

**Watch Dog智能宿舍**

**Watch Dog**

**详细设计说明书**

**Version 1.0**

**2020.05.15**

**Written by 我无言以队**

****

**All Rights Reserved**

目录

[1 引言 3](#_Toc41778599)

[1.1 项目背景 3](#_Toc41778600)

[1.2 方案路线 3](#_Toc41778601)

[1.3 参考资料 4](#_Toc41778602)

[2 系统总体设计 4](#_Toc41778603)

[2.1 系统架构 4](#_Toc41778604)

[2.2 功能模块划分 5](#_Toc41778605)

[2.3 业务流程 5](#_Toc41778606)

[2.4 用户角色 6](#_Toc41778607)

[2.5 软件环境 6](#_Toc41778608)

[3 功能设计 7](#_Toc41778609)

[3.1 视频监控模块 7](#_Toc41778610)

[3.2 温湿度监控模块 9](#_Toc41778611)

[3.3 用户模块 10](#_Toc41778612)

[3.4 提醒模块 11](#_Toc41778613)

[3.5 扩展模块 12](#_Toc41778614)

[3.6 功能接口 13](#_Toc41778615)

[4 数据库设计 13](#_Toc41778616)

[4.1 数据表设计 13](#_Toc41778617)

[4.2 数据库UML图 14](#_Toc41778618)

1. 引言
   1. 项目背景

宿舍是大学生生活和学习的重要场所之一，为了帮助学生在本地和远程了解宿舍特定区域情况，起到威胁、监督、取证和管理大学生宿舍的作用，项目组开发了一套基于Web的智能宿舍监控系统，帮助学校更好的进行宿舍管理，保障同学们的生命财产安全。

* 1. 方案路线

Watch Dog智能宿舍系统由三大部分组成。首先是基础硬件设施，包括树莓派和传感器；其次是云端服务，用于托管传感器数据和运行数据库；最后是Web服务器，用来运行服务端、处理请求和数据。

* + 1. 基础硬件设施

使用树莓派4B+及其配套的视频传感器、温度传感器和湿度传感器等作为数据来源。

视频数据通过imageZMQ开源库进行处理和传输；离散的数字数据通过IoT平台订阅，服务器再通过AMQP SDK和Aliyuncore工具包获取消息队列中的数据。后期可以增加传感器，托管至IoT平台来实现更多功能。

* + 1. 云端服务

项目使用的云端服务包括阿里云ECS、阿里云IoT平台和阿里云RDS数据库。

其中ECS作为Web服务器和数据订阅服务器，用来运行Flask Web应用和获取平台数据存入数据库；云IoT平台则进行传感器数据的集中管理和订阅分发；云数据库则作为数据池，接收传感器数据并供Web服务端调用，同时也存储了用户数据等Web端运行时需要的数据内容。

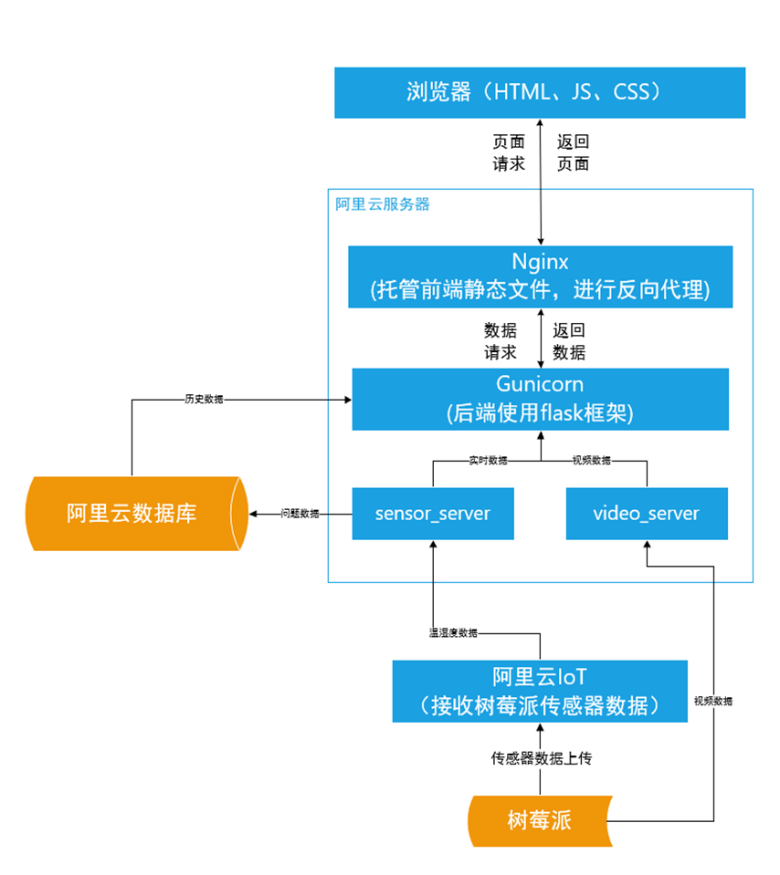
* + 1. Web服务器

后端使用Flask框架，应用服务以Restful形式暴露HTTP接口，使服务与服务间松耦合、可编排，并使用HTTP动词和URL提供统一的接口，便于组内前后端联合开发。服务内高聚合，专注单个服务的开发；服务间松耦合，通过依赖调用其他服务。服务模块化使系统具有良好的拓展性、重构性和易维护性。

* 1. 参考资料

1. 国家标准《软件需求说明书（GB8567——88）》
2. <http://docs.jinkan.org/docs/flask/>
3. <https://flask-cors.readthedocs.io/en/latest/>
4. <https://answers.opencv.org/question/56779/detect-open-door-with-traincascade/>
5. <https://www.cnblogs.com/vovlie/p/4182814.html>
6. 系统总体设计
   1. 系统架构

整体架构图如图表1所示：



图表 1整体架构图

如图所示，Web服务器从IoT平台获取温湿度数据进行处理并存入云端数据库，同时从终端直接获取视频流进行处理。在对这两种数据进行处理的过程中，若出现数据异常则分别向用户发送短信作为提醒。

* 1. 功能模块划分

系统划分为以下5个模块，如图2所示：

图表 2功能模块

功能设计模块将对这5个模块进行详细设计。

* 1. 业务流程

硬件设施：负责实时监控和温湿度检测；

WatchDog系统：负责在异常情况出现时给用户提供警示信息，并帮助用户随时了解宿舍的详细情况。

IoT平台订阅传感器数据，服务器上的脚本程序从IoT平台获取数据并与警戒值进行判断，该脚本文件包含了提醒模块，可以在异常出现时发送提醒，同时Web服务器从数据库调取数据，再在前端向用户展示，历史传感器数据存放在云数据内，也可按日期调用显示。

视频数据则由树莓派直接传送至Web服务器，再判断、存储、提醒之后，同样逐帧图片的形式向前端传送视频数据。历史视频数据存放在ECS服务器本地，也可按日期调用，在前端展示出动态视频。

具体如图3所示：



图表 3系统流程图

* 1. 用户角色

Watch Dog系统有两种系统角色：普通用户、系统管理员。

每个用户拥有唯一用户ID，在注册时填写且不可更改。电话号码作为提醒的渠道，在注册时需要按照格式填写。

在对用户所在宿舍完成硬件安装后，系统管理员在IoT平台上为该硬件分配唯一ID，普通用户在个人资料页面输入该设备ID进行设备的添加，之后便可获取到该设备（传感器）的数据。

* 1. 软件环境

本系统在云服务器上的CentOS系统中部署，使用RDS上的MySQL作为数据库，使用Nginx+Gunicorn的模式实现反向代理并作为项目的Web服务器。同时在IoT平台上托管传感器。





服务器开发和部署工作中的软件环境如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| ***项目过程*** | **操作系统** | **所用工具** | **外部依赖** |
| *开发过程* | Windows10 1909 | Flask1.1.1\Pycharm2019.3.4\VSCode1.45.0\WebStorm2019.3.4\Gitee | Node.js 12.8.1  Python 3.8.0  VueX 3.4.0  Vue.js 2.6.11 |
| *部署过程* | CentOS 8.0 | Nginx1.18.0\Gunicorn 20.0.4\Flask 1.1.1 | 阿里云RDS\阿里云IoT平台\AliyunCore SDK |

1. 功能设计
   1. 视频监控模块
      1. 功能描述

该模块向用户提供宿舍的实时视频，并且提供移动监测功能，当检测到人的运动时就会以逐帧图片的形式存储视频，供之后调用查证。结合提醒模块，可以在寝室有人闯入时向用户绑定的电话号码发送提醒短信。

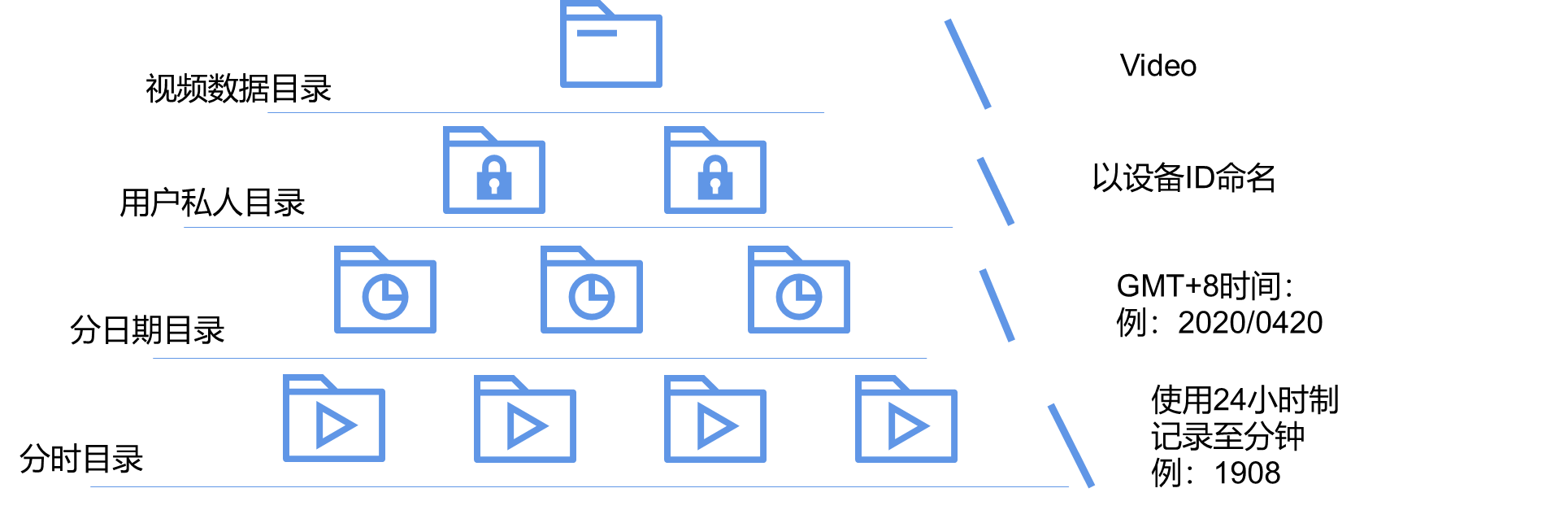
使用ImageZMQ库来实现树莓派和云服务器的视频数据传输。



在运动检测方面，使用FloodFill算法，并进行适当的参数调整，实现了对运动物体的监测和框选。

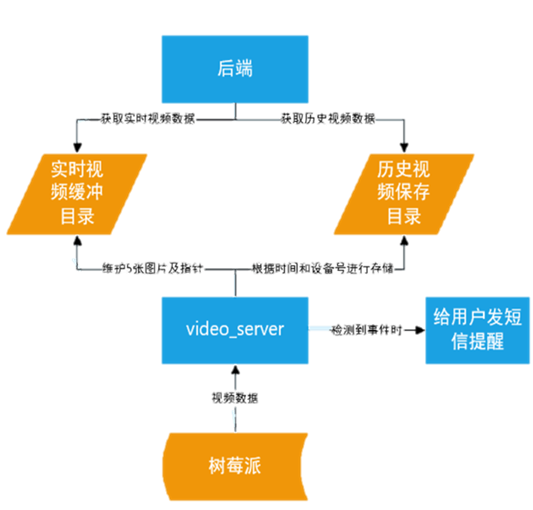
* + 1. 功能建模

历史视频存储规范如图7所示，使用树状文件结构存储各个用户的历史视频数据：



图表 4文件结构

视频模块流程图如图8所示：



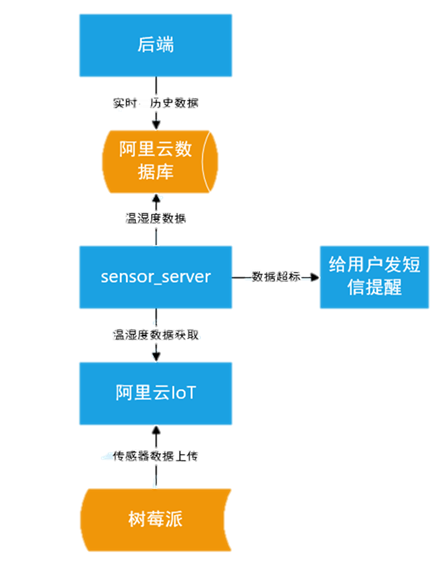
图表 5视频模块流程图

* 1. 温湿度监控模块
     1. 功能描述

用户可以通过该模块实时查看寝室温湿度情况，并且给出直观的图表展示以供监测和查询，当温度超过预警值时，该模块应能够结合提醒模块向用户发送提醒短信。

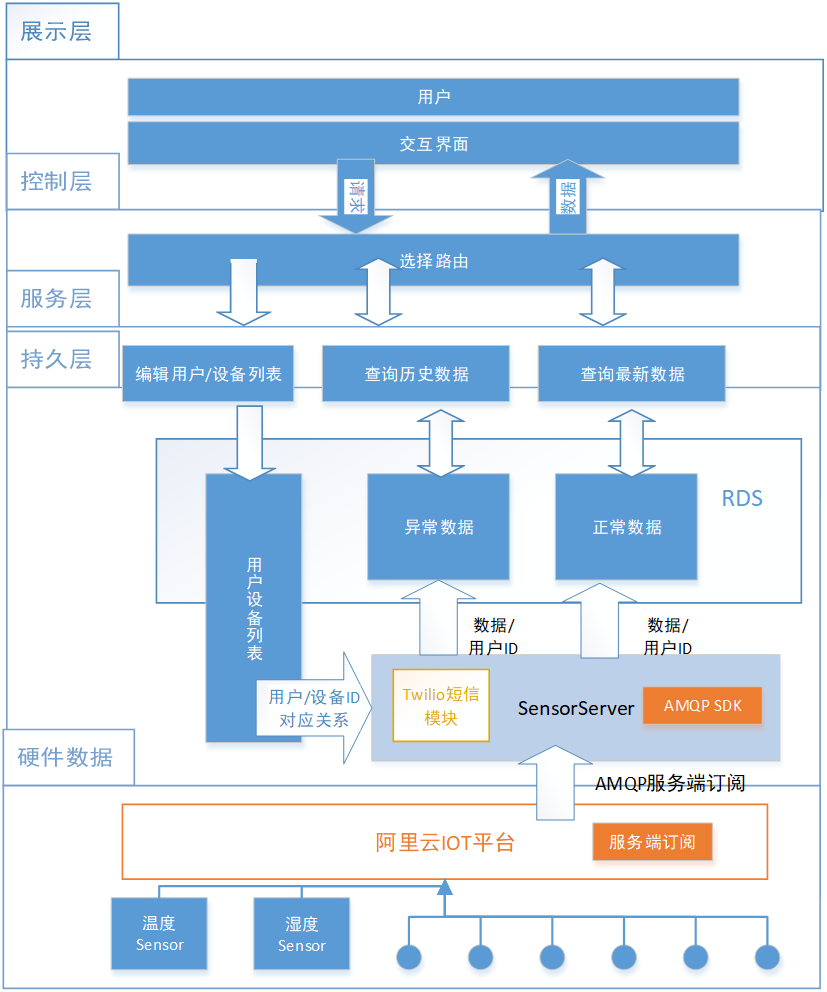
* + 1. 功能建模

该模块流程图如图9所示：



图表 6传感器模块流程图

硬件使用MQTT-TCP协议自定义topic接入IoT平台，便于后续获取数据并实现功能。Web服务端则使用AMQP SDK获取平台数据，便于硬件的统一管理和调配。模块架构图如图10所示：



图表 7传感器模块架构图

* 1. 用户模块
     1. 功能描述

通过该模块，用户可修改或查看个人资料（地址、电话号码等），登录和注册，向系统添加已有的传感器ID来接收传感器数据，该系统同其他模块有一定耦合度，考虑优先实现。

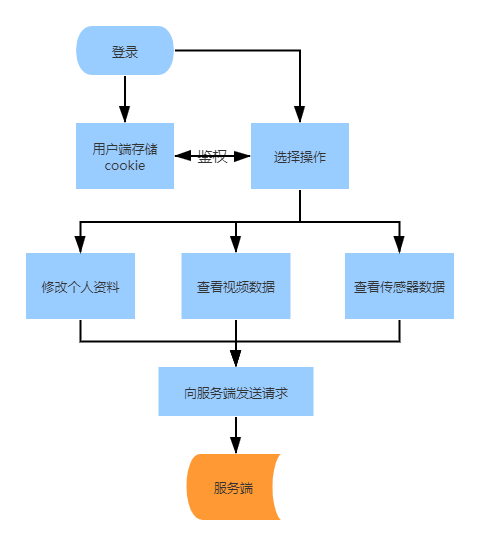
在用户注册方面，使用werkzeug.security库中的加密功能，采用了SHA256算法存储密码，避免明文存储，保证用户信息的安全。



在用户登陆成功后，用户浏览器端建立cookie来存储用户ID，在调用其他模块的功能时，提供用户信息，方便其他模块调用；

* + 1. 功能建模

登陆后使用cookie完成后续用户鉴权，如图8所示：

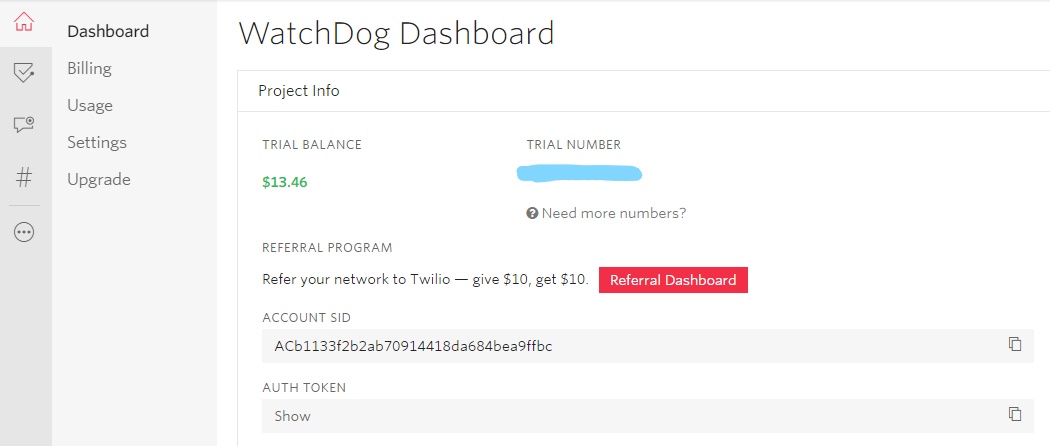


图表 8用户模块流程图

* 1. 提醒模块

该模块通过向各个模块提供接口实现其功能，可以向指定电话号码发送指定内容的短信，如闯入警告和温度警告等，提醒用户注意寝室安全。在后续扩展功能中也可作为日程提醒的媒介使用。

在实现过程中使用了Twilio API，按照文档导入SDK按照要求进行请求即可发送提醒信息。



为了配合该模块的使用，获取到异常数据后，都需要查询设备列表获得用户ID，进而获得用户手机号。

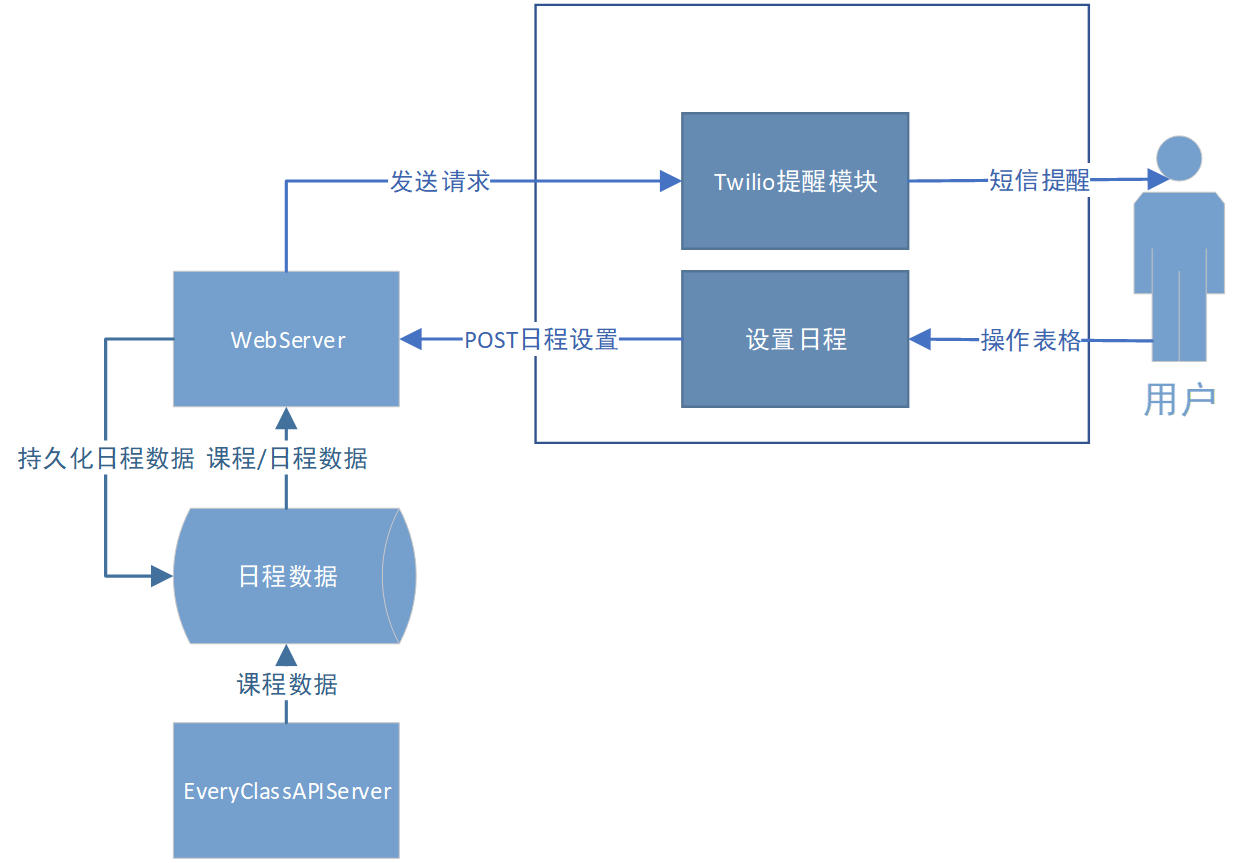
为了避免短时间的重复高频发送，在程序实现时，设定了1分钟的间隔时间。

* 1. 扩展模块
     1. 功能描述

该模块包含了后续可以进行扩展的功能，经组内评估，可以与现有模块较好对接以实现其功能。

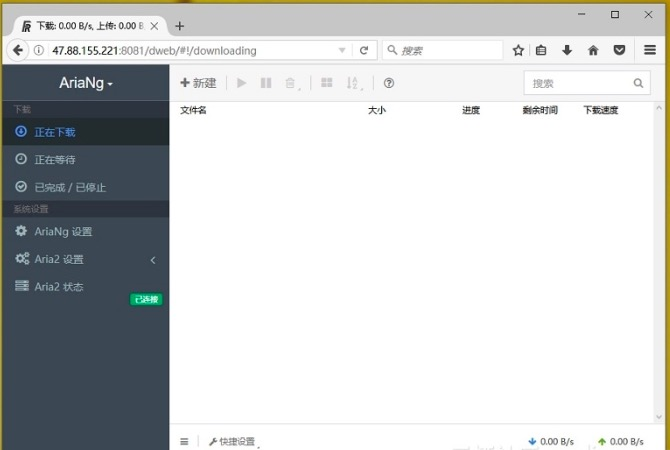
* + 1. 功能建模

部署每课API server V0.4.1获取课表日程进行提醒，也可自行设置提醒时间，如图9所示：



图表 9日程提醒

离线下载功能作为附加功能且有较为完备的部署工具，使用Docker部署较为简单，可以考虑独立于当前Web端提供服务，图10为部署之后的效果图：



图表 10 AriaNG效果图

* 1. 功能接口

后端提供接口及其功能如下表：

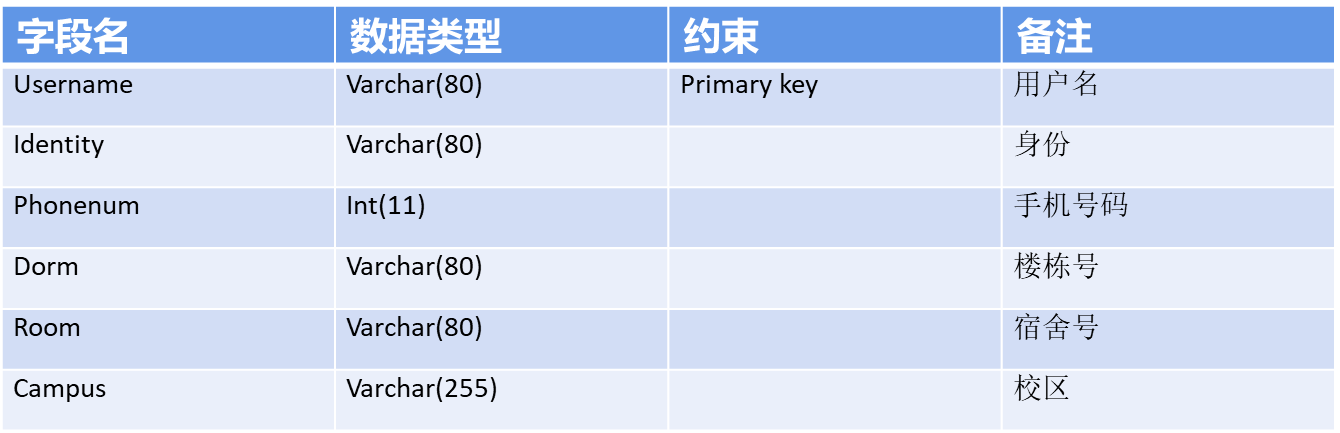
|  |  |
| --- | --- |
| **接口** | **功能** |
| **api/video** | 提供实时视频数据 |
| **api/video\_save\_data** | 存储视频文件 |
| **api/video\_record** | 视频记录 |
| **api/login** | 用户登录 |
| **api/register** | 用户注册 |
| **api/profile** | 查看、修改个人数据 |
| **api/sensor** | 提供传感器数据 |
| **api/reset\_pw** | 用户重置密码 |

1. 数据库设计

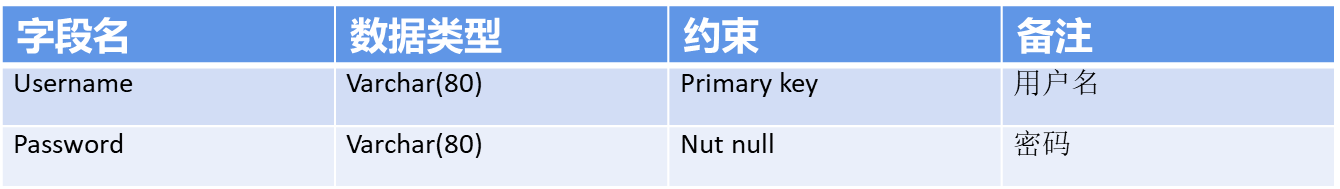
项目使用了阿里云RDS数据库，便于远程协作开发，使用MySQL作为数据库管理系统，一同作为项目的LNMP(Linux/Nginx/MySQL/Python)解决方案包。

* 1. 数据表设计

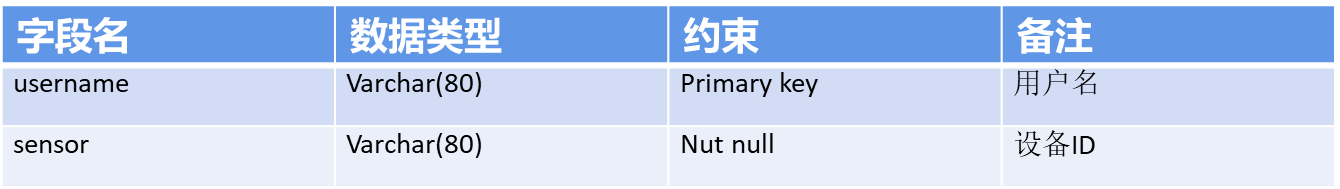
user表：



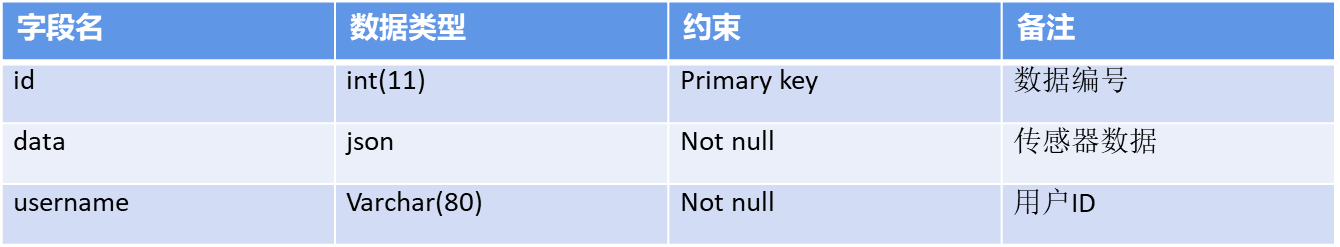
signon表：



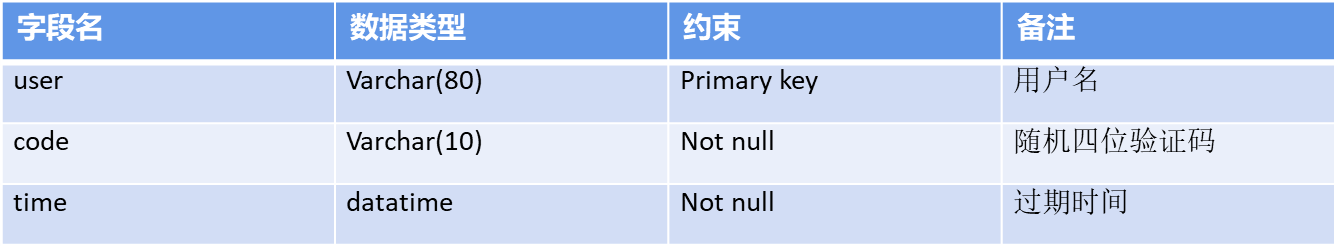
sensorlist表：



sensors表（与异常数据warning表结构相同）：



vercode表



* 1. 数据库UML图

