智能花园

软件架构文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 06/9/2018 | 1.0 | 完成软件架构文档的初步编写 | 李晗东 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 4

3. 逻辑视图 5

3.1 概述 5

3.2 控制层 6

3.3 持久层 8

4. 进程视图 8

5. 部署视图 9

6. 实现视图 10

7. 数据视图（可选） 11

8. 核心算法设计（可选） 11

软件架构文档 （简化版）

# 简介

## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

## 参考资料

1.RUP的软件架构文档模板

2.《软件工程导论》

# 用例视图

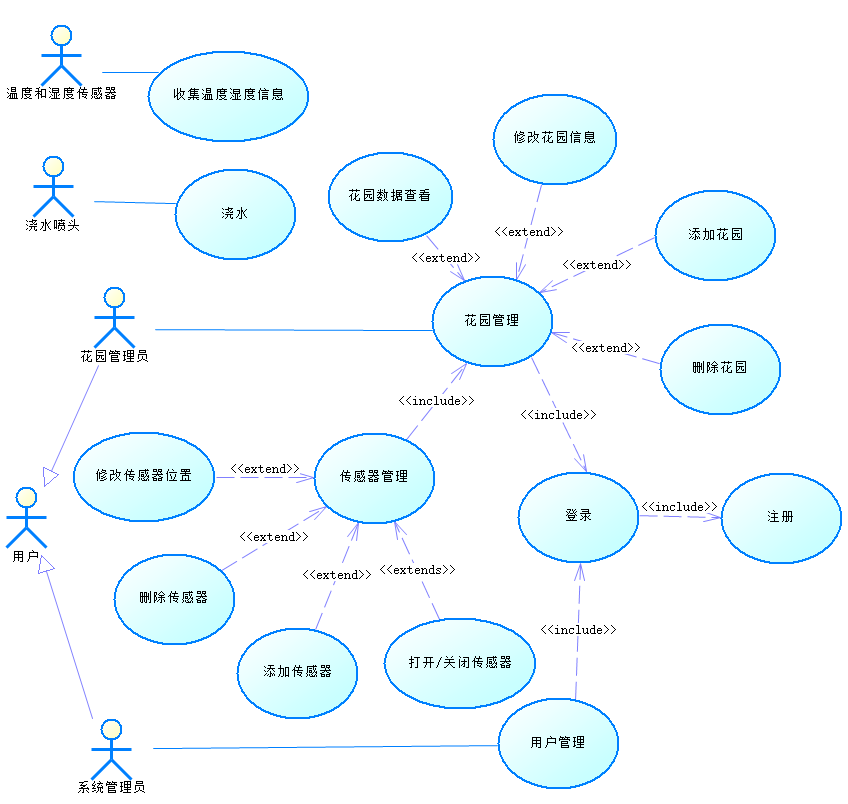


图1 智能花园用例图

# 逻辑视图

本章是对软件架构的逻辑视图的描述。主要内容包括描述重要的类，类的分包，子系统以及子系统 的分层等。另外还包括一些重要用例的实现。

## 概述

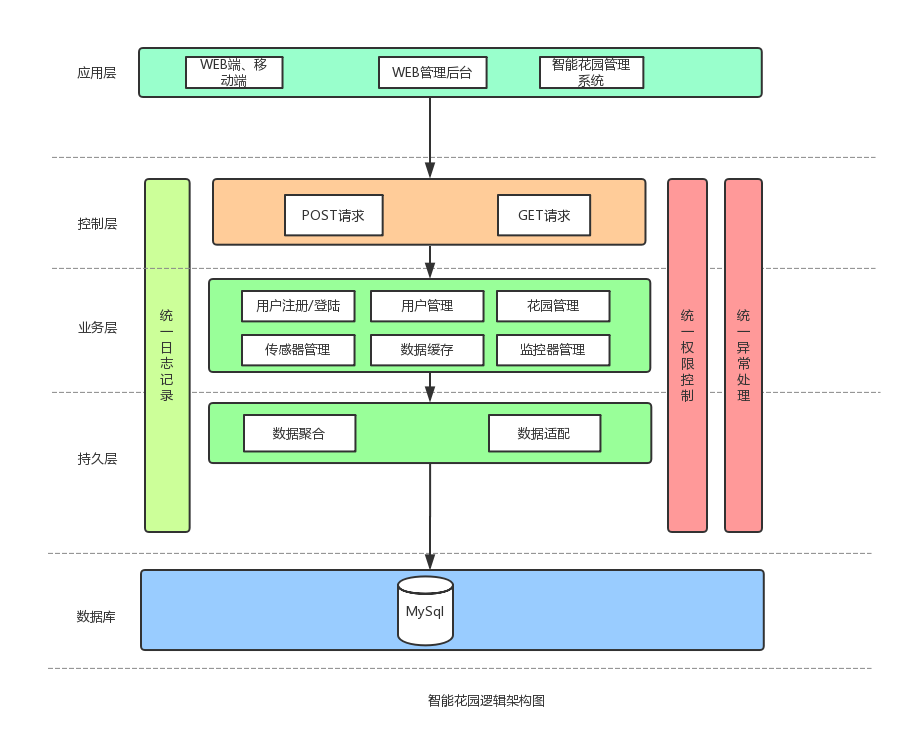


图2 智能花园系统总体架构图

智能花园系统的逻辑视图主要由五层组成，分别是应用层、控制层、业务层、持久层、数据库。

* + 应用层方面
    - 实现基于web端和移动端的应用，前端用React和React native框架，采用JavaScript语言；通过Spring boot搭建后台框架，采用Java语言，实现智能花园管理系统
  + 控制层
    - 接受来自用户的请求并通过相应接口对相应的请求进行接收、回应
  + 业务层
    - 主要包含用户注册/登陆、花园/传感器/监控器管理以及管理员对用户的管理
  + 持久层
    - 主要将收集的数据进行聚合、转化、抽取、适配，最后传入到数据库
  + 数据库
    - 主要收集用户所反馈的数据、传感器/监控器等所记录的数据

## 控制层

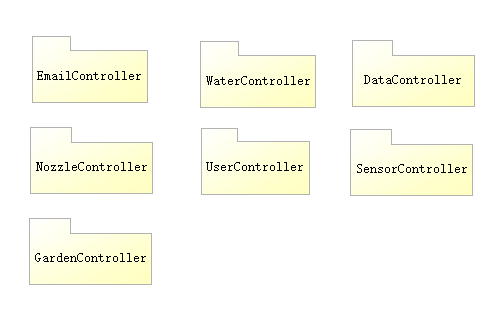


图3 控制层主要包图

控制层主要由7个子包组成，这7个子包如图所示。控制层主要实现对前端发送到后端的请求进行处理的过程。

EmailController处理验证码发送服务的请求。

WaterController处理对自动浇灌进行配置和执行的请求。

DataController处理与数据相关的请求，包括湿度数据，温度数据的获取和测试用接口虚拟数据生成的请求的处理。

NozzleController处理对喷头进行一系列管理操作的请求。

UserController处理用户登录和用户管理操作的请求。

GardenController处理对花园进行一系列管理操作的请求。

SensorController处理对传感器进行一些力管理操作的请求。

**3.3 业务层**

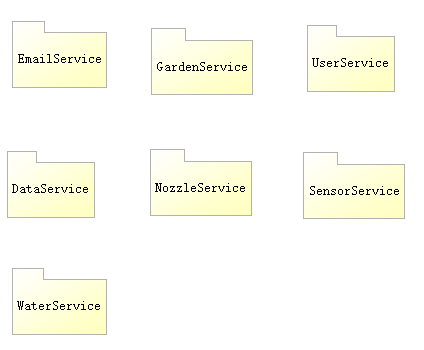


图4 业务层主要包图

业务层主要由7个子包组成，如图所示。七个子包提供了对对应的业务进行处理的功能。

EmailService提供了生成验证码并发送的功能。

GardenService提供了对于花园的CRUD操作的功能。

UserService提供了用户登录注册和后台管理的功能。

DataService提供了对温度湿度数据进行获取、添加和查询，以及虚拟数据生成的功能。包含了TempDataService，HumiDataServic和FakeDataService的三个类，分别对应温度数据的操作，湿度数据的操作和虚拟数据的操作。

NozzleService提供了对喷头进行CRUD操作的功能。

SensorService提供了对传感器进行CRUD操作的功能。

WaterService提供了对自动浇灌进行配置和自动浇灌的功能。

## 持久层

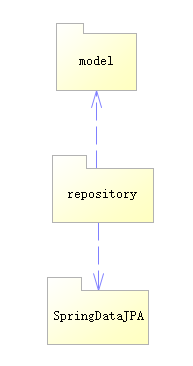


图5 持久层主要包图

持久层主要由3个子包组成，如图所示。其中model包规定了数据格式和相互之间的外键依赖，repository包则利用SpringDataJPA自定义了对数据库的一些操作。SpringDataJPA则是由Spring官方提供的数据库访问包，里面包含了对于数据库的基本操作的实现。

# 进程视图

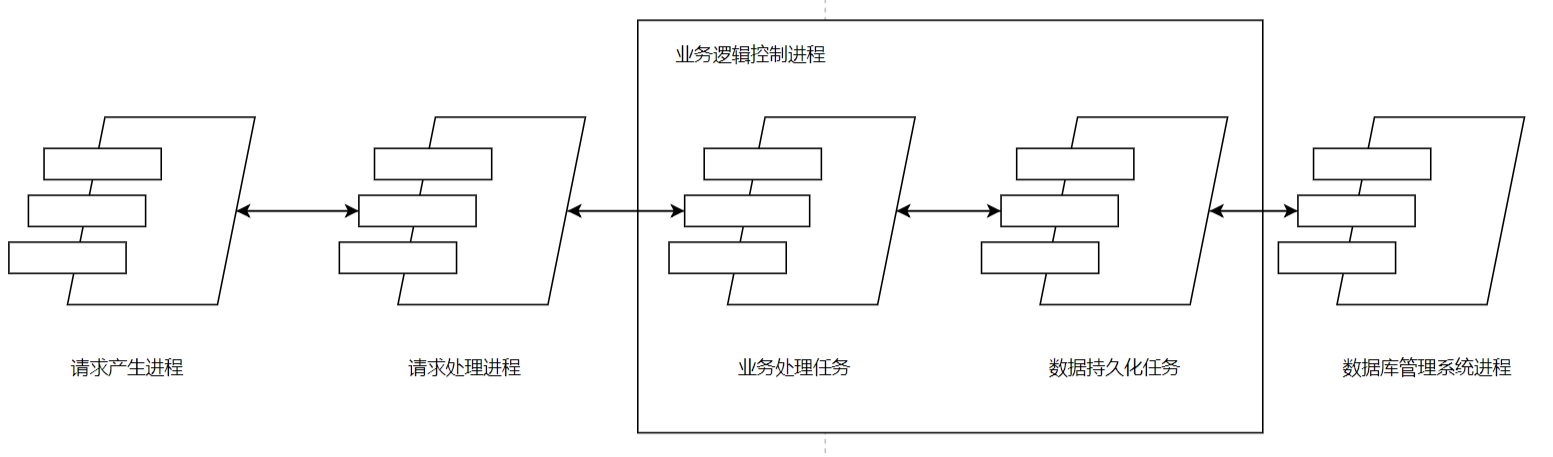


图6 智能花园进程视图

用户在网页或手机App上操作产生一系列请求，后端接受这些请求后创建新的进程对其进行处理，并根据对业务的处理结果给出对数据库的操作请求，该请求又将交由数据库服务器进行处理。

# 部署视图

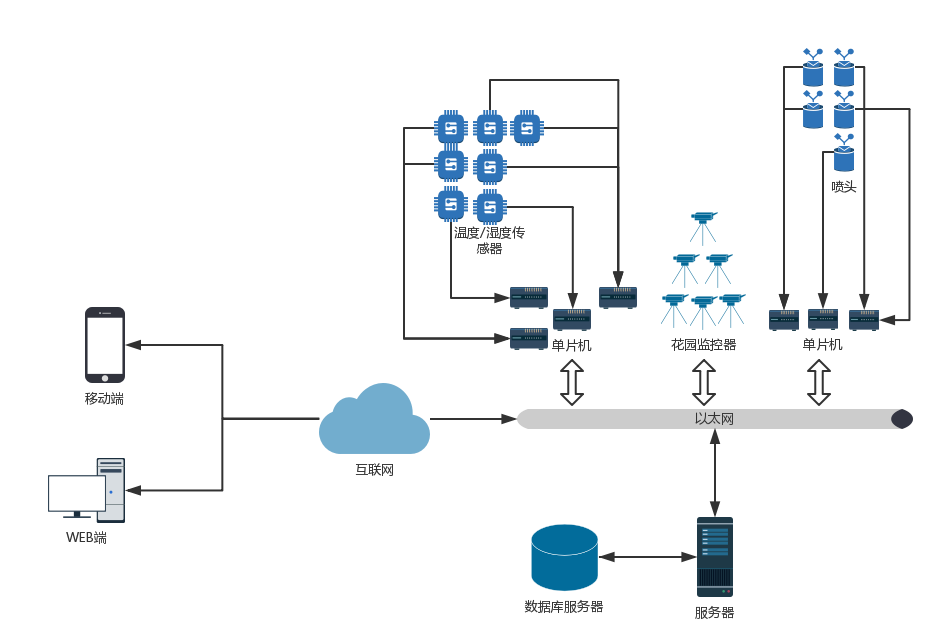


图7 智能花园部署视图

* 1. 客户端

用户主要通过浏览器或者移动App来访问系统，支持的浏览器包括IE 9+和Chrome 60+，

客户端和服务端的连接可以是局域网或互联网。

5.2 服务端

服务器运行智能花园后端系统，系统部署在Tomcat9.0容器上，通过jdbc与数据服务器连接。

单片机运行用于从传感器中实时读取温度和湿度的Python脚本，脚本通过Contab在单片机上定时运行，通过mysqldb连接数据库并更新数据。

5.3 数据库服务器

数据库服务器运行MySQL5.3数据库。

# 实现视图

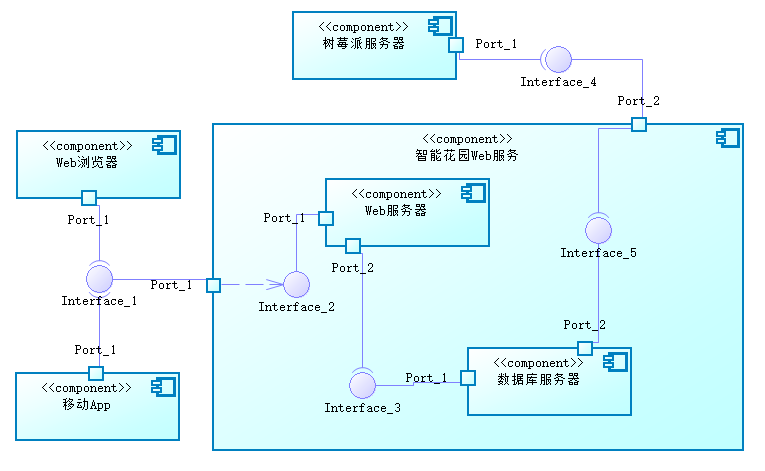


图8 智能花园实现视图

在实现过程中，我们将系统分为了客户端和服务端两大组件。客户端包括Web端与App端，这两个组件都通过请求与服务端联系。服务端包括了处理业务逻辑的服务器与处理数据的服务器。同时运行在树莓派上的传感器直接与数据库服务器相连。

# 数据视图

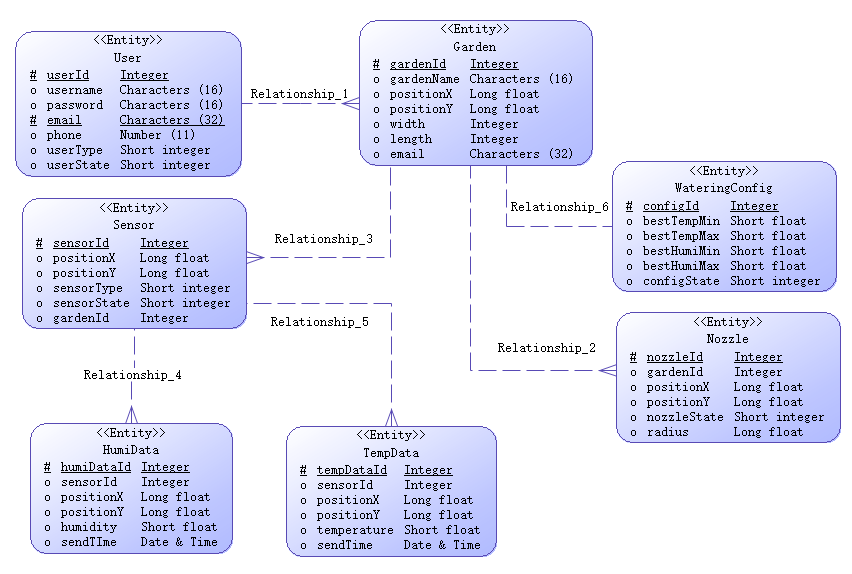


图9 智能花园数据视图

# 核心算法设计

浇灌算法设计概述：

后台系统定期检查配置了自动浇水数据的花园中每个传感器的实时数据，如果发现实时数据符合配置，则试图关闭离传感器最近的喷头，否则则开启离传感器最近的喷头。通过对喷头开关的实时调控来维持温度和湿度在一定范围内。