

在线报修系统

项目研发报告

项目组长： 彭滢茹

项目组员： 张琴、刘思岑、
段雯雯、翟微、
车泽宇

2016/07/21

目 录

1 引言.....	0
1.1 编写目的.....	0
1.2 背景.....	0
2 实际开发结果.....	0
2.1 主要功能.....	0
2.2 基本流程.....	1
2.2.1 数据流总图.....	1
2.2.2 level-1 层次结构图.....	1
2.2.3 level-2 层次结构图.....	4
2.3 进度.....	6
3 开发工作评价.....	7
3.1 对生产效率的评价.....	7
3.2 对产品质量的评价.....	7
3.3 对技术方法的评价.....	7
3.4 技术难点.....	7
4 开发过程中所使用的工具.....	9
5 经验与教训.....	10

1 引言

1.1 编写目的

项目开发的在线报修系统主要为相应的企业设计，方便派单员和管理员对报修单和用户进行管理、工程师和销售人员能够查看自己的报修单并进行一系列职责内的操作。

1.2 背景

在计算机普及、网络技术发达的今天，使用联网的系统代替人工来做重复繁杂的信息统计以及人员管理工作已经成为主流。可以运行在多平台上系统无疑将会为企业将来用于管理信息和进行人员分配的便利工具。

目前已经在社会上已经存在一系列类似的信息管理系统，它们很多是基于电脑端运行的系统，而对于报修系统来说，用户多为工程师、派单员等，可能并不方便随身携带电脑来登录系统进行操作，因此将报修系统设计成可以同时运行在多平台上的手机应用是十分必要的。

为了使在线报修系统更具有灵活性，我们采用 **Activiti** 业务流程管理的框架来对在线报修流程进行设计，使得我们的在线报修系统相比市面上其他在线报修系统，结构更加清晰简洁，便于系统的维护以及将来系统功能的增加。在前端方面，我们采用 **ionic** 这一轻量级框架对手机应用进行开发，使得前端开发多平台的手机应用更加简洁便利，同时也有利于将来的维护。

2 实际开发结果

2.1 主要功能

- | | |
|-------------------|------|
| 1. 用户登录、注销、修改密码 | (完成) |
| 2. 派单员新建报修单 | (完成) |
| 3. 用户按分类、按顺序查看报修单 | (完成) |
| 4. 用户查看报修单详情 | (完成) |
| 5. 派单员修改报修单内容 | (完成) |

- | | |
|-----------------------------|------|
| 6. 派单员、管理员删除报修单 | (完成) |
| 7. 工程师对未接报修单进行接单 | (完成) |
| 8. 工程师对已接报修单进行取消接单或提交完成 | (完成) |
| 9. 销售员对已完成报修单进行审核 | (完成) |
| 10. 用户点击电话信息直接拨打客户电话 | (完成) |
| 11. 消息自动通知 | (完成) |
| 12. 用户查看分值统计 | (完成) |
| 13. 管理员按角色查看、删除、新增用户并设置初始密码 | (完成) |
| 14. 派单员登录后台管理网站新建报修单 | (完成) |
| 15. 后台管理网站对客户信息进行提示输入 | (完成) |

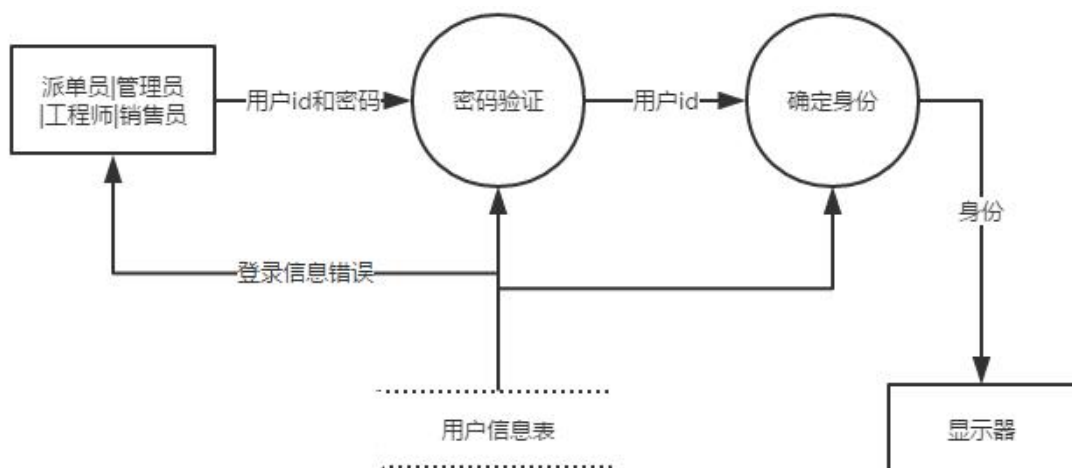
2.2 基本流程

2.2.1 数据流总图

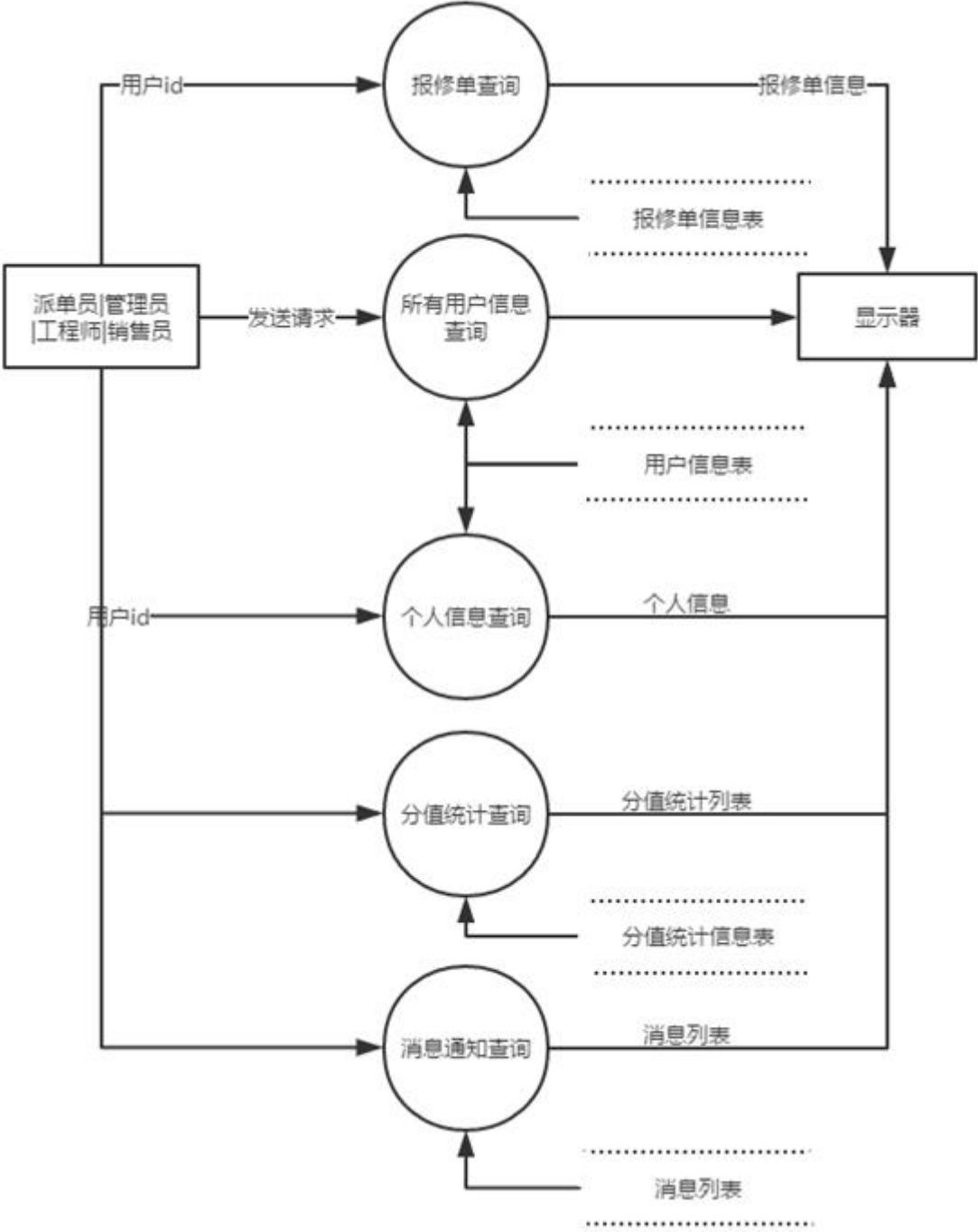


2.2.2 level-1 层次结构图

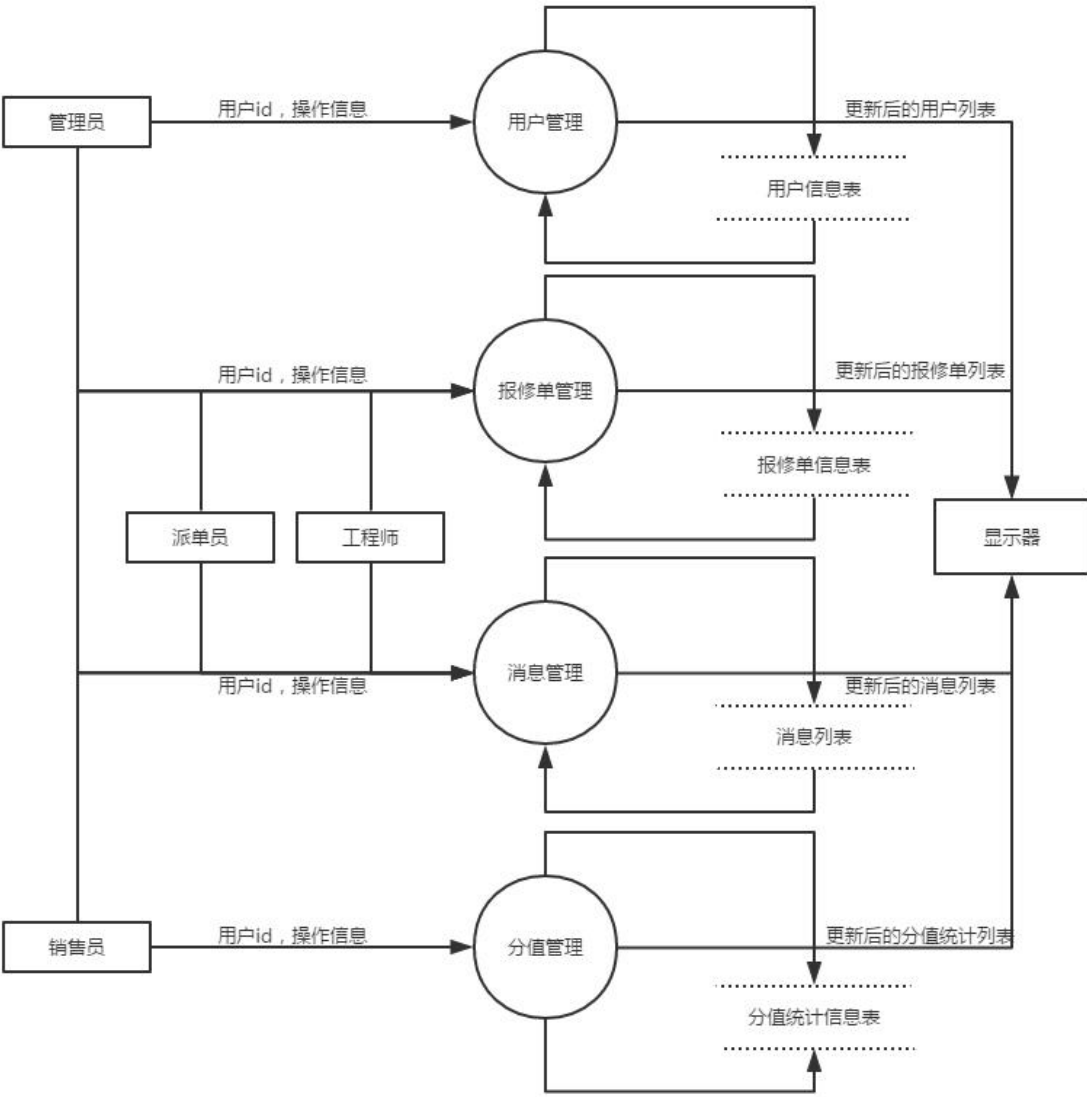
2.2.2.1 登录子系统



2.2.2.2 查询子系统

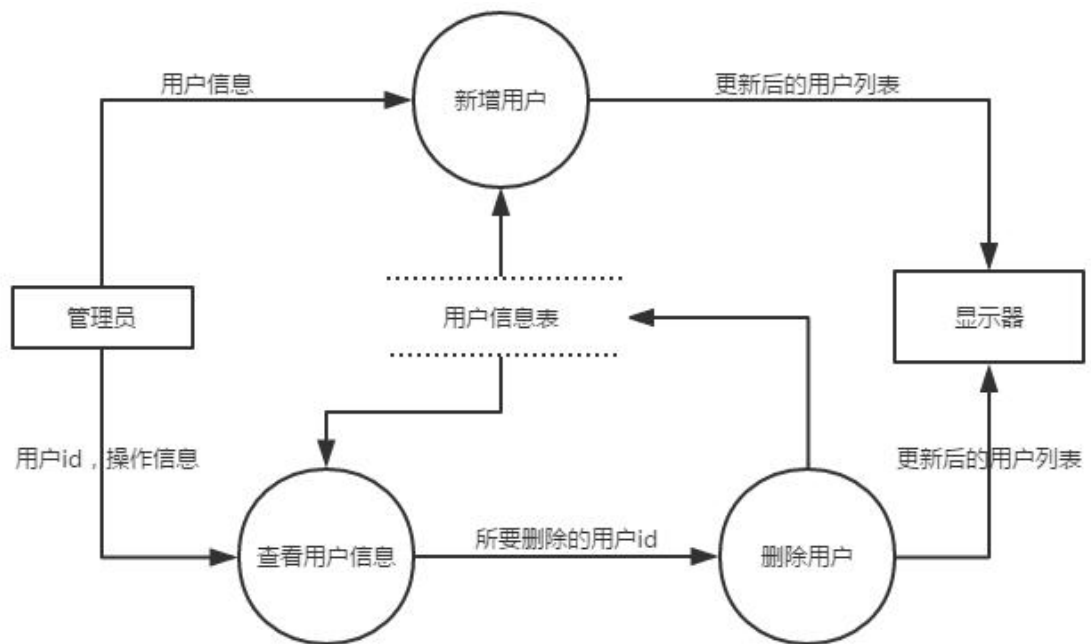


2.2.2.3 管理子系统

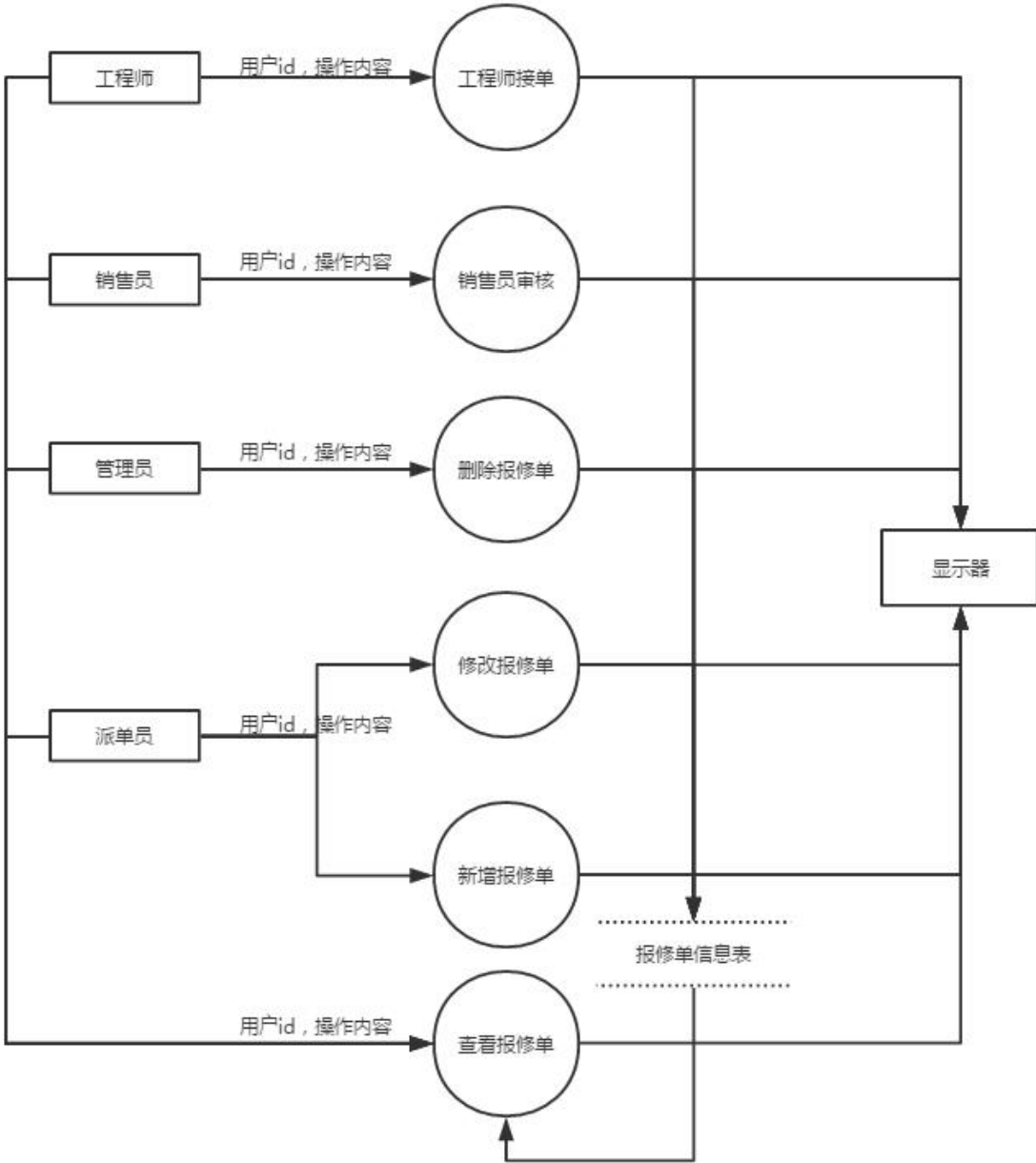


2.2.3 level-2 层次结构图

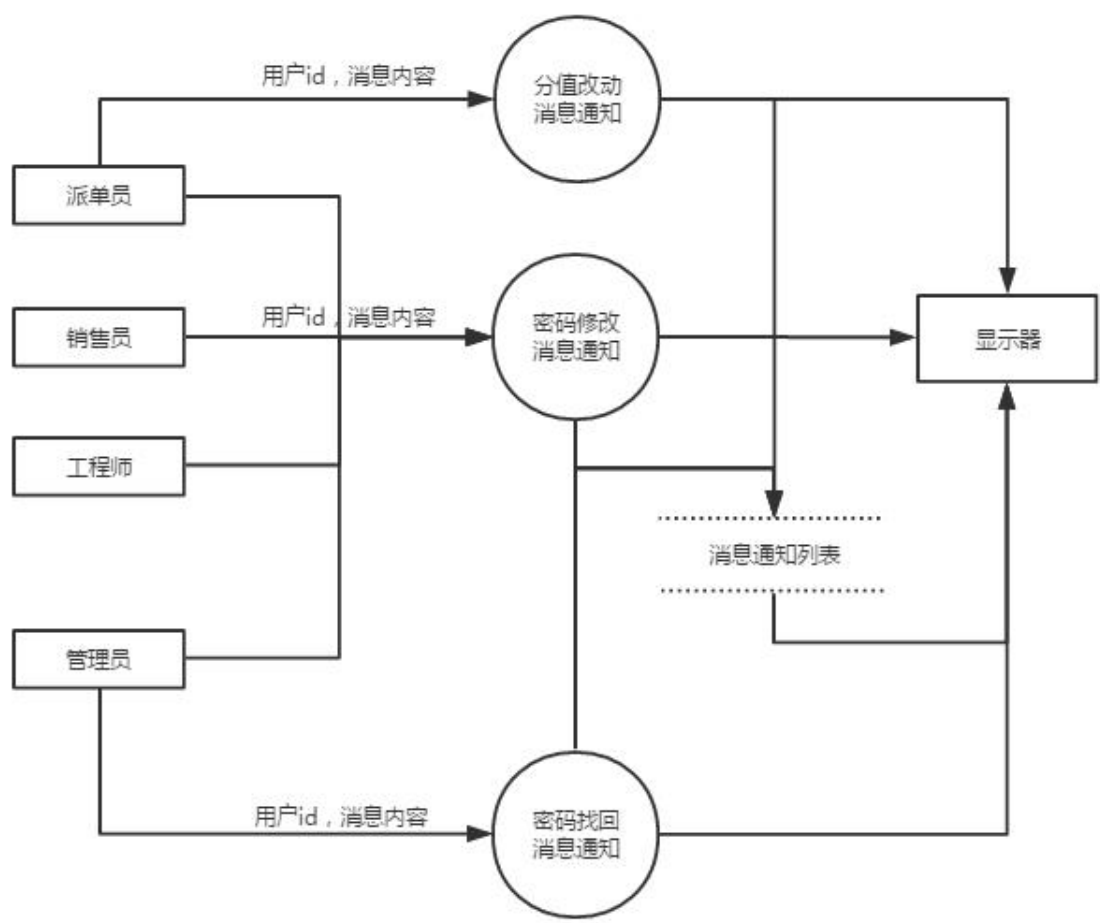
2.2.3.1 用户管理



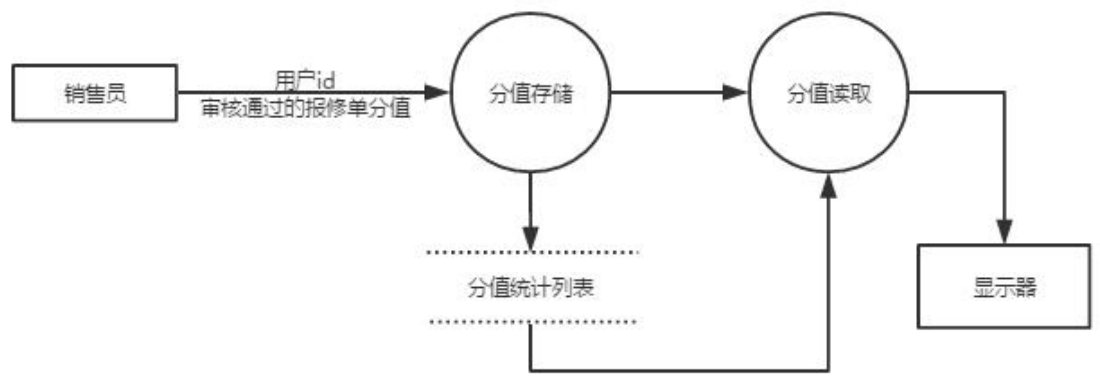
2.2.3.2 报修单管理



2.2.3.3 消息管理



2.2.3.4 分值管理



2.3 进度

工程按照预期的进度如期进行，进度按照计划不断推进，现已完全完成。

3 开发工作评价

3.1 对生产效率的评价

程序代码一共 11096 行（由于 css 代码大部分并非我们自己设计，因此没有计算 css 代码，加上 css 代码后大概 25000 左右）。

平均每人 3700 行代码，历时 3 个月。

效率达到预期的水平。

3.2 对产品质量的评价

产品质量良好，测试中出现的 BUG 已经修复。至今未发现在实际的使用过程中出现过重大 BUG，也未出现过系统崩溃等情况。

3.3 对技术方法的评价

使用了如下的工具：

编译测试环境：IntelliJ idea

数据库：h2

网络服务器：<http://115.159.225.109> 腾讯云服务器

代码版本管理 Github

技术方面，后台管理使用了 Activiti 业务流程管理框架对事件响应进行包装，移动端前端使用 ionic-AngularJS 框架开发多平台的手机应用，网页端前端使用 bootstrap+jQuery 进行开发。

3.4 技术难点

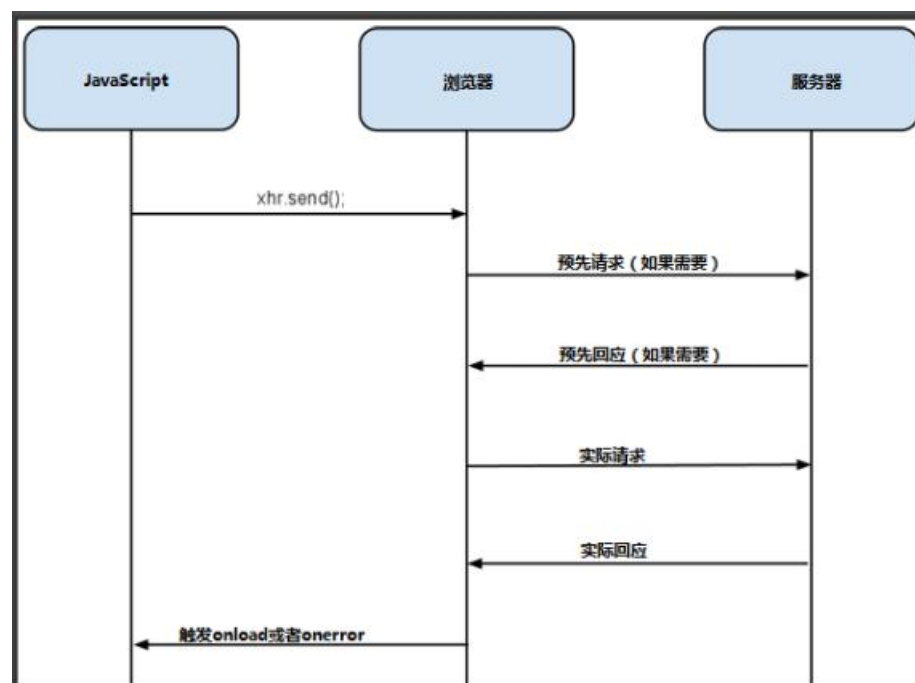
项目中前端和后台所运用的技术均是我们项目组初次学习的，因此在项目进行的过程中遇到了许多技术难点，以下举例列出几点技术难点：

1. 前后端交互问题

问题描述：我们采用了 spring-boot 封装的 rest 接口，这个依赖包里面自动

包含了身份认证，但是在我们将其布置到服务器上以后，这个身份认证与跨域两个问题结合到一起就产生了不可调和的问题，网络访问报错 401 Unauthorized.

问题原因：我们查询了各种资料，百度，google 等，虽然找不到一个非常符合我们情况的描述，但是我们还是获取到了一些有用的信息。我们猜测，当我们从前端发送数据包到后端时，由于跨域，发送的 POST 数据包会变为 OPTIONS 但是浏览器在处理 options 包这种预先请求的时候会丢掉认证的信息，这个时候预先回应就会给我们 401 错误。至此，数据传输失败。



解决方案：我们舍弃了 Spring boot 封装的 rest 接口，这样我们就跳过了认证的必要，所以我们只需要处理跨域的问题了。在服务器的接收加上跨域的头部就行了。问题就解决了，但是它有一个极大的弊端，意味着我们不能使用 activiti 内部提供的用户身份认证。

2. 前端得到数据后以列表形式展示，并能够分类按顺序查看：AngularJS 中自带的 `ng-repeat` 功能可以将数组里面的数据以列表的形式展示出来，并且自带过滤器 (`filter`) 功能，用于对所取数据进行格式化处理。

3. 流程图的整合

问题描述：我们的业务数据结构太过于复杂，所以我们使用的是 Jpa 架构的自身实现的数据库的表，跟 activiti 运行过程中的数据库是不一样的。所以如何将流程图整合到我们的业务中是一个很严重的问题。

问题解决：因为我们没有使用 spring boot 提供的 rest 接口，我们没有使用内

部的用户管理，所以我们通过 `activiti` 实现的表达式信息来对内部的数据库进行管理。相当于将 `activiti` 作为一个中间平台。

4 开发过程中所使用的工具

1. 开发语言

技术描述：java

软件工具：intelij idea

思考：这是由将要使用的一系列技术来决定的

2. 开发框架

技术描述：spring boot

软件工具：配合 intelij idea 开发环境

思考：用来集成各项技术

3. 依赖管理

技术描述：maven

软件工具：IDE 中的 maven 插件

思考：解放了我们对于依赖项的管理和配置

4. 服务器传输

技术描述：scp

软件工具：winscp

思考：这是 windows 系统下的一个软件，用来将我们生成的服务器 jar 包传输至云端服务器，方便的操作提高了我们工作的效率。

服务器远程登录

技术描述：ssh

软件工具：putty

思考：这是一个小巧的远程连接的工具，方便我们后端对服务器远程的登录和部

署。

5.网络封包分析

技术工具：wireshark

思考：这是一个对网络封包进行分析的工具，我们一开始在前后端连接的时候，遇到了一个涉及到身份验证和跨域的问题，我们使用这个工具来对前后端的数据包进行抓取和分析。

6.版本控制

软件技术：git

软件工具：github，sourcetree

思考：这是我们项目最重要的一个环节，前后端在 github 上不同的分支中进行工作，互不干扰，每次项目有任何的进度，都会往 git 仓库中提交新的版本，可以这么说，git 记录完全可以视为我们的一个工作日志。

7.文件共享

软件工具：teambition、QQ 群文件

思考：项目开发的前期一直都将项目共享的资源文件放在 QQ 群文件中，后来升级至 teambition 中去了，我们任何的与项目相关的资源文件都是在这样的平台上进行共享和讨论的，举个例子，我们的前后端接口约定文件就上传到了这个平台，每次后端对接口的改动和重新设计，都能在第一时间通知到所有的团队成员。

5 经验与教训

通过本次工程，我们初步掌握了软件过程设计的方法与步骤，对于软件工程的整体流程也有了初步的把握。

在实践过程中，我们体会到了在正式进行代码编写之前的需求设计，概要设计和数据库设计的重要性。中间也发生过由于开始的数据库设计考虑不够周全，导致代码编写到一半又要去修改数据库的问题等，这些教训让我们更深入的明白了事先做好准备工作的重要性的。

此外，软件工程并非一人之力可以完成，团队分工和协作也是十分重要的，

每个人各司其职。RRC，Git 等软件工程相关的工具也在团队合作的过程中起到了很多的帮助。

最后，开发过程中人员的分配也是很重要的，好的人员分配能够让项目在实施的过程中事半功倍。所以开发经验对于这点真的尤其的重要。

通过此次的软件工程我们受益匪浅，相信在今后的学习和工作中，这次实践所积累的经验都能发挥很大的作用。