

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS



论文题目 空气质量传感器管理系统设计与实现

学生姓名 孙 逢

学生学号 5120309032

指导教师 吴刚副教授

专 业 软件工程专业

学院（系） 软件学院

上海交通大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：某某

日期：某年某月某日

上海交通大学 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密 ☐，在 _____ 年解密后适用本授权书。

本学位论文属于

不保密 ☒。

(请在以上方框内打✓)

学位论文作者签名： 某某

指导教师签名： 某某

日 期： 某 年 某 月 某 日

日 期： 某 年 某 月 某 日

空气质量传感器管理系统设计与实现

摘 要

近几年，空气质量问题已经逐渐成为中国乃至全世界的社会焦点，但大多数地区只有政府部门会有监测站和监测点，普通民众并不能实时了解自己身边的空气质量。Clarity Movement Co. 研发出了火柴盒大小但精度可媲美光学空气质量分析仪的 PM2.5 传感器，并通过与企业和政府合作在传统硬件设备和城市建筑上搭载传感器来销售传感器。公司为之搭建了功能强大的云服务平台，可以实时地处理和推送空气质量数据，但还需要一个信息系统来操作和管理传感器。

在此背景下，本课题调研了大量 JS 框架和库基于现代主流的 WEB 开发技术为之开发了一个传感器管理系统。系统包含三个模块，分别是版本管理、Smart Home 和 Smart City 模块。版本管理模块面向公司的硬件开发人员，让他们能够方便地管理自己开发的传感器及其版本、批次、兼容性和所有者。Smart Home 模块面向家电制造业的合作企业，让其能够使自己的家电互相联系起来。Smart City 模块面向政府合作者，让其能够直观地从不同角度查看和下载空气质量数据。

关键词： 空气质量 传感器 信息系统 WEB 开发

Design and Implementation of Air Quality Sensor Management System

ABSTRACT

In recent years, air quality has gradually become the a social focus of Chinese and even the world. However, in most regional only governments have monitoring stations and sites and ordinary people can not get real-time air quality on their side. Clarity Movement Co. developed a matchbox-sized PM2.5 air quality sensor whose accuracy is comparable to the optical analyzer and sell the sensors by mounting them in the traditional hardware and city buildings in cooperation with businesses and governments. The company had built a powerful cloud service platform which can handle and push real-time air quality data but still needed an information system to manage and operate the sensor.

In this context, this paper investigates a large number of JS frameworks and libraries to develop a sensor management system based on modern mainstream web-development technology. The system consists of three modules, namely, Version Management, Smart Home and Smart City. Version Management module is for the company's hardware developers to easily manage the versions, batches, compatibilities and owners of the sensors. Smart Home module is for cooperative enterprises who are in appliance manufacturing to be able to link their appliances to each other. Smart City module is for government partners to visually view and download air quality from different angles.

KEY WORDS: Air quality, Sensor, Information System, Web-development

目 录

插图索引	vi
表格索引	vii
第一章 绪论	1
1.1 课题背景和研究意义	1
1.2 课题主要研究内容	1
1.2.1 版本管理模块	1
1.2.2 Smart Home 模块	2
1.2.3 Smart City 模块	2
1.3 论文组织结构	2
第二章 WEB 设计开发理论、技术和工具	4
2.1 设计理论	4
2.1.1 敏捷开发	4
2.1.2 Web Components 组件标准	4
2.1.3 Material-design UI 设计	4
2.1.4 Flex 布局和 Responsive 设计	4
2.1.5 RESTful API 设计和 Web Socket 数据更新	4
2.1.6 Flux 架构模式	4
2.2 技术架构	4
2.2.1 前端: Angular VS React	4
2.2.2 后端: Express VS Koa	4
2.2.3 数据库: MongoDB 和 Mongoose	4
2.2.4 服务器: AWS 云服务	4
2.3 开发工具	4
2.3.1 IDE: Webstorm VS Atom	4
2.3.2 版本控制: Git 和 Git Flow	4
2.3.3 代码生成: Yeoman 和 Yeoman Generators	4
2.3.4 文档生成: JsDoc VS EsDoc	4
2.3.5 代码质量: Lint 工具和 Git hooks	4
2.3.6 编译工具: Grunt VS Gulp VS Npm	4
2.3.7 单元测试: Mocha 和 Karma	4
2.3.8 持续集成: Travis-CI VS Solano	4

第三章 可行性分析	5
3.1 技术可行性	5
3.2 经济可行性	5
3.3 操作可行性	5
第四章 需求分析	6
4.1 功能目标	6
4.2 用户用例	6
4.3 快速原型	6
4.4 数据字典	6
4.5 性能需求	6
4.6 用户运行环境	6
第五章 系统设计与开发	7
5.1 总体设计	7
5.2 版本管理模块	7
5.2.1 前端	7
5.2.2 后端和数据库	7
5.3 Smart Home 和 Smart City 模块	7
5.3.1 前端	7
5.3.2 后端和数据库	7
5.4 主要组件详细设计	7
5.4.1 客户端 ORM 表单组件	7
5.4.2 EChart 组件	7
5.4.3 AqiChart 组件	7
5.4.4 AqiMap 组件	7
5.4.5 API 组件	7
第六章 系统最终实现效果	8
6.1 版本管理模块	8
6.1.1 前端	8
6.1.2 后端和数据库	8
6.2 Smart Home 模块和 Smart City 模块	8
6.2.1 前端	8
6.2.2 后端和数据库	8
6.3 主要组件详细设计	8
6.3.1 客户端 ORM 表单组件	8
6.3.2 AqiChart 组件	8
6.3.3 AqiMap 组件	8

第七章 系统测试和部署	9
7.1 开发环境配置	9
7.2 测试环境配置和持续集成	9
7.3 运行环境配置	9
第八章 总结与展望	10
8.1 工作总结	10
8.2 工作展望	10
谢 辞	11

插图索引

表格索引

第一章 绪论

1.1 课题背景和研究意义

随着“雾霾”、“PM2.5”等话题的升温，空气质量问题逐渐成为人们关注的焦点，出门不仅要看“天气”还要看“空气”。虽然如今各式各样的空气质量网站、APP 层出不穷，有实时的也有预测的，但大多数数据来源于政府环境部门监测站，少部分来自网站自己搭建的监测点。然而空气质量与天气不同之处在于，它受时间和空间的变化影响更大，对人们健康的影响也更大。因此，如何满足个人用户对周边空气质量的实时掌控成为一大难题。

Clarity Movement Co. (以下简称 Clarity) 是一家研发“世界上最小的空气质量传感器”的创业公司，目前完成研发的第二代传感器尺寸与火柴盒差不多大，便携度大增的同时，精度依旧不弱于光学空气质量分析仪。Clarity 的主要销售途径是与相关企业和政府合作，将传感器搭载在传统硬件设备和城市建筑上。Clarity 现在依旧组建了自己的云服务平台，传感器通过手机蓝牙或者 Wi-fi 上传空气质量数据，服务器处理并保存相关数据。然而，要让该产品能够为大众所用，配套的软件系统还未完备。

随着业务的发展，一方面 Clarity 自己内部需要一个信息系统来管理自己生产的传感器和传感器的版本、批次、拥有者等信息，另一方面 Clarity 的合作方往往是政府部门和传统硬件厂商如空调、汽车行业的企业，他们一般都不具备很强的软件开发能力，所以 Clarity 需要为他们定制合适的传感器管理系统来查看空气质量数据和管理传感设备。因此，本课题研究的“空气质量传感器管理系统”（以下简称本系统）应运而生。

1.2 课题主要研究内容

本系统分 3 个模块，版本管理模块、Smart Home 模块和 Smart City 模块。

1.2.1 版本管理模块

本模块的名字叫“Balanar”¹，用户是 Clarity 的硬件开发人员。原本用户使用 Excel 记录传感器版本、批次、拥有者等信息，但随着传感器的不断改进，合作方越来越多，制作给合作方的特殊版本越来越多，零件进货的批次各有不同，使用 Excel 的时间成本和精力成本越来越高。所以 Balanar 解决的问题和实现的功能如下：

1. 版本、批次和传感器之间的关系混乱，包括硬件版本、固件版本、软件版本、五种零件版本和五种零件的批次；Balanar 要理清关联关系，用最佳的数据结构体现；
2. 版本编号格式难以维护；Balanar 要能够自动校验格式并给出错误描述；
3. 硬件版本和固件版本兼容，固件版本又和软件版本兼容；Balanar 要能够记录和判断版本兼容性

¹此名字本身无实际含义，只作代号

1.2.2 Smart Home 模块

本模块的名字叫“Robotic”¹，目标客户是日本一家家电制造企业，最终用户则是使用家电和智能家居的个人。该企业生产的产品包括空调和空气净化器，因此 Clarity 为其量身打造了把两者相结合的 Smart Home。空气净化器只能一直开着或者在人的操控下间歇性地净化空气，而搭载空气质量传感器的空调也只能检测空气质量并报告给用户，并不能实际改善空气质量，于是 Robotic 引入了一个可编程的扫地机器人完美地弥补了两者的不足。Robotic 按以下流程工作：

1. 不同房间的空气质量传感器持续上传数据到服务器，服务器保存数据；
2. 服务器定时地计算并评估空气质量的好坏；
3. 一旦有房间空气质量低于一定水平，服务器就会给机器人下达净化指令，机器人会移动到对应的房间并开启空气净化器；
4. 空气质量好转之后，服务器再给机器人下达停止净化指令，机器人会关掉空气净化器原地待命；
5. 机器人有手动模式，用户可以开启手动模式并直接指挥其到某房间净化；

另外用户还可以查看个人携带的传感器空气质量、城市的空气质量以及每个房间的空气质量折线图。

1.2.3 Smart City 模块

本模块的名字叫“Azwraith”²，目标客户是美国一家计划做智慧城市的政府部门，最终用户是政府工作人员。该城市计划大量安装 Clarity 的传感器以监控城市空气质量，Azwraith 为之提供了配套功能如下：

1. 城市会安装和拆除传感器；Azwraith 需要能够添加和删除设备（传感器）；
2. 政府工作人员可以地图形式直观的查看传感器位置和当前空气质量；
3. 政府工作人员可以图表形式直观地查看并下载传感器最近历史数据，并且可以选择不同时间精度和空气质量度量；
4. 政府工作人员可以表单形式选择不同时间跨度和精度来下载设备的长期历史数据；
5. 政府工作人员可以控制设备在地图和图表上是否显示；

1.3 论文组织结构

本文一共分为八章，各章节介绍如下

第一章 绪论 简单阐述了本课题研究的背景、意义和主要内容以及本论文的组织结构。

第二章 WEB 设计开发理论、技术和工具 通过对比 WEB 设计开发的一些理论、技术和工具介绍了本系统主要使用的设计理论、技术架构和开发工具以及使用它们的原因。

第三章 可行性分析 分析了本系统的技术可行性、经济可行性和操作可行性。

第四章 需求分析 分析了本系统的功能目标、用户用例、性能要求和用户运行环境，并根据设计稿设计了快速原型和数据字典

¹此名字含义为机器人的，因为 robot 太短太普遍

²此名字本身无实际含义，只作代号

第五章 系统设计与开发 介绍了总体上的设计思路，然后按模块介绍了设计和开发过程，最后介绍了几个主要组件的详细设计

第六章 系统最终实现效果 按不同模块和组件展示实现效果

第七章 系统测试和部署 介绍了本系统的开发、测试和运行环境以及持续集成的配置

第八章 总结与展望 总结了研究内容是否完成，思考了研究过程中的不足，并介绍了下一步的工作计划。

第二章 WEB 设计开发理论、技术和工具

2.1 设计理论

2.1.1 敏捷开发

2.1.2 Web Components 组件标准

2.1.3 Material-design UI 设计

2.1.4 Flex 布局和 Responsive 设计

2.1.5 RESTful API 设计和 Web Socket 数据更新

2.1.6 Flux 架构模式

2.2 技术架构

2.2.1 前端：Angular VS React

2.2.2 后端：Express VS Koa

2.2.3 数据库：MongoDB 和 Mongoose

2.2.4 服务器：AWS 云服务

2.3 开发工具

2.3.1 IDE: Webstorm VS Atom

2.3.2 版本控制：Git 和 Git Flow

2.3.3 代码生成：Yeoman 和 Yeoman Generators

2.3.4 文档生成：JsDoc VS EsDoc

2.3.5 代码质量：Lint 工具和 Git hooks

2.3.5.1 ESLint

2.3.5.2 JSLint

2.3.5.3 JSCS

2.3.5.4 Stylelint

2.3.5.5 Pre-commit

2.3.6 编译工具：Grunt VS Gulp VS Npm

2.3.7 单元测试：Mocha 和 Karma

2.3.8 持续集成：Travis-CI VS Solano

第三章 可行性分析

3.1 技术可行性

3.2 经济可行性

3.3 操作可行性

第四章 需求分析

- 4.1 功能目标
- 4.2 用户用例
- 4.3 快速原型
- 4.4 数据字典
- 4.5 性能需求
- 4.6 用户运行环境

第五章 系统设计与开发

5.1 总体设计

5.2 版本管理模块

5.2.1 前端

5.2.1.1 客户端 ORM

5.2.1.2 代码生成器

5.2.1.3 响应式设计

5.2.2 后端和数据库

5.2.2.1 CRUD Controller

5.2.2.2 on-save 实时更新

5.3 Smart Home 和 Smart City 模块

5.3.1 前端

5.3.1.1 redux 管理数据流

5.3.1.2 Echarts 绘制图表

5.3.2 后端和数据库

5.3.2.1 socket-io 实时更新

5.4 主要组件详细设计

5.4.1 客户端 ORM 表单组件

5.4.2 EChart 组件

5.4.3 AqiChart 组件

5.4.4 AqiMap 组件

5.4.5 API 组件

第六章 系统最终实现效果

6.1 版本管理模块

6.1.1 前端

6.1.1.1 客户端 ORM

6.1.1.2 代码生成器

6.1.1.3 响应式设计

6.1.2 后端和数据库

6.1.2.1 on-save 实时更新

6.2 Smart Home 模块和 Smart City 模块

6.2.1 前端

6.2.1.1 redux 管理数据流

6.2.1.2 Echarts 绘制图表

6.2.2 后端和数据库

6.3 主要组件详细设计

6.3.1 客户端 ORM 表单组件

6.3.2 AqiChart 组件

6.3.3 AqiMap 组件

第七章 系统测试和部署

- 7.1 开发环境配置
- 7.2 测试环境配置和持续集成
- 7.3 运行环境配置

第八章 总结与展望

8.1 工作总结

8.2 工作展望

谢 辞

感谢悉心指导论文设计思路和撰写的吴刚老师！

感谢在工作中给出指导的施宇晨同学！

感谢在工作中通力合作的傅浩南同学！