哈尔滨工业大学国家示范性软件学院

《面向服务的软件系统》大作业

项目题目：校园设备报修平台

项目组成员：

姓名	学号
袁野	1190200122
詹儒彦	1190202307
金俊秀	1190202212

完成日期：2021 年 12 月 25 日

1. 选题

1.1 作业题目

题目二：基于微服务框架开发项目——校园设备报修平台

1.2 题目概述

在校园中，大部分公共场所人流量大，设备使用频繁，设备经过大量使用后，常常容易损坏，而损坏的设备常常会给用户带来不便，且影响用户体验。而设备损坏由于没有人通知维修人员，常常会使维修滞后，直到例行检查时才会使损坏暴露，而有一些损坏有时不能被常规检查发现，故一个成体系的及时反馈的设备报修系统显得尤为重要。

本大作业所想设计的报修系统很好的解决了这些问题，更能实实在在的让同学成为学校的建设者、维护者。并能使损坏的设备快速通知到维修人员手上并得到维修。

本项目计划实现一款以在校师生为主要服务对象的校园设备报修平台。

2. 实现方案

2.1 约束

在前端，我们使用 WXML、WXSS、JS、HTML5 开发微信小程序；

- 后端使用 Spring 框架、MySQL 数据库进行开发；
- 使用 dubbo RPC 实现服务的输出和输入；
- 使用 zooKeeper 进行项目中的服务管理。其中 ZooKeeper 注册中心通过 Docker 安装在虚拟机中。

2.2 用户分析

用户身份分为师生、维修人员、管理员三种。

- 师生为系统的主要受众，可以随时向系统上报校园内的设备问题；
- 管理员可以对用户进行修改维护，对设备信息进行维护；
- 维修人员可以针对师生的上报内容对设备进行检测维修，更改设备状态信息

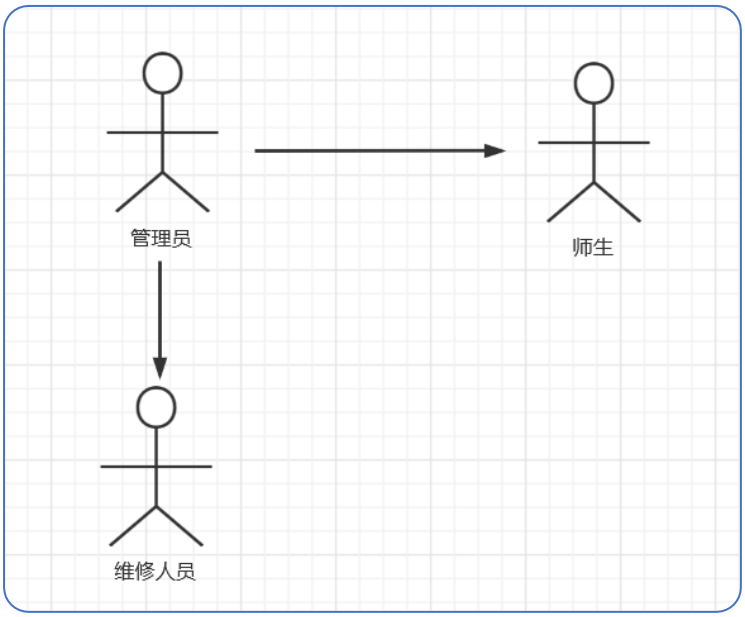

图 2-1 执行者分析

2.3 功能性分析

实现方案按照用户类型分为三个模块：

1. 管理员模块：

- 普通用户、维修人员的增加、删除
- 设备种类的增加
- 设备的增加、删除、通过设备编号查询设备详细信息
- 查询某一个普通用户的所有报修记录
- 查询某一个维修人员的所有维修记录
- 查询某一台设备的所有报修维修记录
- 查询设备列表

2. 普通用户模块：

- 根据设备编号查询设备的维修事件历史（报修人身份不予展示）
- 查询当前登录用户的所有报修事件
- 报修（分为常见故障及排除方式提供与故障上报两步）
- 查询设备具体信息
- 查询设备列表

3. 维修人员模块：

- 根据设备编号查询设备的维修事件历史（报修人身份不予展示）
- 查询当前登录用户的所有维修事件
- 查看所有待维修事件列表
- 维修（即修改事件状态和对应设备状态）
- 查询设备具体信息
- 查询设备列表

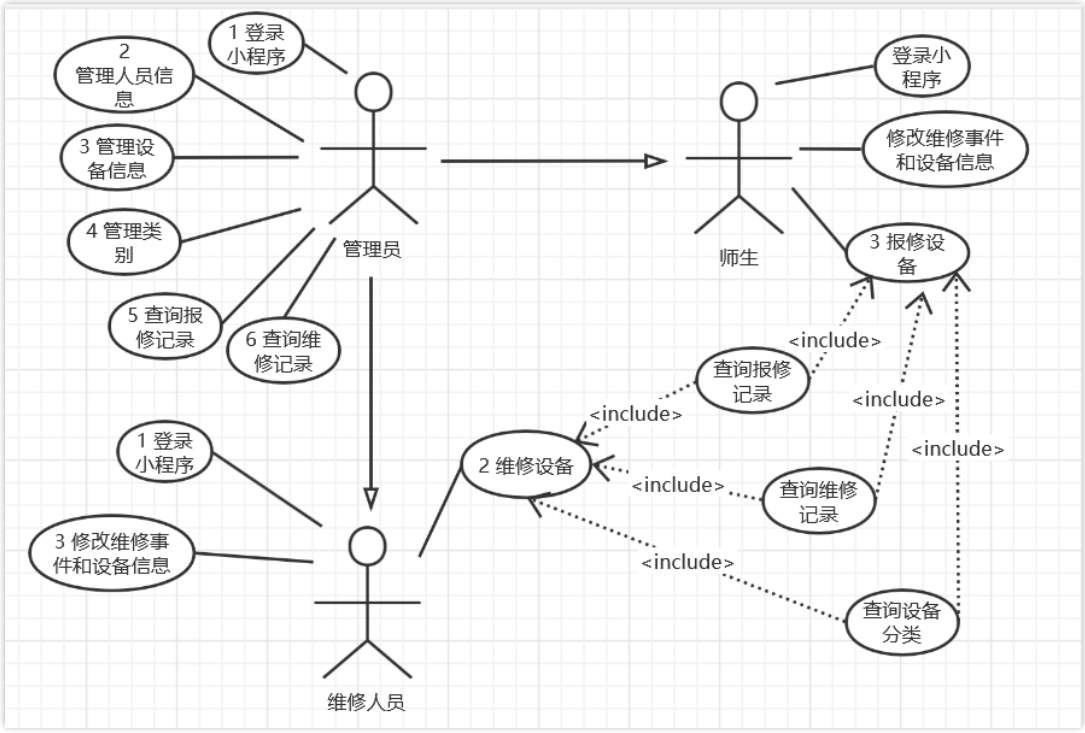

图 2-2 总用例图

2.4 代码构造：

本项目运用 MVC 三层架构编写：

上层 Controller 层负责连接外部接口与下层的 service 接口，连接外部接口封装成 json 形式的数据发送；

中层的 mapper 接口层负责给上层提供功能接口，Dao 层的接口给 service 层提供数据接口，Entity 层直接操作数据库实现对数据库的基本操作。数据库在数据库设计，数据层 pojo 层给上层的 service 提供 4 个接口，每一个表对应一个接口，每一个接口对应一个实体类负责直接实现对数据库执行相应的功能。

数据库中的 4 个表对应 mapper 中的 4 个实体类，可以实现基础的数据库的操作，查询某个属性值或是设置某个属性的值到数据库，这部分直接与数据库相连。

pojo 层的接口与 mapper 相对应，负责给上层的 service 层提供相应的接口来实现相应的功能。

service 负责实现各个页面的接口的功能，并将处理的返回给上层的

controller 层。

controller 负责接收来自小程序前端的 url 的数据请求并调用 service 层的接口将数据封装成 json 串的形式并返回给小程序前端。

2.5 数据库设计

列名	类型	属性
c_id	Integer	设备类型编号
c_name	String	类型名称
text	String	相关文字性描述，如常见故障及修理方法等
e_num	Integer	该类型下的设备数量

2.5.1 设备类型 category

列名	类型	属性
e_id	Integer	设备 ID
cate_name	String	设备所属种类
text	String	地址等其他信息
statement	Integer	设备状态： 0 表示正常 1 表示待检修

2.5.2 具体设备个体 equipment

列名	类型	属性
u_id	Integer	用户 ID
u_name	String	用户姓名
password	String	用户密码
statement	Integer	用户身份 1：师生 2：维修人员 3:系统管理员

2.5.3 用户 user

列名	类型	属性
ev_id	Integer	报修事件
user_id_stu	Integer	报修学生 ID
user_id_pro	String	修理人员 ID， 如果未检修则为空
eq_id	Integer	被维修设备编号
statement	Integer	该事件状态： 0 表示已报修未检修，1 表示检修完成

start_time	String	报修时间：年月日时分秒
end_time	String	修理时间：年月日时分秒
text_stu	String	故障描述
text_pro	String	维修备注

2.5.4 用户 user

3. 实现结果

3.1 主界面

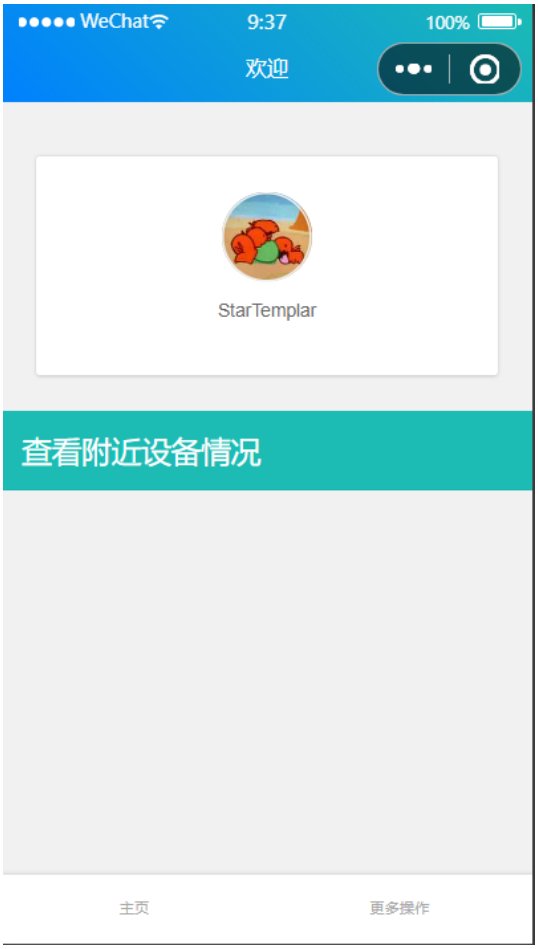

图 3-1 主界面

3.1.1 查看附近设备情况

图 3-2 查看附近设备情况

本界面展示了所有设备的信息，以及工作正常还是待检修。

3.1.2 登录

输入用户名和密码即可登录。用户分为师生、维修人员和管理员。

图 3-3 登录界面

3.2 师生界面

图 3-4 师生界面

列举了师生可以进行的操作。

3.2.1 查询我的报修记录

图 3-5 查询报修记录

用来查询本用户的所有报修记录。

3.2.2 我要报修

图 3-6 报修界面

在这里可以选择损坏的设备，选择之后会有“尝试使用此方法解决”的信息。如果仍需报修，则可以点击提交反馈来提交设备损坏信息

3.2.3 提交反馈

图 3-7 提交反馈界面

在这里可以对损坏设备提交反馈。需要指定设备并写好备注再提交。如果之前有人对同一设备提交过反馈且未被修理，则提交失败。

3.3 维修人员界面

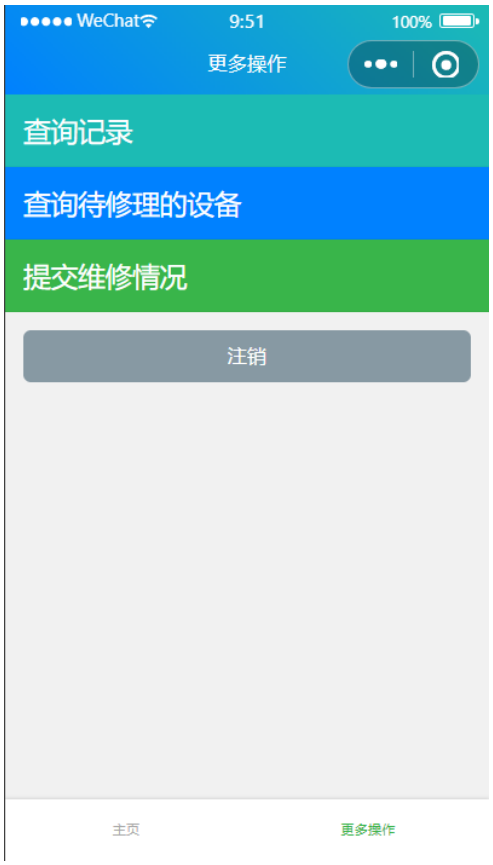

图 3-8 维修人员界面

列举了维修人员可以进行的操作。

3.3.1 查询记录

图 3-9 查询维修记录

用于查询用户的所有维修记录。

3.3.2 查询待修理的设备

图 3-10 查询待修理设备

用于查询所有已报修未修理的设备。

3.3.3 提交维修情况

图 3-11 提交维修情况

用于维修人员修理设备后提交修理情况。选定设备种类和已报修且已维修设备 id 后，填写维修备注即可提交，并把该设备的情况更新为可正常使用。

3.4 管理员界面

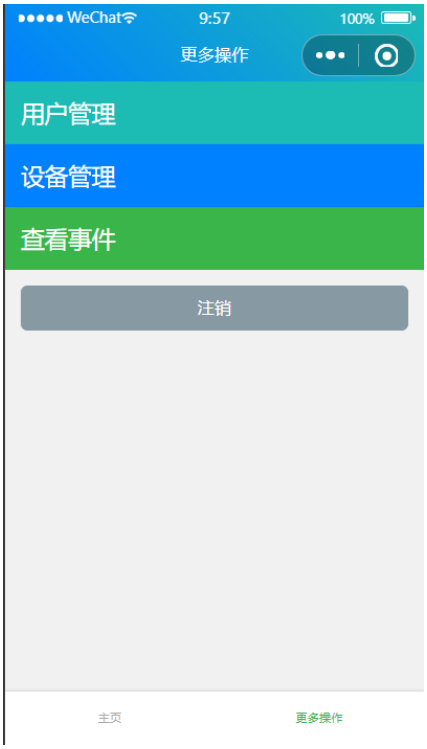

图 3-12 管理员界面

列举了管理员可以进行的操作。

3.4.1 用户管理

图 3-13 用户管理界面

在用户管理界面，管理员可以选择添加特定身份的用户或删除一个用户。

3.4.2 设备管理

图 3-14 设备管理界面

管理员可以添加新的设备种类、设备个体或删除设备。

3.4.3 查看事件

图 3-15 查看事件界面

可以查看任意用户的报修记录或维修记录，如下图：

图 3-16 查看维修记录

3.5 docker 与 zookeeper 交互

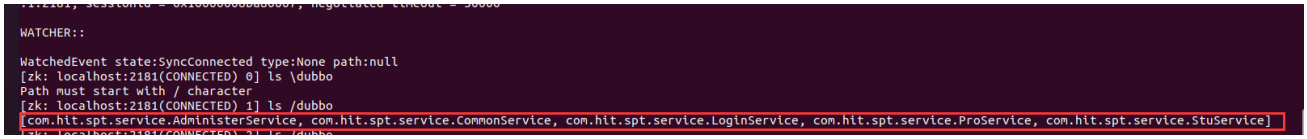

图 3-17 提供者成功注册到虚拟机 zookeeper

图 3-18 docker 拉取一个 zookeeper 注册中心的截图

4. 总结

本次大作业中，我们通过 Springboot 微服务框架和 MySQL 数据库开发后端程序，并使用 Dobbo 实现服务的输入和输出，使用 ZooKeeper 对项目中的服务进行管理。通过这次大作业，我们对微服务有了更深刻的理解，学会使用各种工具辅助开发，并且通过实现前后端的交互，获取了一些前后端分离的开发模式的经验。同时，我们也对课程主要内容产生了更深刻的理解，对面向服务的软件开发有了基本的概念。