计数据是否及时更新。

#### 测试用例构建

##### 测试用例编写约定

测试用列是为了实施测试而编写的，它是有规范性的输入与输出的，它的主要目的就是为了便于测试某个程序段结果或者用来检验是否实现某个特定的功能。

##### 测试用例设计

用例编写有最基本的要求。第一，具有清晰的目的、良好的预置条件、清晰的测试步骤、期望结果的；第二，能够被他人理解的；第三， 能够被他人执行的。

##### 测试用例的具体分项要求如下：第一，测试目的要能够清晰表达出测试到的功能点。第二，预置条件。是在测试用列在使用前的要求，系统应该处于什么状态，或者说进行到哪个步骤，在哪个页面的情况下，可以看到什么的基础之上。最好所有测试用例都要有前置条件，而且一个测试用例的前置条件不能够将其他的测试用例作为前置条件，而且前置条件应该用语言描述好。第三，测试步骤。对于测试用例的测试步骤，首先应该描述清楚如何操作，而且操作和结果应该是对应的，但是操作中不能有对于结果的检查，而且连词、介词等不可以出现在原理之中，同时假设性词汇和二义性语句也不应该在一列描述内。第四，预期结果。一般对于每一个用例都应该有预期结果，其结果不可以是空的。而且在结果中只包括了结果，没有任何步骤。

* + 1. 关键测试用例

##### 针对本系统的所有测试用例，从中选取了关键测试用例。其中项目问题管理模块关键测试用例如下表 8-3 所示：

表 8-3 项目问题管理模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 测试目的 | | 测试项目问题管理模块中项目（法人）单位填报项目信息问题的功能 | |
| 预置条件 | | 登录注册界面显示 | |
| 序号 | 测试步骤 | | 预期结果 |
| 1 | 在登录界面输入邮箱与密码后点击登录按钮 | | 跳转至项目（法人）单位首页界面 |
| 2 | 点击左边导航栏内的问题督办中的问题填报 | | 跳转至项目（法人）单位问题填报页面 |
| 3 | 选择项目问题类型中的项目信息，填写项目信  息问题详细信息，点击提交按钮 | | 跳转至问题申请界面，且问题申请界面出现  刚刚填报的项目信息问题的信息 |
| 测试状态 | | 上述操作均达到“测试”要求 | |
| 测试结果 | | 经过测试，该模块能够对项目（法人）单位填报项目信息问题进行正确处理，也可以对错误的操作进行提示，达到了预期目标。 | |

##### 项目问题统计模块关键测试用例如下表 8-4 所示：

表 8-4 项目问题统计模块测试用例

|  |  |
| --- | --- |
| 测试目的 | 测试项目问题统计模块中查看已结项项目问题的功能 |
| 预置条件 | 登录注册界面显示 |

续表 8-4

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试步骤 | | 预期结果 |
| 1 | 在登录界面输入邮箱与密码后点击登录按钮 | | 跳转至项目（法人）单位首页界面 |
| 2 | 点击左边导航栏内的问题督办中的问题结项 | | 跳转至项目（法人）单位问题结项页面即可  查看到已结项项目问题 |
| 测试状态 | | 上述操作均达到“测试”要求 | |
| 测试结果 | | 经过测试，该模块能够对查看已结项项目问题进行正确处理，达到了预期目标。 | |

##### 前期工作统计模块关键测试用例如下表 8-5 所示：

表 8-5 前期工作统计模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 测试目的 | | 测试前期工作统计模块中前期工作问题的填报的功能 | |
| 预置条件 | | 登录注册界面显示 | |
| 序号 | 测试步骤 | | 预期结果 |
| 1 | 在登录界面输入邮箱与密码后点击登录按钮 | | 跳转至行业主管部门首页界面 |
| 2 | 点击左边导航栏内的问题督办中的问题填报 | | 跳转至行业主管部门的问题填报页面 |
| 3 | 选择项目问题类型中的前期工作，填写项目信  息问题详细信息，点击提交按钮 | | 跳转至问题申请界面，且问题申请界面出现  刚刚填报的前期工作问题的信息 |
| 测试状态 | | 上述操作均达到“测试”要求 | |
| 测试结果 | | 经过测试，该模块能够对行业主管部门填报前期工作问题进行正确处理，也可以对错误的操作进行提示，达到了预期目标。 | |

##### 测试用例维护

针对已经存在的测试用例，考虑到系统的升级及改变带来的影响，应该注意以下几点：

##### 选择删除已过时的测试用例。有时候受客户需求改变的影响，在被测试系统中，会有原本合适的测试用例也不再适合使用,我们应该及时删除这些过时的测试用例。

1. 改进不受控制的测试用例。因为软件是不断发展的,所以在测试用例库中的测试用例会一直增多,对于这里面的一些测试用例，如果其对输入或运行状态十分敏感，那么容易不受控制，应该及时删除。

##### 删除冗余的测试用例。在测试用例库中，检查是否存在两个或者更多个测试用例，它们的输入和输出是一样的，如果存在这样的测试用例,那么这些测试用例就有的是冗余的，应该删除。

1. 增添新的测试用例。因为后续开发会增加模块或接口，或者某程序段或关键的接口在现有的测试中没有被测试,在这样的情况下就有必要去开发出新的测试用例对它们进行测试。

# 总结与展望

#### 设计工作总结

##### 本次完成互联网+固定资产投资服务问题督办与统计分析平台的设计与实现，经历了选题和资料收集阶段、分析计划阶段、设计阶段、实现和测试阶段等软件开发的过程， 设计工作总结如下：

1. 在技术学习方面，学习 Spring Boot 框架，理解 MVC 开发模式，学习使用 Echart 进行可视化分析，学习使用了 ajax 的原理和基本步骤，实现数据的异步交互。开发过程中遇到许多的问题，明白到在学习技术的过程中需要有自己的思考，学会找到解决问题的方式和方法，学习利用已有的知识和掌握的技术，来快速理解和使用新技术服务开发。

##### 在团队合作方面，作为团队中的一员，既要及时完成自己负责的工作，也要在项目开发和整合上积极配合。

1. 在总体能力提高方面，更加熟悉了开发系统的一个完整流程，并且有了更深刻的认识； 提升了自己分析问题和解决问题的能力，并且掌握了一些解决问题的方法；同时也提高了技术总结和撰写论文的能力和逻辑思维能力。

#### 未来工作展望

##### 计算机作为大家日常生活中的重要工具，其中软件开发技术对于提升工作效率和生活便捷程度都具有重大意义。在未来的工作中，自己要持续不断地学习新技术，努力提高自己的个人能力，打下坚实的技术基础，为以后能够更好地解决问题时刻准备着。同时在工作中要学会从整体看问题和要有全局观意识。同时在软件开发的过程中，自己以后工作中需要注意一下几点：

1. 要遵循编码规范。从学生身份转变成职工身份，其中很重要的就是要学会遵守规范。在学校时，没有着重强调编码的规范性，自己在编码规范上存在不足，需要一步步改正。

##### 要对代码进行充分的自测再进行提交。因为软件开发人员要对代码质量负责，我们要保证提交的代码中的问题尽量少。

1. 要编写好开发过程中的相关文档。好的文档能够有效帮助维护程序版本、升级程序版本等后续工作的进行。

# 谢 辞

# 参考文献

1. 冀潇,李杨.采用 ECharts 可视化技术实现的数据体系监控系统[J].计算机系统应用,2017,26(06):72-7 6.
2. 安璐.“互联网+”打造政务服务新模式——山西省投资项目在线审批监管平台正式运行[J].中国集体经济,2018(19):159-160.
3. 彭良英.基于“互联网+”的政府投资建设项目公众参与研究[D].华南理工大学,2017. [4]张德申.微信公众号开发订阅号功能开发研究[J].电子技术与软件工程 2013,(19):66-6

[5]林世祺,罗志勇,黄源,李秋健.基于“互联网+”技术的防雷安全监管平台研究[J].价值工程,2020,39(0 5):226-228.

[6]赵晓丹.网页开发中的 JSP 技术分析[J].南方农机,2019,50(20):247-248.

[7]林寒超.SpringMVC 技术的研究[J]．计算机技术与发展.2006，16(11)：112-113． [8]王学鸣.微信公众号的现状、类型及发展趋势研究[J].无线互联科技 ,2018,(5):60-61.

1. 刘康,徐同坤.基于提高学生兴趣和实践能力的 Java 程序设计微课教学改革研究[J]. 电脑迷,2019(0 1):224.
2. 雷静,叶煜.融入信息化教学手段的高职Java 语言程序设计课程教学模式探索与实践[J]. 当代教育实践与教学研究,2019(01):15-16.
3. 张之涵,谭霞,温克欢,魏恩伟.基于 Android 和 Java 编程的家庭智能用电管理系统设计[J]. 自动化与仪器仪表,2019(01):121-124.
4. 杨开振.深入浅出 Spring Boot 2.x [M].人民邮电出版社，2018.
5. 王瑶.系统软件开发过程中的软件工程技术应用策略[J].科技展望，2017,27（30）：1-2+15. [14]吴为胜、杨章伟.Spring+Spring MVC+MyBatis [M]. 北京：清华大学出版社,2019.
6. 黎连业;李龙.软件测试实用技术与常用模板（第 2 版）.机械工业出版社.2018.3
7. 王祥顺.软件工程的安全检测和维护[J].电子技术与软件工程，2017,（19）：36. [17]张峰.应用 SpringBoot 改变 web 应用开发模式[J].科技创新与应用,2017(23):193-194.

[18]熊永平.基于 SpringBoot 框架应用开发技术的分析与研究[J].电脑知识与技术,2019,15(36):76-77. [19]朱璐.基于 SpringBoot 和 Echarts 的仪表盘Dashboard 系统的设计与实现[D].南京大学,2019. [20]佟宇涵,朱欣彤,张文静.基于 SVG、Echarts 的数据可视化分析[J].湖北农机化,2020(02):164.

1. 韩中豪,王梓名.基于 Spring Boot 框架的电子评标系统的设计与实现[J].现代信息科技,2019,3(23): 15-16+19.
2. 刘丽华,薛玉倩.基于 spring MVC 的学生信息管理系统[J].通讯世界,2019,26(10):173-174.
3. 刘杰,孙浩,郭东旭,吴雨洽.基于 Spring MVC 及 MyBatis 框架的在线教育平台的设计与实现[J]. 沈阳师范大学学报(自然科学版),2019,37(03):268-273.
4. 孔晓.软件工程中的常用软件生命周期模型[J].电子技术与软件工程，2017,（14）：58. [25]于博文.基于计算机软件工程的数据库编程技术[J].中国高新区，2017,（24）：182

[26]M.H. Mehrabi,Zainah Ibrahim,S.S. Ghodsi,Meldi Suhatril. Seismic characteristics of X-cable brace

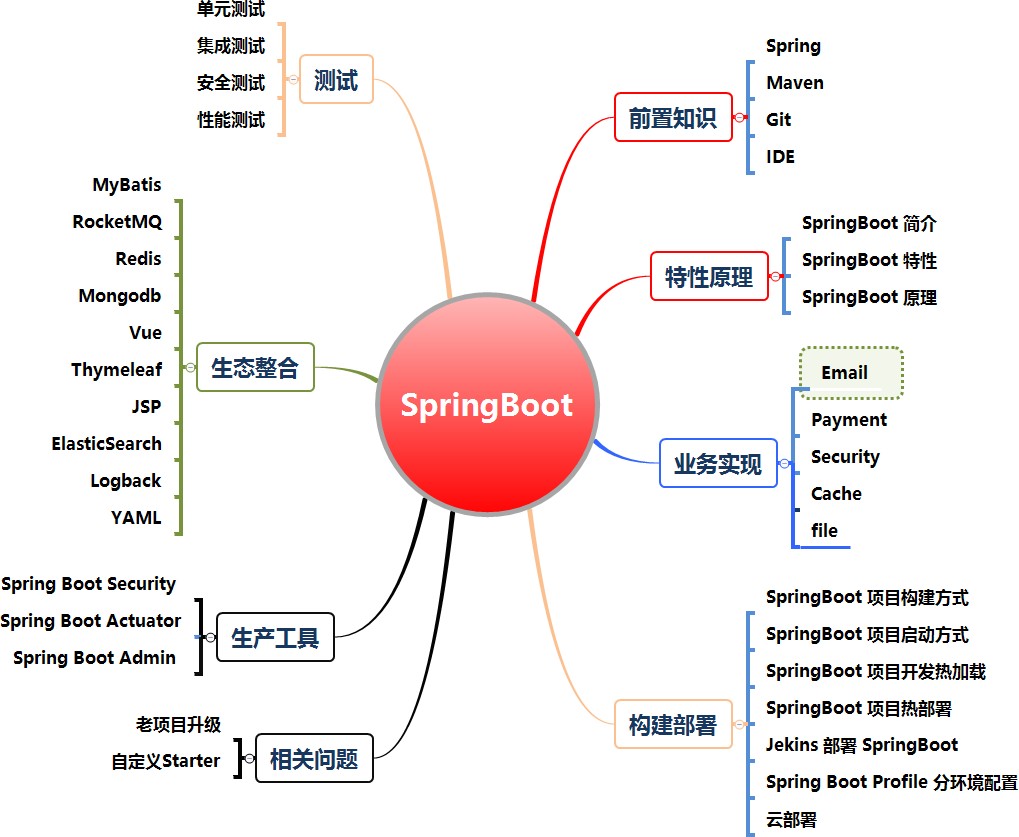
d frames bundled with a pre-compressed spring[J]. Soil Dynamics and Earthquake Engineering,2019,11 6.

1. Min Zhang, MA Weihua, Shihui Luo. Application of levitation frame with mid-set air spring on maglev vehicles. 2019, 1187(3)
2. Yuping Liu, Hongyang Wang, Canxing Qiu, et al. Seismic Behavior of Superelastic Shape Memor y Alloy Spring in Base Isolation System of Multi-Story Steel Frame.2019, 12(6)
3. Hatma Suryotrisongko, Dedy Puji Jayanto, Aris Tjahyanto. Design and Development of Backend Application for Public Complaint Systems Using Microservice Spring Boot. Procedia Computer Scienc e[J].2017, Vol.124: 736-743
4. [Catalyst 5.8, the Perl MVC Framework: Build scalable and extendable web applications using the](http://gen.lib.rus.ec/book/index.php?md5=DAFA71FDFBBB04D20C40528704047F4D) [Agile MVC framework](http://gen.lib.rus.ec/book/index.php?md5=DAFA71FDFBBB04D20C40528704047F4D) [J]．Physics in Medicine and Biology,2012，57(5)：29-33.
5. Effect of flexible patterns of health education on enhancing the compliance of pregnant women fr om Tibet, China. Medicine (Baltimore) 2020 Jan;99(1).
6. Information Technology - Cloud Computing; Reports on Cloud Computing from Shandong Univer sity Provide New Insights (Design and Implementation of International Engineering Project Manageme nt System based on Cloud Platform)[J]. Journal of Engineering,2018.
7. Haifeng Fan. Research on Network Security System Detection Based on Mysql[C]. Institute of Management Science and Industrial Engineering.Proceedings of 2019 International Conference on Information Science,Medical and Health Informatics(ISMHI 2019).Institute of Management Science and Industrial Engineering:计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society),2019:689-693.
8. Junho Lee, Jungwoong Woo, Cheongan Lee, et al. A Software Development Methodology for Secure Web Application. 2019, 9(1):336-341.
9. Xiaowei Fan,Longhe Xu,Zhongxian Li. Seismic performance evaluation of steel frames with pre-pressed spring self-centering braces[J]. Journal of Constructional Steel Research,2019,162.

### 附录 A 外文翻译—原文部分

Original source：https://[www.tutorialspoint.com/spring\_boot/spring\_boot\_introduction.htm](http://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm)

Spring Boot is an open source Java-based framework used to create a microService. It is develop ed by Pivotal Team and is used to build stand-alone andproduction ready spring applications.



* 1. Micro Service is an architecture that allows the developers to develop and deploy services independently. Each service running has its own process and this achieves the lightweight model to support business applications.
  2. Micro services offers the following advantages to its developers ： easy deployment,simple

scalability,compatible with containers,minimum configuration,lesser production time.

Spring Boot provides a good platform for Java developers to develop a stand-alone and productio n-grade spring application that you can just run. You can get started with minimum configurations wit hout the need for an entire Spring configuration setup.Spring Boot offers the following advantages to its developers :easy to understand and develop spring applications,increases productivity, reduces the d evelopment time.

Spring Boot is designed with the following goals :to avoid complex XML configuration in Spring,to develop a production ready Spring applications in an easier way,to reduce the development time and run the application independently,offer an easier way of getting started with the application.Spring Boot offers

advantages as given here :

* + 1. It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions.
    2. It provides a powerful batch processing and manages REST endpoints.
    3. In Spring Boot, everything is auto configured; no manual configurations are needed.
    4. It offers annotation-based spring application.
    5. Eases dependency management.
    6. It includes Embedded Servlet Container.

Spring Boot automatically configures your application based on the dependencies you have added to the project by using @EnableAutoConfiguration annotation. For example, if MySQL database is on your classpath, but you have not configured any database connection, then Spring Boot auto-configures an in-memory database.

The entry point of the spring boot application is the class contains @SpringBootApplication annotation and the main method.

Spring Boot automatically scans all the components included in the project by using @ComponentScan annotation.

Handling dependency management is a difficult task for big projects. Spring Boot resolves this problem by providing a set of dependencies for developers convenience.

For example, if you want to use Spring and JPA for database access, it is sufficient if you include spring-boot-starter-data-jpa dependency in your project.

Note that all Spring Boot starters follow the same naming pattern spring-boot-starter- \*, where \* indicates that it is a type of the application.

Look at the following Spring Boot starters explained below for a better understanding −

Spring Boot Starter Actuator dependency is used to monitor and manage your application. Its code is shown below −

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

Spring Boot Starter Security dependency is used for Spring Security. Its code is shown below −

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

Spring Boot Starter web dependency is used to write a Rest Endpoints. Its code is shown below −

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

Spring Boot Starter Thyme Leaf dependency is used to create a web application. Its code is shown below −

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

Spring Boot Starter Test dependency is used for writing Test cases. Its code is shown below −

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

</dependency> Auto Configuration

Spring Boot Auto Configuration automatically configures your Spring application based on the JAR dependencies you added in the project. For example, if MySQL database is on your class path, but you have not configured any database connection, then Spring Boot auto configures an in-memory database.

For this purpose, you need to add @EnableAutoConfiguration annotation or @SpringBootApplication annotation to your main class file. Then, your Spring Boot application will be automatically configured.

Observe the following code for a better understanding import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration; @EnableAutoConfiguration

public class DemoApplication { public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}

The Spring framework is the most widely used framework for Java. Its success comes from the idea, not the technology itself. Its concepts include IoC Cinversion of Control (Control Inversion) and AOP (Aspect Oriented Programming).

IoC containers, in Spring, it will think that Java resources are all Java Beans, the goal of the container is to manage these beans and their relationship. Therefore, the various Beans loaded in Spring IoC can also be understood as various resources of Java, including the creation, events, and behavior of Java Beans. They are managed by the IoC container. In addition, there will be certain between each Java Bean. Dependency, for example, the class depends on the teacher and the student. Assuming that both the teacher and the student are

Java Beans, then obviously there is a dependency relationship between the teacher and the student. The teacher and the student have an educational and educated relationship. These Spring IoC containers can manage it. It's just that Spring IoC manages objects and dependencies and relationships. It is not artificially created, but is created by Spring IoC itself through description. That is, Spring relies on description to create objects and their dependencies.

For a long time, there have been many problems in the combination of uts2 Spring, such as compatibility. Plus recent years,The truts2 vulnerability problem occurred frequently, resulting in a greatly reduced usage rate. At the same time, The View Controller framework has come to us. The Spring MVC structure is clear and the classes are relatively simple. The core IoC AOP of Spring is seamlessly connected and has become the mainstream framework in the Internet era.

The MVC pattern divides the application (input logic, business logic, and UI logic) into different aspects, while providing

Easily merge between these elements.

* Model, which encapsulates the application data and the OJO composed of them.
* View, which is responsible for rendering model data onto the view and presenting the data to users in a certain form.
* Controller (Controller), is responsible for processing user requests, and establish an appropriate model to pass it to the view rendering.

You can also define logical views in Spring MVC. You can easily find the corresponding view for rendering through the view resolver provided by it, or use its message conversion function. For example, after adding the annotation ResponseBody in the Contro ller method, SpringMVC can Through its message conversion system, the data is converted to ISON, providing Java EE networking lightweight framework integrated development-SSM framework (Spring MVC + Spring + MyBatis) and Red is implemented to the front-end Ajax request. The important point in SpringMVC is its flow and some important annotations, including important content such as controllers, view resolvers, and views.

The B / S architecture is the browser and server architecture mode. It is a change or improved architecture for the C / S architecture with the rise of Internet technology. Under this architecture, the user working interface is implemented through the WWW browser, and very little transaction logic is implemented on the front end (Browser), but the main transaction logic is implemented on the server side, forming a so-called three-tier 3-tier structure. The B / S architecture is a network architecture mode after the rise of WEB, and the WEB browser is the main application software of the client. This model unifies the client, centralizes the core functions of the system to the server, and simplifies the development, maintenance, and use of the system. Only a browser (Browser), such as Netscape Navigator or Internet Explorer, is installed on the client, and databases such as Oracle, Sybase, Informix, or SQL Server are installed on the server. The browser interacts with the database through the Web Server. This greatly simplifies the client computer load, reduces the cost and workload of system maintenance and upgrades, and reduces the user's total cost (TCO).

With the development of network technology, especially with the continuous maturity of Web technology, B / S software architecture has emerged. B / S (Browser / Server) architecture is also known as browser / server architecture. This architecture can be understood as a change and promotion of the C / S architecture. Due to the rapid development of the network, the functions of the B / S structure are becoming more and more powerful. This structure can carry out distributed information processing, which can effectively reduce resource costs and improve the designed system performance. The B / S architecture has a wider application range, greatly simplifying the client in the processing mode, users only need to install a browser, and the application logic is concentrated on the server and middleware, which can improve data processing performance. In terms of software versatility, the client of the B / S architecture has better versatility and less dependence on the application environment. At the same time, because the client uses a browser, it is more convenient in development and maintenance, which can reduce system development and The cost of maintenance. Facing the future, even the platoon-level units can access and use the system through the handheld computer (Android system) in the training ground, exercise ground and other environments.

The characteristics and basic structure of B / S: In the B / S structure, each node is distributed on the

network, and these network nodes can be divided into browser side, server side, and middleware, through the link and interaction between them Complete the system's functional tasks. The division of the three levels is logically divided, and in actual applications, different physical divisions are often made according to the actual physical network.

Browser side: The browser used by the user is the interface of the user's operating system. The user makes a request to the server side through the browser interface, and processes and displays the results returned by the server side. Show it well.

Server side: Provide data services, manipulate data, and then return the results to the middle layer, the results are displayed on the system interface.

Middleware: This is running between the browser and the server. This layer mainly completes the system logic, implements specific functions, accepts user requests and transmits these requests to the server, and then returns the results of the server to the user. The information that needs to be exchanged between the browser and the server is completed through middleware.

### 附录 B 外文翻译—译文部分

##### 原文出处：https:/[/www.tutorialspoint.com/spring\_boot/spring\_boot\_introduction.htm](http://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm)

Spring Boot 是用于创建微服务的基于 Java 的开源框架。它是由 Pivotal Team 开发的， 用于构建独立的，可立即投入生产的弹簧应用程序。

##### 微服务是允许开发人员独立开发和部署服务的体系结构。每个运行的服务都有其自己的流程，这实现了支持业务应用程序的轻量级模型。

1. 微服务为其开发人员提供以下优势：易于部署，简单的可伸缩性，与容器兼容， 最小的配置，更少的生产时间。

##### Spring Boot 为 Java 开发人员提供了一个很好的平台，用于开发可以运行的独立的生产级 Spring 应用程序。您可以从最低配置开始，而无需整个 Spring 配置设置。Spring Boot 为开发人员提供了以下优势：易于理解和开发 spring 应用程序，提高了生产率，减少了开发时间。

Spring Boot 的设计目标如下：避免在 Spring 中进行复杂的 XML 配置，以更简单的方式开发可用于生产环境的 Spring 应用程序，减少开发时间并独立运行该应用程序，并提供一种更简单的应用程序入门方法。

##### Spring Boot 具有以下优点：

1. 它提供了一种灵活的方法来配置 Java Bean，XML 配置和数据库事务。

##### 它提供了强大的批处理功能并管理 REST 端点。

1. 在 Spring Boot 中，一切都是自动配置的；无需手动配置。

##### 它提供基于注释的 Spring 应用程序。

1. 简化依赖管理。

##### 包括嵌入式 Servlet 容器。

Spring Boot 通过使用@EnableAutoConfiguration 批注，根据添加到项目中的依赖项自动配置应用程序。例如，如果 MySQL 数据库位于类路径上，但尚未配置任何数据库连接， 则 Spring Boot 会自动配置内存数据库。

##### Spring Boot 应用程序的入口点是包含@SpringBootApplication 批注和main 方法的类。

Spring Boot 使用@ComponentScan 注释自动扫描项目中包含的所有组件。

##### 对于大型项目，处理依赖关系管理是一项艰巨的任务。Spring Boot 通过提供一组依赖关系来解决此问题，以方便开发人员。

Spring 和 JPA 进行数据库访问，则在项目中包含 spring-boot-starter-data-jpa 依赖项就足够了。

##### 请注意，所有 Spring Boot 启动器都遵循相同的命名模式 spring-boot-starter- \*， 其中\*表示这是应用程序的一种。

查看下面解释的以下 Spring Boot 启动器，以更好地理解-

##### Spring Boot Starter Actuator 依赖项用于监视和管理您的应用程序。其代码如下所示-

<依赖性>

##### <groupId> org.springframework.boot </ groupId>

<artifactId> spring-boot-starter-actuator </ artifactId>

##### </ dependency>

Spring Boot Starter Security 依赖项用于 Spring Security。其代码如下所示-

##### <依赖性>

<groupId> org.springframework.boot </ groupId>

##### <artifactId> spring-boot-starter-security </ artifactId>

</ dependency>

##### Spring Boot Starter Web 依赖项用于编写 Rest Endpoints。其代码如下所示-

<依赖性>

##### <groupId> org.springframework.boot </ groupId>

<artifactId> spring-boot-starter-web </ artifactId>

##### </ dependency>

Spring Boot Starter Thyme Leaf 依赖关系用于创建 Web 应用程序。其代码如下所示-

##### <依赖性>

<groupId> org.springframework.boot </ groupId>

##### <artifactId> spring-boot-starter-thymeleaf </ artifactId>

</ dependency>

##### Spring Boot Starter Test 依赖关系用于编写测试用例。其代码如下所示-

<依赖性>

##### <groupId> org.springframework.boot </ groupId>

<artifactId> spring-boot-starter-test </ artifactId>

##### </ dependency>

自动配置

##### Spring Boot Auto Configuration 根据您在项目中添加的JAR 依赖项自动配置Spring 应用程序。例如，如果 MySQL 数据库位于类路径上，但尚未配置任何数据库连接，则 Spring Boot 会自动配置内存数据库。

为 此 ， 您 需 要 在 主 类 文 件 中 添 加 @EnableAutoConfiguration 注释或

##### @SpringBootApplication 注释。然后，您的 Spring Boot 应用程序将被自动配置。观察以下代码以更好地理解

导入 org.springframework.boot.SpringApplication;

##### 导入 org.springframework.boot.autoconfigure.EnableAutoConfiguration; @EnableAutoConfiguration

公共类 DemoApplication {

##### 公共静态 void main（String [] args）{

SpringApplication.run（DemoApplication.class，args）;

##### }

}

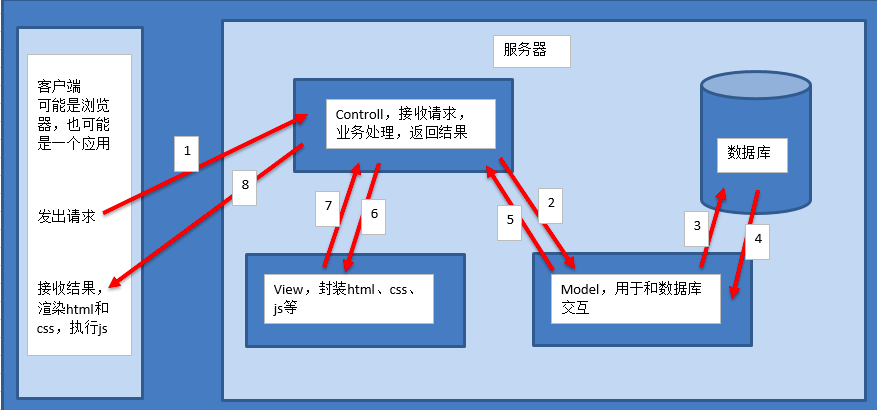
##### Spring 框架是 Java 应用最广的框架。它的成功来源于理念，而不是技术本身，它的概念包括 IoC Cinversion of Control ，控制反转）和 AOP (Aspect Oriented Programming 面向切面编程）。

(1) Spring loC 简介

##### IoC 个容器，在 Spring 中，它会认为 Java 资源都是 Java Bean ，容器的目标就是管理这些 Bean 和它们之间的关系。所以在 Spring IoC 里面装载的各种 Bean ，

也可以理解为 Java 的各种资源，包括 Java Bean 的创建、事件、行为等，它们由 IoC 容器管理此之外，各个 Java Bean 之间会存在 定的依赖关系，比如班级是依赖于老师和学生组成的，假设老师、学生都是 Java Bean，那么显然 者之间形成了依赖关系，老师和学生有教育和被教育的关系。这些 Spring IoC 容器都能够对其进行管理。只是 Spring IoC 管理对象和依赖、关系，采用的不是人为的主动创建，而是由 Spring IoC 自己通过描述创建的，也就 Spring 是依靠描述来完成对象的创建及其依赖关系的。

##### 生于 Spring Web 项目的 MVC (Model View Controller）框架走到了我们的面前， Spring MVC 结构层次清晰，类比较简单，并且 Spring 的核心 IoC AOP 无缝对接，成为了互联网时代的主流框架。

MVC 模式把应用程序（输入逻辑、业务逻辑和 UI 逻辑）分成不同的方面，同时提供这些元素之间的轻松合并。

##### Model（模型），封装了应用程序的数据和由它们组成的 OJO。

* View（视图） ，负责把模型数据渲染到视图上，将数据以 定的形式展现给用户。

##### Controller（控制器），负责处理用户请求，并建立适当的模型把它传递给视图渲染。

Spring MVC 中还可以定义逻辑视图，通过其提供的视图解析器就能够很方便地找到

##### 对应的视图进行渲染，或者使用其消息转换的功能，比如在 Contro ller 方法内加入注解 ResponseBody 后， SpringMVC 就可以通过其消息转换系统，将数据转换为 ISON ， 提供 Java EE 联网轻量级框架整合开发－SSM 框架（ Spring MVC+Spring+MyBatis ） 和 Red is 实现给前端 Ajax 请求使用。SpringMVC 中的重点在于它的流程和一些重要的注解，包括控制器、视图解析器、视图等重要内容。

B/S 架构即浏览器和服务器架构模式，是随着 Internet 技术的兴起，对 C/S 架构的一种变化或者改进的架构。

##### 

在这种架构下，用户工作界面是通过 WWW 浏览器来实现，极少部分事务逻辑在前端

##### （Browser）实现，但是主要事务逻辑在服务器端(Server)实现，形成所谓三层 3-tier 结构。B/S 架构是 WEB 兴起后的一种网络架构模式，WEB 浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser），如 Netscape Navigator 或 Internet Explorer，服务器安装 Oracle、Sybase、Informix 或 SQL Server 等数据库。浏览器通过Web Server 同数据库进行数据交互。 这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本(TCO)。

随着网络技术的发展，特别随着 Web 技术的不断成熟，B/S 这种软件体系结构出现了。B/S（Browser/Server）架构也被称为浏览器/服务器体系结构，这种体系结构可以理解为是对 C/S 体系结构的改变和促进。由于网络的快速发展，B/S 结构的功能越来越强大。这种结构可以进行信息分布式处理，可以有效降低资源成本，提高设计的系统性能。B/S 架构是有更广的应用范围，在处理模式上大大简化了客户端，用户只需安装浏览器即可，而将应用逻辑集中在服务器和中间件上，可以提高数据处理性能。在软件的通用性上，B/S 架构的客户端具有更好的通用性，对应用环境的依赖性较小，同时因为客户端使用浏览器，在开发维护上更加便利，可以减少系统开发和维护的成本。面向未来， 连排级单位可通过掌上电脑（安卓系统），在训练场、演习场等环境下访问并使用该系

##### 统。

B/S 的特征和基本结构：在 B/S 结构中，每个节点都分布在网络上，这些网络节点可以分为浏览器端、服务器端和中间件，通过它们之间的链接和交互来完成系统的功能任务。三个层次的划分是从逻辑上分的，在实际应用中多根据实际物理网络进行不同的物理划分。

##### 浏览器端：即用户使用的浏览器，是用户操作系统的接口，用户通过浏览器界面向服务器端提出请求，并对服务器端返回的结果进行处理并展示，通过界面可以将系统的逻辑功能更好的表现出来。

服务器端：提供数据服务，操作数据，然后把结果返回中间层，结果显示在系统界面上。

##### 中间件：这是运行在浏览器和服务器之间的。这层主要完成系统逻辑，实现具体的功能，接受用户的请求并把这些请求传送给服务器，然后将服务器的结果返回给用户， 浏览器端和服务器端需要交互的信息是通过中间件完成的。

附录 C 软件使用说明书

##### 系统默认进入角色登录页。如无账号，可按不同角色注册账号，注册界面如图 C-1 所示：



图 C-1 注册界面图

##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\BB2N9{XD`2N}CJ4L%HF_R)Y.png注册成功之后可跳转至登录界面，登录界面如图 C-2 所示：

图 C-2 登录界面图

##### 发改部门登录系统之后跳转至首页，左侧导航栏分为：项目审核、项目入库、项目调度、前期工作事项、问题督办、统计分析。发改部门可根据不同需求选择点击进行管理。发改部门首页如图 C-3 所示：



图 C-3 发改部门主界面图

##### D:\360Downloads\Tencent\2902319071\FileRecv\MobileFile\Image\3)`9N5S%BW@NE4V%2`~FQ`0.png项目（法人）单位登录系统之后跳转至首页，左侧导航栏分为：项目管理、项目调度、问题督办、统计分析。项目（法人）单位可根据不同需求选择点击进行管理。项目（法人）单位首页如图 C-4 所示：

图 C-4 项目（法人）单位主界面图

##### 行业主管部门登录系统之后跳转至首页，左侧导航栏分为：前期工作事项、问题督办、统计分析。行业主管部门可根据不同需求选择点击进行管理。行业主管部门首页如下图 C-5 所示：

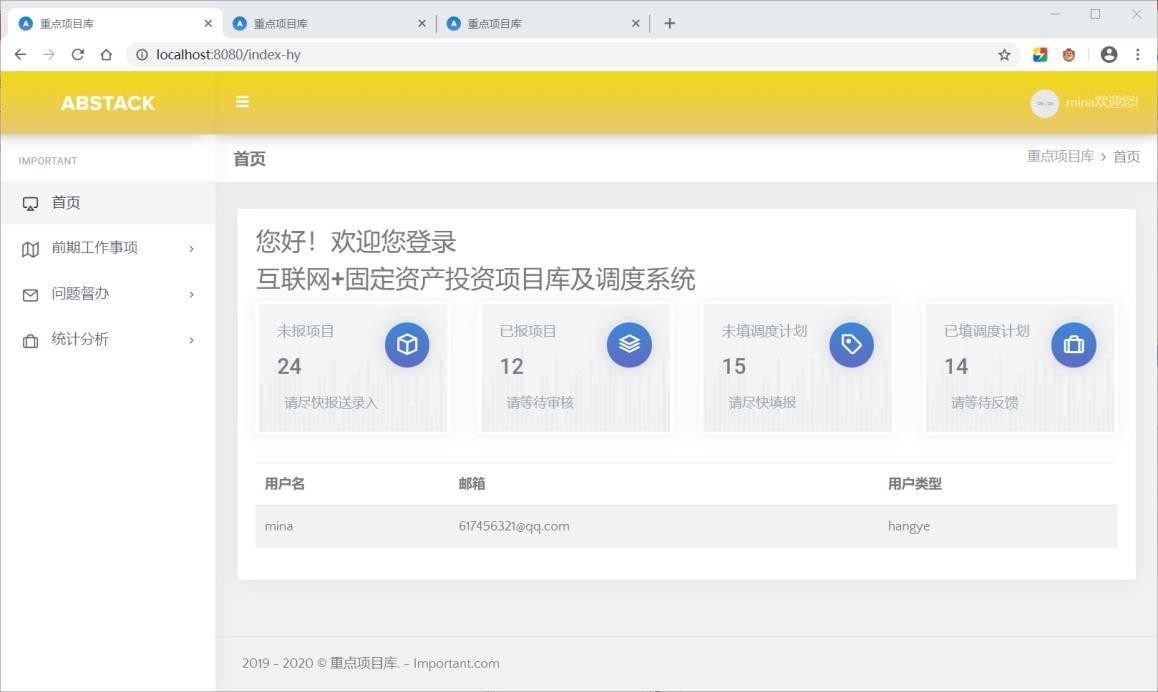


图 C-5 行业主管部门主界面图

## 附录 D 主要源代码

前期工作事项源码：

@Repository

public interface ProinfoMapper extends BaseMapper<Proinfo> {

@Select("select proinfo.\*, projectinfo.`projectname`, projectinfo.`directdept1` from proinfo , projectinfo

" +

" where proinfo.pid = projectinfo.pid and proinfo.prostatus like '%填'") List<Prostage> selectByPid(@Param(Constants.WRAPPER) Wrapper<Prostage> wrapper);

@Select("select proinfo.\*, projectinfo.`projectname`, projectinfo.`directdept1` from proinfo , projectinfo

" +

" where proinfo.pid = projectinfo.pid and proinfo.prostatus = '已填'")

List<Prostage> selectByStatus(@Param(Constants.WRAPPER) Wrapper<Prostage> wrapper); @Select("select proinfo.\*, projectinfo.`projectname`, projectinfo.`directdept1` from proinfo , projectinfo

" +

" where proinfo.pid = projectinfo.pid and proinfo.prostatus = '结项'")

List<Prostage> selectByStatuso(@Param(Constants.WRAPPER) Wrapper<Prostage> wrapper);

//hyt

@Select("select count(\*) from proinfo where prostatus='未填'") int countweitian();//统计所有未填状态的前期工作@Select("select count(\*) from proinfo where prostatus='已填'") int countyitian();//统计所有未填状态的前期工作@Select("select count(\*) from proinfo where prostatus='结项'") int countjiexian();//统计所有未填状态的前期工作@Select("select \* from proinfo")

List<Prostage> selectall();//

@Select("select count(\*) from proinfo where prostatus=#{prostatus}") int countByprostatus(String prostatus);//前期工作状态分类及占比@Select("select count(\*) from proinfo where psland=#{psland}")

int countBypsland(String psland);//前期工作状态分类及占比@Select("select count(\*) from proinfo where pifnland=#{pifnland}") int countBypifnland(String pifnland);//前期工作投资估计

//hyt

}

用户控制器相关代码：

@Controller

public class UserController {

@Autowired

private UserMapper userMapper;

@GetMapping("/login")

public String loginPage(Model model) { model.addAttribute("user", new User());

//返回登录页面return "login";

}

@PostMapping("/login")

public String login(@ModelAttribute("user") User user, Model model,ServletRequest servletRequest, ServletResponse servletResponse) {

HttpServletRequest request = (HttpServletRequest)servletRequest; HttpSession session = request.getSession();

LambdaQueryWrapper<User> wrapper = Wrappers.<User>lambdaQuery().eq(User::getEmail, user.getEmail());

User loginUser = userMapper.selectOne(wrapper);

//用户存在且密码相同if (loginUser != null &&

user.getPassword().equals(loginUser.getPassword())&&"faren".equals(loginUser.getRole())) { session.setAttribute("login", loginUser.getEmail());

session.setAttribute("role", loginUser.getRole()); session.setAttribute("loginuser", loginUser); return "redirect:index-fr";

}else if (loginUser != null && user.getPassword().equals(loginUser.getPassword())&&"fagai".equals(loginUser.getRole())){

session.setAttribute("login", loginUser.getEmail()); session.setAttribute("role", loginUser.getRole()); session.setAttribute("loginuser", loginUser);

return "redirect:index-fg";

}else if (loginUser != null && user.getPassword().equals(loginUser.getPassword())&&"hangye".equals(loginUser.getRole())){

session.setAttribute("login", loginUser.getEmail()); session.setAttribute("role", loginUser.getRole()); session.setAttribute("loginuser", loginUser);

return "redirect:index-hy";

}

else {

model.addAttribute("errorMessage", "用户名或密码错误");//否则不存在，报错return "login";

}

}

@GetMapping("/register")

public String registerPage(Model model) {

//给页面一个空的 user 属性

model.addAttribute("user", new User());

//返回注册页面return "register";

}

@PostMapping("/register")

public String register(@ModelAttribute("user") User user, Model model) { userMapper.insert(user);

//重定向到登录页面return "redirect:login";

}

//退出登录@GetMapping("/exit")

public String exit(HttpSession session){ session.invalidate();

return "redirect:/login";

}

}

项目总体统计数据库相关代码：

@Repository @Mapper

public interface ProjectinfoMapper extends BaseMapper<Projectinfo> { @Select("select count(\*) from projectinfo")

int countprojects();//声明方法@Select("select \* from projectinfo") List<Projectinfo> selectall();// @Select("select count(\*) from projectinfo") int countsum();

@Select("select count(\*) from projectinfo where projecttype='审批'") int countshenpi();//统计所有 sehnpileix 的 xmushu

@Select("select count(\*) from projectinfo where projecttype='核准'") int countshezhun();

@Select("select count(\*) from projectinfo where projecttype='备案'") int countbeian();

@Select("select count(\*) from projectinfo where projecttype=#{projecttype}") int countByprojecttype(String prostatus);//项目分类及占比@Select("select count(\*) from projectinfo where buildnature='新建'")

int countxinjian();//统计所有 sehnpileix 的 xmushu

@Select("select count(\*) from projectinfo where buildnature='扩建'") int countkuojian();

@Select("select count(\*) from projectinfo where buildnature='改建'") int countgaijian();

@Select("select count(\*) from projectinfo where buildnature='恢复'") int counthuifu();

@Select("select count(\*) from projectinfo where buildnature=#{buildnature}") int countBybuildnature(String buildnature);

@Select("select count(\*) from projectinfo where buildplace=#{buildplace}") int countBybuildplace(String buildplace);

@Select("select count(\*) from projectinfo where nstarttime like CONCAT(#{nstarttime},'%')") int countBynstarttime(Integer nstarttime);

@Select("select count(\*) from projectinfo where nendtime like CONCAT(#{nendtime},'%')") int countBynendtime(Integer nendtime);

@Select("select count(\*) from projectinfo where projectaffiliation=#{projectaffiliation}") int countByprojectaffiliation(String projectaffiliation);

@Select("select count(\*) from projectinfo where projectstatus=#{projectstatus}") int countByprojectstatus(String projectstatus);

@Select("select count(\*) from projectinfo where industry=#{industry}") int countByindustry(String industry);

}

问题督办数据库相关代码：

@Repository @Mapper

public interface QuesMapper extends BaseMapper<Ques> { @Select("select \* from ques")

List<Ques> selectall();// @Select("select count(\*) from ques") int countsum();//统计所有问题数

@Select("select count(\*) from ques where qtype='项目信息'")

int countinfo();//统计所有 sehnpileix 的 xmushu @Select("select count(\*) from ques where qtype='项目调度'") int countdis();

@Select("select count(\*) from ques where qtype='前期工作事项'") int countpro();

@Select("select count(\*) from ques where qtype=#{qtype}") int countBytype(String qtype);

@Select("select count(\*) from ques where qstatus=#{qstatus}") int countByqstatus(String qstatus);

}