# 1 系统分析

1.1系统概念分析

防雷技术服务系统，就是能够顺利完成对防雷一系列过程的安排管理，让防雷工作各个环节能够快速有序的进行，用户能够自己填写相关信息来申报需要检测的住房，管理员收到相关申报信息之后可以对申报信息进行处理，将检测任务安排给相应的检测员，检测员完成检测之后再系统中上传相应检测信息，并提交或更新结果。用户、检测员和管理员都可以按需将检测结果进行打印操作，作为纸质存档，同时信息也存储于数据库中。

1.2功能需求分析

功能性需求说明了产品在特定条件下所展示出来的行为，主要表明了开发者需要实现的功能，使得用户能以此完成它们的任务，从而满足用户的业务需求。需求分析不是一个一蹴而就的过程，它是一个持续的工作，在需求分析初期，我们只有对项目整体的、宏观的认识，随着后期工作的不断深入，需求才会逐渐细化和明晰。基于此，我们在进行系统功能需求分析的时候，应该首先从系统的使用者的角度来分析。要分析某一系统的外部角色和功能的关系，最有效的分析方式是绘制用例图。

### 1.2.1系统的总体功能

该防雷系统主要功能是面向三种用户群体：普通用户（申请者）、检测人员（领取者）和管理员（分配者）。

普通用户的权限最小，登录之后只能进行项目的申请，查看自己的申请记录，在当前申请的项目未被分配前也可以编辑更改相应的内容，当已经完成检测人员分配之后则无法进行项目细节的更改。用户也可以申请下载当前申请的项目的报表进行打印。

管理员可以查看当前新收到的尚未处理的申请，并为这些申请安排相应的检测人员，管理员可以查看所有的申请记录，也可以下载每个项目的报表。

检测人员在接到分配的项目后，登录系统会显示当前的分配项目；在检测工作完成之后，检测人员将建筑相关信息在系统中更新上传并进行提交，该项目进入更新阶段，在更新阶段，检测人员需要将该项目下的所有建筑信息上传、更新，当全部的建筑信息上传、更新完成之后，则该项目检测工作就宣告完成。进入该检测人员的已完成项目列表，检测人员也可以申请将完成的报表下载打印进行存档。

至此，一个项目的申请到分配，再到检测完成，最终实际数据上传完成之后，该项目就走完了它的流程，在整个过程中，系统为不同的角色分配了不同的功能，需要各个角色相互配合完成。

### 1.2.2不同角色登录系统的功能

### 1.2.2.1角色介绍

在整个的防雷系统的流程中，一共有三个角色参与其中，普通用户、检测人员以及管理员，这三个角色在系统中虽然使用的功能不尽相同，但也有相互重叠的功能，三个角色的具体说明如下表：

表 1-1 系统角色表

|  |  |  |
| --- | --- | --- |
| 角色 | 角色功能权限 | |
| 用户 | 申请项目、编辑申请项目 | 下载报表 |
| 管理员 | 分配检测人员 |
| 检测人员 | 上传、更新检测信息 |

### 1.2.2.2角色的具体功能及用例图

由参与者、用例以及他们之间的关系构成的用于描述系统功能的动态视图成为用例图。其中用例和参与者之间的对应关系又叫做通讯关联，他表示参与者使用了系统中的哪些用例。用例图是从软件需求分析到最终实现的第一步，他显示了系统的用户和用户希望提供的功能，有利于用户和软件开发人员之间的沟通。

用例图是需求分析中的产物，主要作用是描述参与者和用例之间的关系，帮助开发人员可视化的了解系统的功能。借助于用例图，系统用户、系统分析人员、系统设计人员、领域专家能够以可视化的方式对问题进行探讨，减少了大量交流上的障碍，便于对问题达成共识。

根绝各个用户角色的系统功能和他们的关系进行建模，整个防雷系统的用例图如图所示。

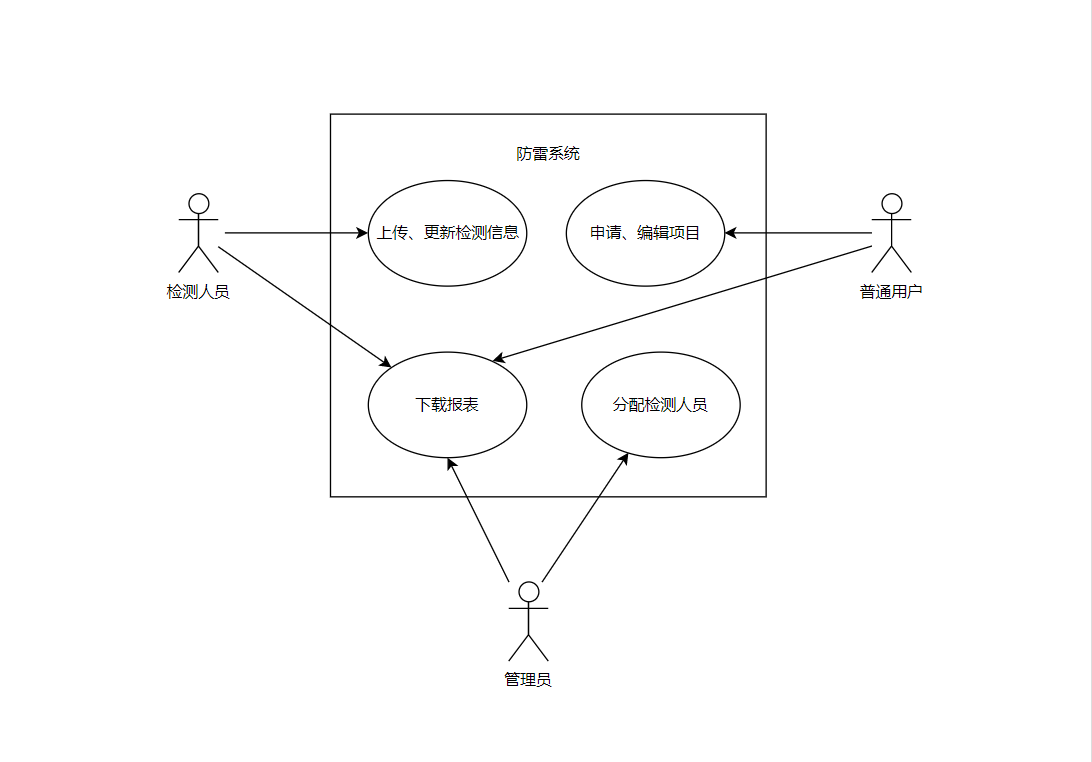


图 1-2 系统用例图

对以上系统的四个用例进行用例描述：

用例编号：1

用例名：申请、编辑项目

行为角色：普通用户

简要说明：普通用户可以选择添加项目填写相关信息进行申请项目；同时当当前申请未被管理员处理时可进行重新编辑当前项目

前置条件：普通用户成功登录进入系统

后置条件：申请新项目成功或者更新项目成功

用例编号：2

用例名：分配检测人员

行为角色：管理员

简要说明：管理员为普通用户申请的新项目分配检测员来为新项目服务

前置条件：管理员成功登录进入系统，并且有用户成功进行新项目的申请

后置条件：检测人员成功被分配，当检测人员登录系统时会领取到该项目的信息

用例编号：3

用例名：上传、更新检测信息

行为角色：检测人员

简要说明：检测人员将现场的勘测数据，图片等相关信息分别写入或更新该项目并录入系统

前置条件：检测人员成功登录进入系统，有新项目被管理员分配给该检测人员

后置条件：上传、更新检测信息成功，进入更新项目或已完成项目当中

用例编号：4

用例名：下载报表

行为角色：普通用户、管理员、检测人员

简要说明：三类用户对于下载报表都有权限（普通用户和检测人员需要打印申请被管理员通过），用户只能下载自己申请项目的报表，管理员可以下载所有项目的报表，检测人员可以下载自己参与检测上传、更新数据的所有报表

前置条件：普通用户、管理员、检测人员其中的一类用户成功登入系统，进入项目详情页面点击导出报表进行报表导出操作

后置条件：报表成功导出

### 通过以上整体系统的分析后，将以上四个大用例分解细化。申请、编辑项目的用例有：申请项目，编辑项目，搜索项目等。申请项目包括填写项目信息以及提交申请，查看项目包含了编辑项目，编辑项目又有取消编辑和提交更新。该用例图如图1-3所示。

### 

图 1-3 普通用户申请、编辑项目用例图

### 除了申请、编辑项目的用例外，普通用户、检测人员和管理员还通过系统执行下载报表这个用例。普通用户、检测人员和管理员略有区别，在普通用户和检测人员执行的用例中有：搜索项目，查看项目等，用例图如图1-4所示；在管理员执行的用例中少了批准打印和提交打印申请两个用例，用例图如图1-5所示。其中查看项目中包含了选择建筑，选择项目又包括提交打印申请和打印，打印需要依赖于管理员的批准打印。

### 

图 1-4 普通用户（检测人员）下载报表用例图

### 

图 1-5 管理员下载报表用例图

### 检测人员除了下载报表之外最主要的还是为项目中的建筑进行一一检测然后分别上传检测结果和现场照片，在上传更新检测信息的用例中主要包含的是查看正在更新项目，下面依次有选择项目和新增检测信息，新增检测信息包括了上传建筑照片和填写建筑检测信息两个关键用例，如图1-6所示。

### 图 1-6 检测人员上传更新检测信息用例图

### 管理员还有一个重要的用例就是分配检测人员，在该用例中包括了查看用户新申请的项目，选择检测人员以及分配这几个用例，如图1-7所示。

### 

### 图 1-7 管理员分配检测人员用例图

## 1.3业务流程分析

在使用用例图对系统的功能角色进行分析后，防雷系统的整体逐渐明朗，但是上一节的工作只是抽象描述了防雷系统提供给不同用户的各种功能，而没有体现它们之间流程和调用关系，下面本节需要对以上的一些主要功能进行更细化的需求分析，即业务流程分析。在这个部分主要使用数据流图和泳道图来对业务流程进行描述。

### 1.3.1数据流图分析

在需求分析阶段， 数据流(也称信息流)是分析的基础。所谓数据流，形象地说就是系统中“流动的数据结构”。数据流图(DFD-Data Flow Diagram)是描述软件系统中数据处理过程的一种有力的图形工具。数据流图从数据传递和加工的角度出发，刻画数据流从输入到输出的移动和变换过程。由于它能够清晰地反映系统必须完成的逻辑功能(业务逻辑)，所以它已经成为需求分析阶段最常用的工具。

根据分析确定出数据来源、终点、数据流、数据存储、数据处理等要素后开始构建数据流图如下，同时对每一个数据流图都采用了相对应的数据字典（Data Dictionary）配套，用于描述数据流图中的数据流、文件的数据构成、 加工处理及外部实体。

顶层数据流图如图1-8所示，只包含一个加工，用以表示待开发的系统，在该系统中输入数据即是用户登录时输入的账号密码，输出数据流为报表。它的作用在于表明防雷系统的范围以及它和周围环境的数据交换关系。

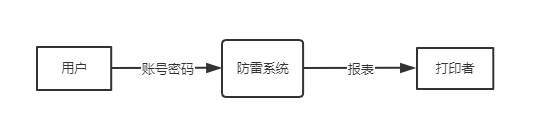


图 1-8 顶层数据流图

名称：用户

简述：为能够登录该系统的人员，包括：普通用户、检测人员和管理员

有关数据流：账号密码

名称：打印者

简述：在系统中执行打印操作并成功打印的用户

有关数据流：报表

数据流名称：账号密码

描述：通过账号密码来登录防雷系统

定义：账号密码=登录账号+密码+验证码

数据流来源：来自用户登录系统

数据流去向：用于登录系统以及用户判断

数据流名称：报表

描述：防雷系统的报表导出

定义：报表=项目信息+建筑信息+打印时间

数据流来源：进行报表打印

数据流去向：由打印申请者所拥有

接下来为下层数据流图，即系统的内部情况，一般从0层数据流图开始，采用自顶向下、由内到外的原则对加工进行分解，不能分解的加工就是基本加工。0层数据流图如图1-9所示。

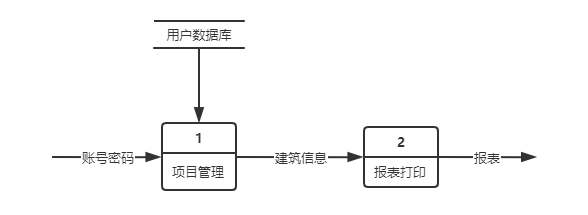


图 1-9 0层数据流图

数据流名称：建筑信息

描述：申请打印成功批准后从数据库中查询出来的建筑信息

定义：建筑信息=建筑名称+建筑状态+危险程度+建筑地址+长度+宽度+高度+经度+纬度+图片信息

数据流来源：用户的项目管理

数据流去向：报表打印

对0层数据流图中的1号加工操作进行分解得到1层数据流图如下图1-10所示。

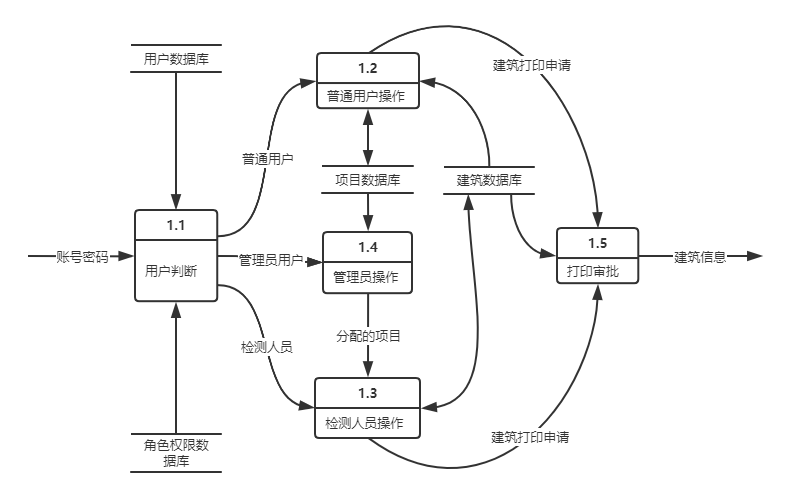


图 1-10 1层数据流图

数据流名称：普通用户

描述：防雷系统中的普通用户

定义：普通用户=账号信息+菜单信息

数据流来源：进行登录用户的判断之后

数据流去向：进入普通用户操作

数据流名称：检测人员

描述：防雷系统中的检测人员

定义：检测人员=账号信息+菜单信息

数据流来源：进行登录用户的判断之后

数据流去向：进入检测人员操作

数据流名称：管理员用户

描述：防雷系统中的管理员用户

定义：管理员用户=账号信息+菜单信息

数据流来源：进行登录用户的判断之后

数据流去向：进入管理员用户操作

数据流名称：分配的项目

描述：防雷系统中管理员为检测人员分配新申请的项目

定义：分配的项目=项目信息+检测人员名字

数据流来源：来源于管理员的操作

数据流去向：流入检测人员操作中

数据流名称：建筑打印申请

描述：防雷系统中普通用户或检测人员在建筑检测数据上传提交之后可查看具体信息并提交打印的申请

定义：建筑打印申请=申请人+建筑信息

数据流来源：来源于普通用户的操作或检测人员的操作

数据流去向：流入打印审批中

第2层数据流图是继续对第1层数据流图中的加工部分的继续分解。第1层数据流图中的1.2、1.3、1.4、1.5分别对应下图中的图1-11、图1-12、图1-13、图1-14。

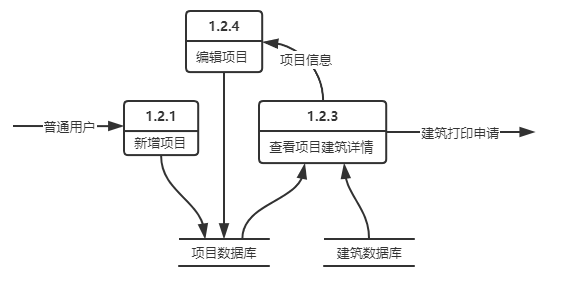


图 1-11 2层数据流图（1）

数据流名称：项目信息

描述：记录新增的项目未被管理员分配时的项目信息

定义：项目信息=项目名称+项目地址

数据流来源：查看项目建筑详情

数据流去向：编辑项目

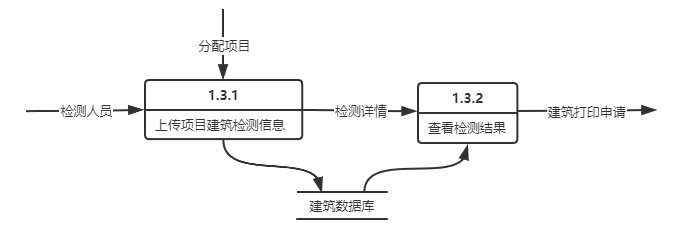


图 1-12 2层数据流图（2）

数据流名称：检测详情

描述：记录检测人员查看自己更新的项目的某一个建筑检测数据

定义：检测详情=建筑信息

数据流来源：上传项目建筑检测信息

数据流去向：查看检测结果

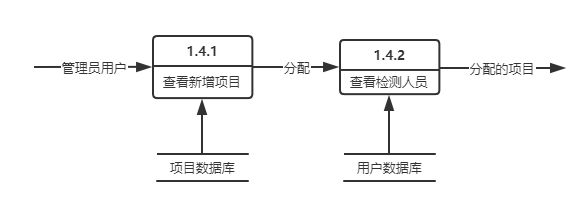


图 1-13 2层数据流图（3）

数据流名称：分配

描述：记录该新增项目中的工作人员

定义：分配=项目信息

数据流来源：查看新增项目

数据流去向：查看检测人员

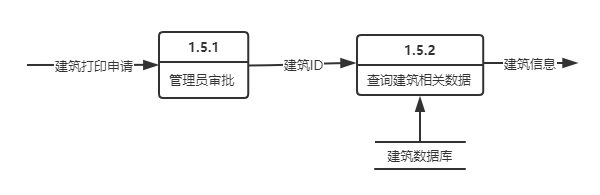


图 1-14 2层数据流图（4）

数据流名称：建筑ID

描述：记录该次申请打印的建筑ID

定义：建筑ID=建筑ID

数据流来源：管理员审批

数据流去向：查询建筑相关数据

1)、建筑信息：

建筑ID：1{[0|1|2|3|4|5|6|7|8|9]}11

建筑名称：4{字符}24

建筑状态：4{字符}255

危险程度：2{字符}24

建筑地址：4{字符}255

长度：0{[0|1|2|3|4|5|6|7|8|9]}255+.+0{[0|1|2|3|4|5|6|7|8|9]}255

宽度：0{[0|1|2|3|4|5|6|7|8|9]}255+.+0{[0|1|2|3|4|5|6|7|8|9]}255

高度：0{[0|1|2|3|4|5|6|7|8|9]}255+.+0{[0|1|2|3|4|5|6|7|8|9]}255

经度：0{[0|1|2|3|4|5|6|7|8|9]}3+.+0{[0|1|2|3|4|5|6|7|8|9]}255+[E|W]

纬度：0{[0|1|2|3|4|5|6|7|8|9]}2+.+0{[0|1|2|3|4|5|6|7|8|9]}255+[S|N]

项目ID：1{[0|1|2|3|4|5|6|7|8|9]}11

2)、项目信息：

项目ID：1{[0|1|2|3|4|5|6|7|8|9]}11

项目名称：4{字符}24

检测人员：2{字符}24

项目地址：4{字符}255

3)、账号密码：

账号：4{字母}10

密码：6{字母}24+6{[0|1|2|3|4|5|6|7|8|9]}24

验证码：4{字母}4

4)、图片信息：

图片ID：1{[0|1|2|3|4|5|6|7|8|9]}11

图片地址：7{字符}255

### 1.3.2用户申请功能 普通用户登录后，从后台数据库获取该用户的相关信息，包括权限及菜单等，权限存储于服务器中，用于对用户每次的请求进行权限判断，菜单树则返回到前端，前端根据此菜单树渲染对应的菜单项。进入系统后点击项目管理，会向后台请求该用户所申请的所有项目，之后点击新增项目，用户可以按照提示填写相关信息来申请项目，前端将项目信息发送到服务器，之后在数据库中添加一条项目记录，此时项目申请成功。再次向服务器请求项目之后，新申请的项目就会添加到已申请项目的列表之中。用户申请功能的泳道图如图1-15所示。

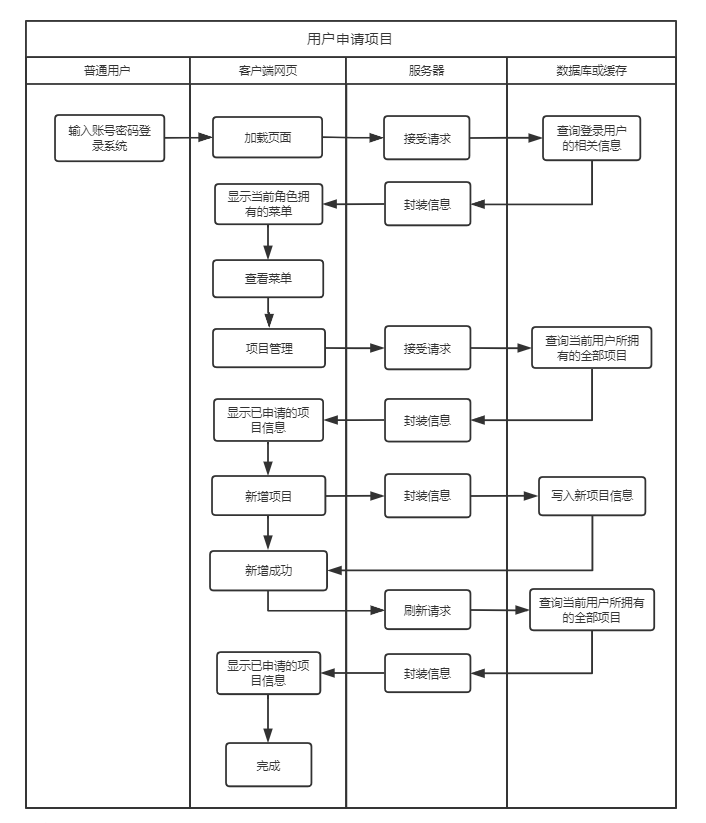


图 1-15 用户申请项目功能泳道图

### 1.3.3管理员分配检测人员功能

管理员进入系统之后，从后台数据库获取管理员的相关信息。管理员点击项目管理可查看未分配项目以及已分配项目，点击未分配项目，将从数据库中查询出未分配检测人员的项目，管理员为一项目点击分配检测人员将查询出数据库中的检测人员信息，选择其一之后数据库会在该项目中插入该检测人员，此时该项目进入已分配项目列表当中。管理员分配检测人员功能的泳道图如图1-16所示。

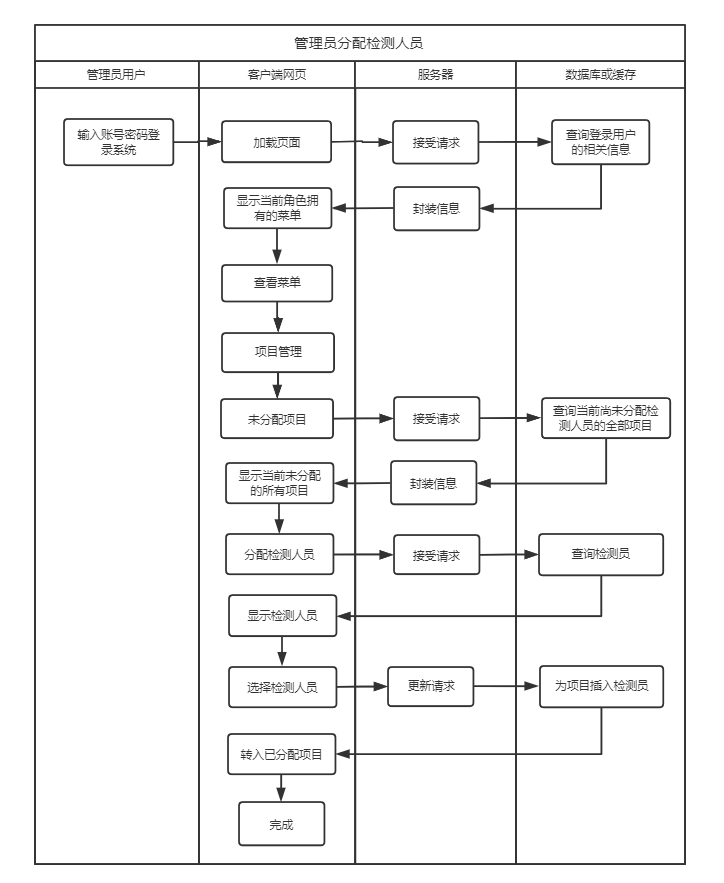


图 1-16 管理员分配检测人员功能泳道图

### 1.3.4检测人员上传更新建筑信息功能

检测人员进入系统后，从后台数据库获取检测人员的相关信息。点击项目管理之后查看待更新项目会向数据库查询带有该检测人员信息的项目，传到前端并显示所有项目。选择一个项目，对该项目中的某一建筑进行检测之后选择上传建筑信息，填写完毕确认提交后，服务器端处理来自网页的信息，将数据写入数据库中，并将该项目标识为正在更新项目，此时服务器端将数据库中的正在更新项目查询出来并返回到客户端在正在更新栏进行展示。当更新完毕后该项目进入已完成项目中，并在数据库中进行标识，如果还需更新则继续重复上传建筑信息，直到更新完毕为止。检测人员上传更新建筑信息功能的泳道图如图1-17所示。

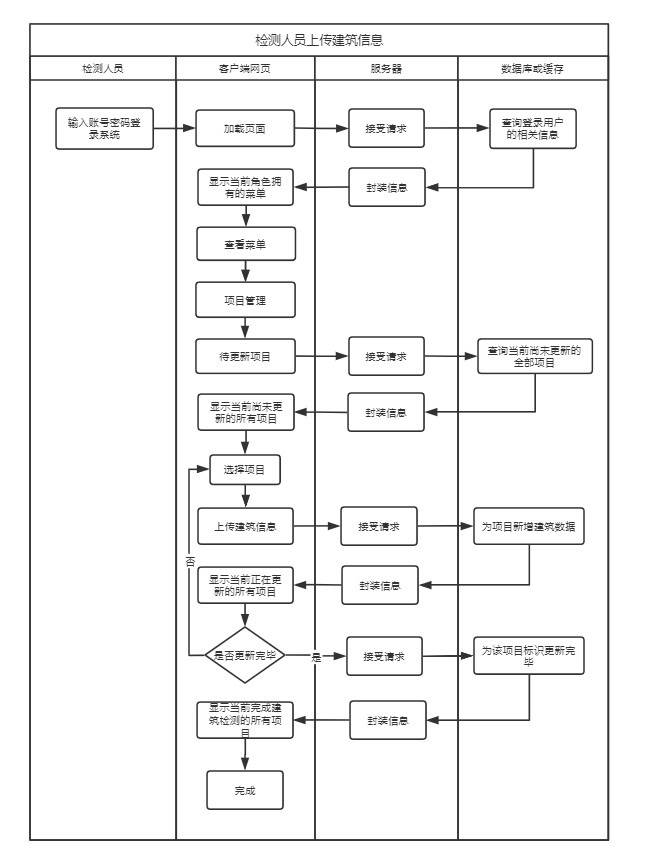
****

图 1-17检测人员上传建筑信息功能泳道图

## 1.4非功能性需求分析

非功能性需求指的是软件系统为满足用户业务需求而必须具有的除了功能需求以外的特性，一般包括系统的性能、可靠性、可维护性、可扩展性以及对技术和业务的适应性等方面的需求，非功能性需求的定义不仅决定着软件的质量，还在一定程度上影响着产品的功能需求能否得到保障。一般来说，系统的非功能性需求主要包含了：

性能，是最常提起的非性能需求，性能指的是软件系统对各种用户行为的相应性能力，主要包括响应时间，吞吐量，系统运行时的资源消耗等。本系统在网络情况较好的环境下，一般要求页面的相应时间不得超过3秒，最大并发数量不得少于50。

可靠性，软件的可靠性是指在规定的一段时间内，系统无故障运行的概率。影响可靠性的因素主要有：软件编码的质量、文档的正确性、硬件设备的稳定性等。本系统要求在持续运行的状态下，系统平均无故障时间应该不少于90天。

健壮性，它是指软件在发生了不合法输入或者错误操作的情况下，系统的功能还可以继续正常运行的可能性。设计一个具有健壮性的系统，需要开发人员提前对系统运行过程中可能出现的各种异常情况进行分析，并做出相应的异常处理，保障发生错误时系统仍然可以继续正常运行。

可扩展性，可扩展性是软件系统的主要设计原则之一，它指的是系统在原来的基础上上添加新功能或者修改完善现有的功能的能力。为了使防雷系统在以后可以快速地添加和修改功能，需要开发者以模块化的方式进行编码，同时要保留详细完整的开发文档，以方便系统的二次开发。

## 1.5可行性分析

设计一个应用软件之前，首先要对它进行多方面的可行性分析，确保软件发的有实际应用价值。下面将从经济上、技术上、操作上以及法律上分析基于前后端分离的防雷系统的可行性。

经济可行性：开发和部署本系统中所需要的集成开发环境、CentOS系统、MySQL、Maven构建工具以及Docker容器等大部分都是开源的项目，可以在相应的官网上免费下载使用的，本系统主要的支出来自服务器的费用，而购买普通云服务器的费用对于餐厅来说是完全可以接受的，因此本系统在经济上可行。

### 技术可行性：本系统是一个基于前后端分离的防雷系统，在技术上主要使用了Java 8编程语言、RESTful软件架构、Spring Boot框架、Vue框架、[MyBatis](http://www.baidu.com/link?url=O4dK3i8g5ijKB9s0Wj7AApN_DE4ZWvQo1vjUT9m5oylTq_GC_OunBMR0lG8RfZdg)技术等技术都已经较为成熟，并且这些技术也是目前软件开发的主流技术，在社区论坛上和相关的书籍中都能够学习并应用这些技术，在技术上是可行的。

操作可行性：本系统功能模块划分清楚，操作简单，用户在使用的过程中不需要额外的指导即可迅速理解掌握系统的使用方式。

法律可行性：本系统使用的开发和部署技术工具完全符合相关的法律法规，在信息的处理和业务执行中没有违反法律规定。

## 1.6小结

本章对防雷系统做了一个简要的概述，也围绕用户角色对本系统的功能需求和用例进行了较为详细的介绍和分析，系统分析是实现系统的必不可少的一个环节。在这个环节对防雷系统有了一个大致的了解，分析了每一个角色在系统中扮演什么角色，每个角色的职责所在，接下来用软件工程的建模工具对系统的用例图进行了细致的分析，使每一个环节的内容更加的清晰、交互更加明确，为后面的系统设计、数据库设计、原型设计以及系统实现打下坚实的基础。

# 2系统设计

2.1 系统设计思路

在开始系统开发之前需要对需求做进一步的落实，根据实际情况规划出实现系统的具体路线，所以需要制定一个系统设计的总体方案。

首先需要搭建开发环境，然后对系统分析中提出的角色权限、功能模块进行分析和设计，根绝分析得出的结果作为参照，设计对应的数据库表结构，以及系统的原型界面，接下来选择合适的技术栈来配合进行系统的代码部分编写，最后对系统进行相应的测试。

开发环境选择[Microsoft](https://baike.baidu.com/item/Microsoft" \t "_blank)推出的针对于编写现代 Web 和云应用的跨平台源代码编辑器visual studio code（VSCode）。角色和模块功能是有密切联系的，不同的角色应该具有的模块功能可能有一些区别，同时同类角色的模块不是孤立的，而是密切关联的，我们需要做更深入的分析，细节处多考虑，在降低模块之间的耦合程度、增强单个模块内聚性的同时，弄清楚模块之间的关系，将其与界面完美的融入系统中，既能给开发带来便利，也能给用户带来不错的使用体验。

数据库设计是系统中非常重要的一环。不同的表结构会导致在做业务需求时所需要的查询方式、SQL语句、查询表的数量等会有一定差异，一个优秀的数据库设计可以为系统开发提供较大的便捷，也可以提升系统的数据存储性能。除此之外还要遵守数据库设计规范，考虑各个表之间的约束和数据完整性，避免冗余。

针对系统中的一些特定环节或功能，可能存在一些专门的技术栈解决方案，或是当前主流的开发方式，可以在系统中采用，因为它们相对成熟，也便于后期维护。

2.2 开发环境搭建

本系统所用到的开发环境如下：

1. 操作系统环境：使用windows 10专业版64位。
2. 开发软件：VScode，VScode是一个轻量级但功能强大的源代码编辑器，它运行在桌面上，可用于Windows、macos和Linux。它内置了对Javascript、Typescript和Node的支持[1]( <https://code.visualstudio.com/docs>)。 IntelliJ IDEA，它是java编程语言开发的集成环境。

③ 数据库：使用MySQL docker镜像，docker是一个容器，它的目的就是运行应用或者服务[2]，通过docker运行MySQL镜像来为系统提供MySQL服务。

④ 项目环境：安装node.js，npm会随同一起安装，npm是包管理工具，用来安装系统前端开发中缺少的包，以及启动前台项目。使用npm安装vue-cli，vue-cli是vue脚手架，它可以轻松的创建新的应用程序而且可用于自动生成vue和webpack的项目模板。

2.3 系统业务流程设计

该系统对用户使用顺序有明显的要求，首先，普通用户进入系统，添加需求，然后管理员进入系统为需求分配检测人员，最后检测人员登录系统领取需求任务，完成对信息的采集，信息的上传，最终结果将在三类用户之中进行同步。

流程图如下：

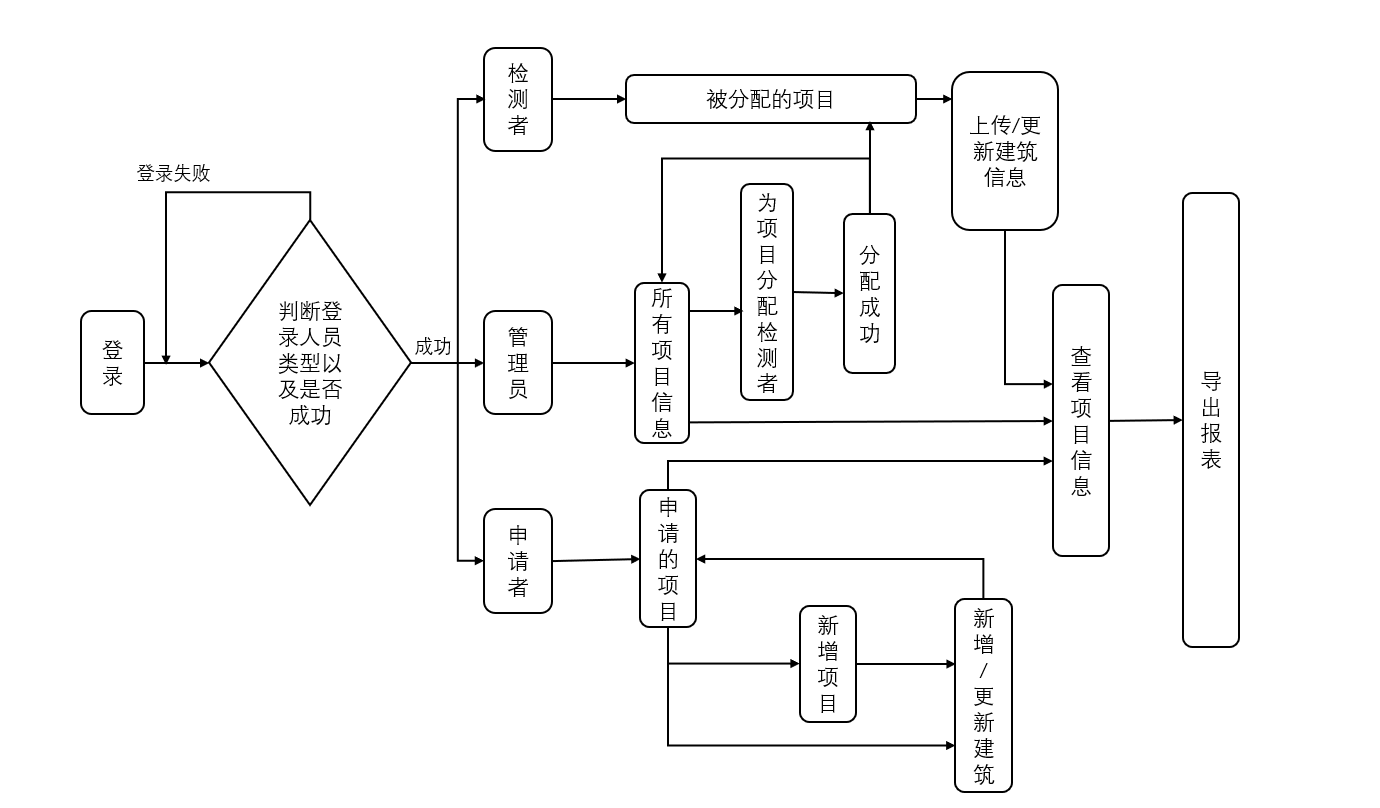


图2-1 系统业务流程图

2.4 系统功能设计

根据以上章节的分析得出以下的功能模块划分：

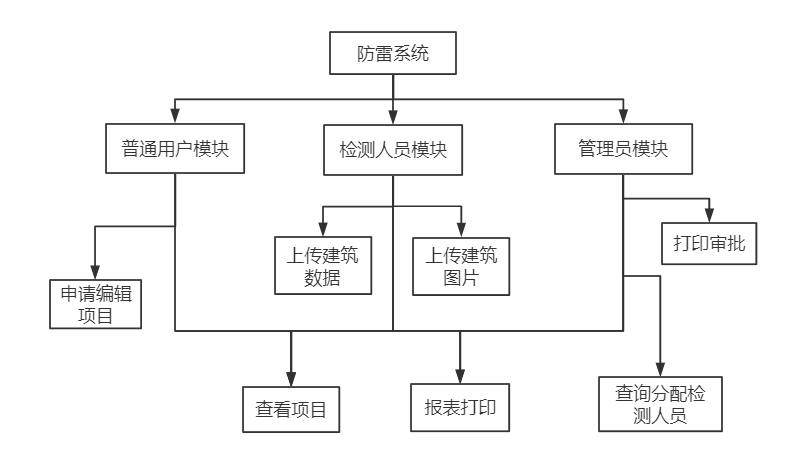


图2-2 系统功能模块图

2.5 数据库设计

项目所涉及的表主要有：用户表、角色表、用户角色表、建筑信息表、建筑检测结果表。

### 2.5.1 数据库分析

我们这里采用了docker来运行MySQL镜像提供数据库服务，运行环境为centos，绝大多数的使用linux操作系统的大中小型互联网网站都在使用mysql，作为其后端数据库存储。除此之外还有很多优点[3]：

1） Mysql性能卓越，服务稳定，很少出现宕机。  
2） 开放源代码，自主性，使用成本低  
3） 历史悠久，社区用户非常活跃，遇到问题可以寻求帮助。  
4） 软件体积小，安装使用简单，并且易于维护

### 2.5.2 概念模型设计

概念性数据模型关心的如何完整、正确地反映客观实际情况，不关心在数据库中如何实现。这种数据模式能够真实地反应用户要求的实际情况，是一种容易被人们理解的直观的数据库结构模式。同时也是一种相对稳定统一的数据模式，一般情况下很少变动。概念性数据在用户和设计者之间建立了桥梁。是设计数据库结构的基础[4]。

概念设计中自顶向下的实体分析方法，即常用的实体联系模型（简称E-R模型），对具体事物数据进行抽象加工，将实体集合抽象为实体类型。用实体间联系反映现实世界事物间的内在联系。E-R模型是建立概念性数据模型的有力工具。该系统的E-R图如下图所示：

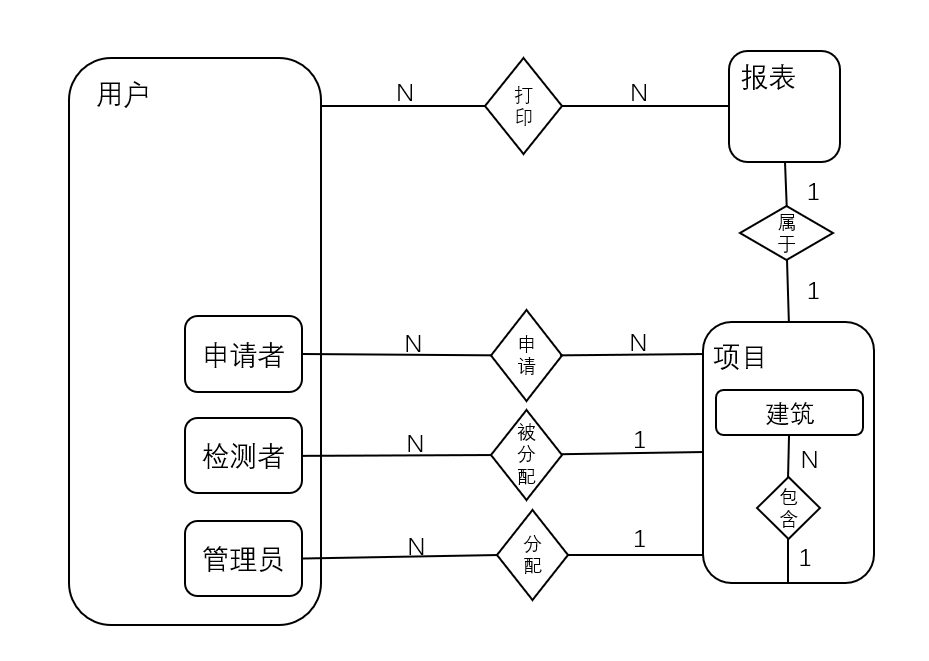


图2-3 E-R图

主要实体的具体信息将在下面的实体图中详细的说明。

用户实体图：

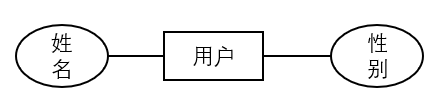


图2-4 用户实体

项目实体图：

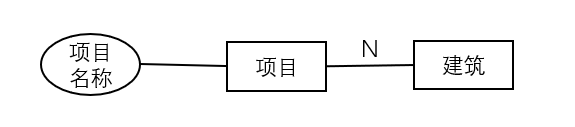


图2-5 项目实体

建筑实体图：

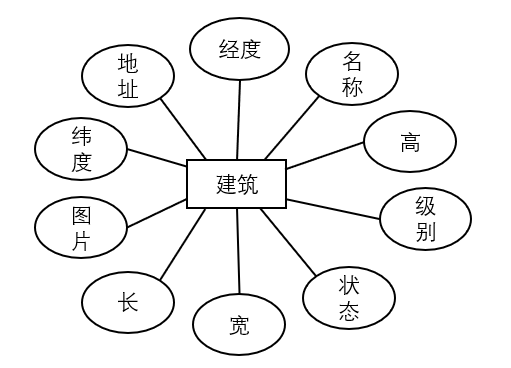


图2-6 建筑实体

报表实体图：

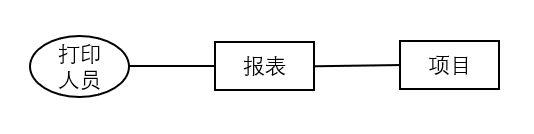


图2-7 项目实体

### 2.5.3 数据库结构设计

设计时，我们将确定下来的实体对象的属性值映射为数据库表中的列，在不同级的表之间，我们按名应用映射父表与子表间的引用关系，同级表之间我们仍然沿用主键、外键来映射它们之间的引用关系。这样我们就建立了一个物理数据模型。

数据库逻辑结构设计：概念结构设计是独立于实际数据模型的信息结构，必须将其转化为逻辑结构后才能进行数据库应用的设计，也就是要将概念上的结构转化为数据库系统所支持的实际数据模型。

第一种转化是将实体转化为关系表。这种转化比较简单，只需要将实体的属性定义为表的属性即可。

第二种转化是联系的转化。即将各实体之间的联系转化为表格之间的关系，如外键的定义。

在以上的工作基础上归纳出系统所需数据库表格的组成、列的属性、不同表之间的联系等。

该数据库系统具有以下多方面的特点：

① 结构合理，对一个用户建立多条记录。

② 所建立的数据库冗余较小，独立性较强。

③ 数据查询、修改、创建快而准确。

④ 可靠性好。

要做到以上要求，在数据库的设计中必须尽量做到满足数据库设计的第三范式要求，也就是满足第三范式的定义：如果关系模式R(U,F)中的所有非主属性对任何候选关键字都不存在传递依赖，则称关系R是属于第三范式的[5]。从第一范式到第三范式是逐次“包含于”的关系。遵循关系数据库设计的范式要求才能达到规范化目的从而使结构更加合理，消除存储异常，减少冗余，该系统数据表最终的结构和关系如下图。

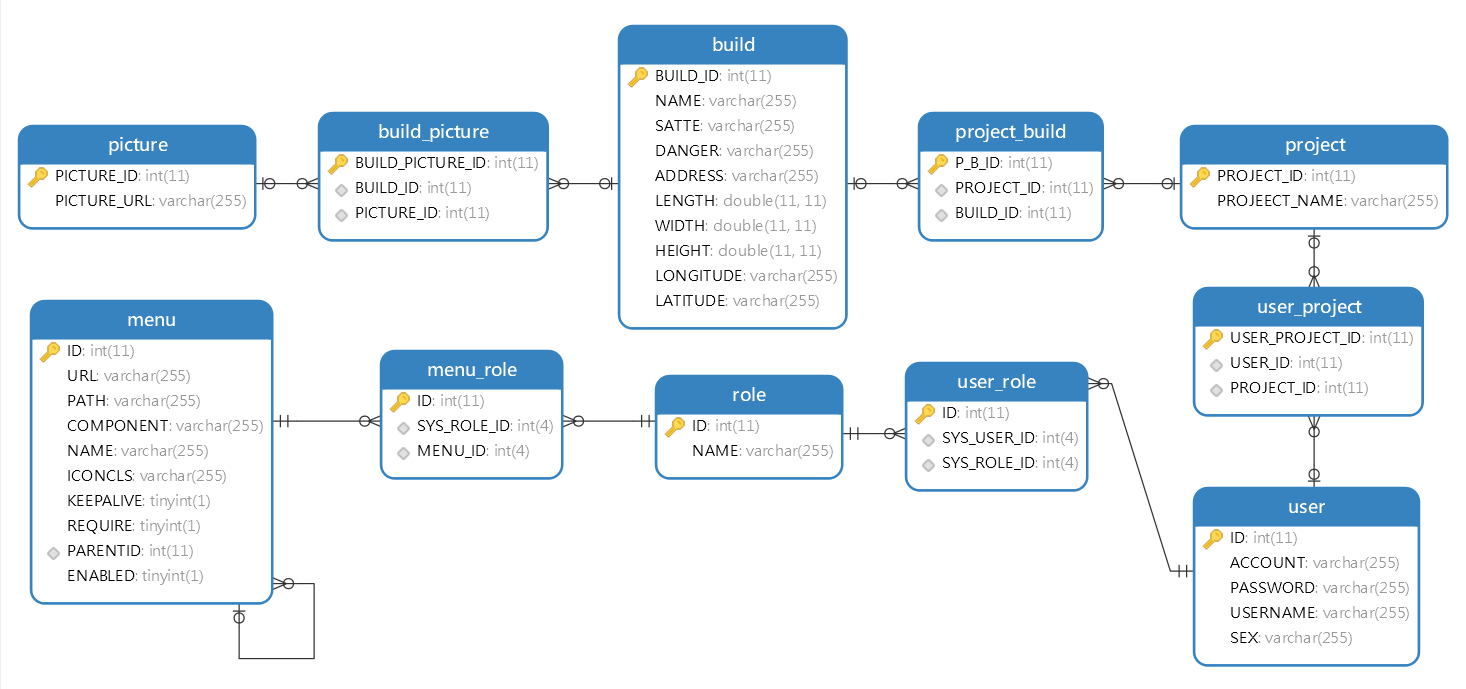


图2.8 数据库结构图

2.6页面逻辑设计

# 3 系统实现