

成绩: \_\_\_\_\_



江 苏 大 学

JIANGSU UNIVERSITY

2018-2019 学年第 2 学期

《信息系统集成》大作业

学院名称: 计算机科学与通信工程学院

班级: 网络 1601

学生姓名: 李济上 3160610010

王勇翔 3160610011

张志远 3160610012

完成时间: 2019 5 27

# 一、实验目标

- 了解 Onenet 物联网信息集成平台、熟悉其中的数据集成方法（读写访问方法）；
- 掌握 Excel 等多数据源的数据提取方法；
- 熟悉天气网等外部系统的数据集成（提取）方法；
- 掌握基于 SQL 的数据库层面数据接口封装方法，包括函数、存储过程等使用方法；
- 掌握基于面向对象的可配置型数据访问接口的设计方法；
- 掌握典型的远程服务接口定义、实现方法；
- 熟悉基于远程服务接口的 APP、动态网站设计方法；

# 二、实验内容

按图 1 所示的体系结构和功能描述实现整个系统。

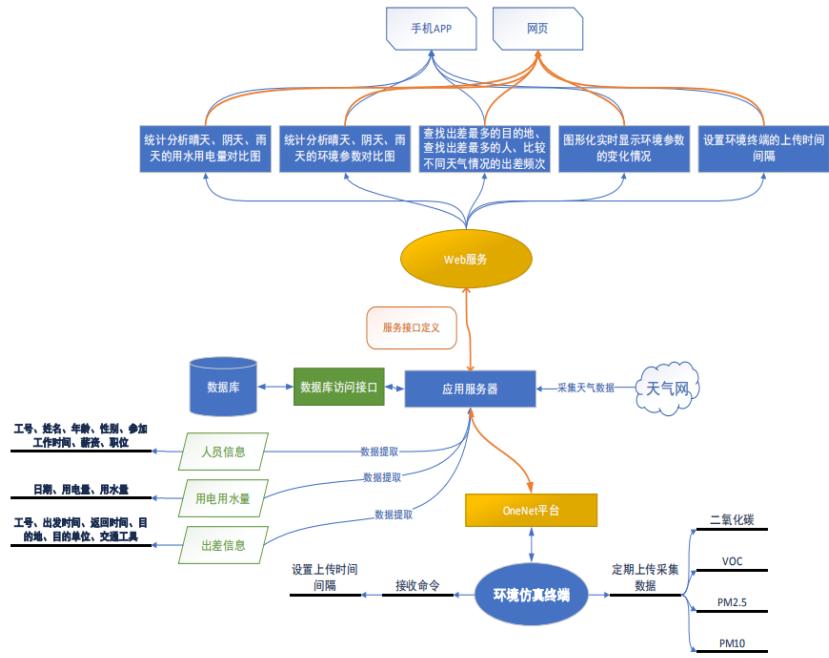


图 1 系统体系结构

其中具体要求如下：

- (1) 实现环境仿真终端的设计与实现，用随机数模拟二氧化碳、VOC、PM2.5、PM10 四种环境参数，并定期上传到 onenet 平台对应的设备中。
- (2) 从给定的 excel 表中提取人员信息、用水用电量信息、出差信息等；
- (3) 从天气网中提取近一年的天气数据（与出差信息的时间区间对应起来）；
- (4) 设计数据模型、利用 SQL 完成数据库层面数据接口封装，要求对外提供存储过程、视图、或函数的调用形态，屏蔽数据表；
- (5) 基于 Java 编写数据访问接口，要求可以通过简单配置便可支持 MySQL、SqlServer 等数据库；
- (6) 根据应用需求定义远程服务接口；
- (7) 基于所定义的远程服务接口完成 APP 和网站的设计开发。

### 三、实验设计

#### 1. 环境仿真端设计：

仿真端主要实现的功能是随机生成环境参数，并上传到 onenet 平台上，这里与 onenet 交互主要使用的是 `urllib.request` 这个模块来实现的，随机生成环境参数则是用 `random` 模块来实现，多线程则是通过 `threading` 模块实现，中间数据要转换为 json 格式则是用 `json` 模块来实现的

例：

```
#上传函数
def http_put_data():
    url = "http://api.heclouds.com/devices/" + deviceld + '/datapoints'
    d = time.strftime('%Y-%m-%dT%H:%M:%S')
    CO2 = random.randint(0,999)
    VOC = random.randint(0,999)
    PM25 = random.randint(0,999)
    PM10 = random.randint(0,999)
    values = {"datastreams": [ {"id": "CO2", "datapoints": [{"value": CO2}]}, {"id": "PM25", "datapoints": [{"value": PM25}]},
               {"id": "PM10", "datapoints": [{"value": PM10}]}, {"id": "VOC", "datapoints": [{"value": VOC}]} ]}

    jdata = json.dumps(values).encode("utf-8")
    request = urllib.request.Request(url, jdata)
    request.add_header('api-key', APIKey)
    request.get_method = lambda: 'POST'
    request = urllib.request.urlopen(request)
    print(values)
    timer = threading.Timer(2, http_put_data)
    timer.start()
    return request.read()
```

#### 2. Sql 语句部分的数据库设计：

这里选择使用 php 的 `mysqli` 相关的方法来连接数据库，从数据库取得相关数据后，用

来画图表(折线图等),

例:

```
//创建连接
$conn = new mysqli($servername,$username,$password,$dbname);

//检测连接
if($conn->connect_error){
    die("连接失败: ".$conn->connect_error);
}

//echo "连接成功\n";

// 设置编码,防止中文乱码
mysqli_query($conn, "set names utf8");

// 晴、阴、雨对应的 sql 语句
$sql_qing = "SELECT 用电用水量.用水量,用电用水量.用电量,用电用水量.日期\n .
    "FROM 天气,用电用水量\n .
    "WHERE (天气.日期=用电用水量.日期 AND 天气.天气='晴'));

$result_qing = mysqli_query($conn, $sql_qing);
$flag = 0;

if(mysqli_num_rows($result_qing) > 0)
{
    while($row = mysqli_fetch_assoc($result_qing))
    {
        //echo "日期: " . $row["日期"] . " 用水量: " . $row["用水量"] . " 用电量:
        " . $row["用电量"] . "<br>";
        if($flag<100)
        {
            $flag = $flag + 1;
            array_push($qing_T_date, date('m-d',strtotime($row["日期"])));
            array_push($qing_E_date, $row["用电量"]);
            array_push($qing_W_date, $row["用水量"]);
        }
    }
}
```

```

        }
    }

else
{
    echo "0 结果";
}

```

### 3. 应用服务器的设计:

应用服务器这里选择的是用 Tomcat 来实现的， tomcat 作为一个 servlet 容器，前端向 tomcat 发送一个请求(可以为 post 或者 get 形式),tomcat 收到请求后会生成一个 httprequest 和一个 httpresponse 对象，并将其发送给 servlet 实例，servlet 重写 doget () 和 dopost () 方法来处理 httprequest，并将结果写入 httpresponse 返回；

### 4. 网页前端的设计:

这里网页的设计，因为之前比较呆，就没有想到可以找个框架来写，因此.....就自己用 html+css+js+php 来自己写了一个；

(1) 写了一个登陆界面，用户输入使用 html 的表单属性来写的，而验证码则是想到登陆界面一般都会有，所以用 canvas 写了一个，使用的 js，另外这个用户信息是以明文的方式直接写在了 html 中：

- 验证码.js:

```

/**生成一个随机数**/

function randomNum(min,max){
    return Math.floor( Math.random()*(max-min)+min);
}

/**生成一个随机色**/

function randomColor(min,max){
    var r = randomNum(min,max);
    var g = randomNum(min,max);
    var b = randomNum(min,max);
    return "rgb("+r+","+g+","+b+")";
}

var code = drawPic();
//alert(code);

function clear_it()
{
    document.getElementById("changelmg").onclick = function(e){
        e.preventDefault();
    }
}

```

```
code = drawPic();
}

}

clear_it()

/**绘制验证码图片**/

function drawPic(){
    var canvas=document.getElementById("canvas");
    var width=canvas.width;
    var height=canvas.height;
    var ctx = canvas.getContext('2d');
    ctx.textBaseline = 'bottom';

    /**绘制背景色**/
    ctx.fillStyle = randomColor(180,240); //颜色若太深可能导致看不清
    ctx.fillRect(0,0,width,height);

    /**绘制文字**/
    var str = 'ABCCEFGHJKLMNPQRSTWXY123456789';
    var last_list = new Array(4);
    for(var i=0; i<4; i++){
        var txt = str[randomNum(0,str.length)];
        last_list.push(txt);
        ctx.fillStyle = randomColor(50,160); //随机生成字体颜色
        ctx.font = randomNum(15,40)+'px SimHei'; //随机生成字体大小
        var x = 10+i*25;
        var y = randomNum(25,45);
        var deg = randomNum(-45, 45);
        //修改坐标原点和旋转角度
        ctx.translate(x,y);
        ctx.rotate(deg*Math.PI/180);
        ctx.fillText(txt, 0,0);
        //恢复坐标原点和旋转角度
        ctx.rotate(-deg*Math.PI/180);
        ctx.translate(-x,-y);
    }
}
```

```

/**绘制干扰线*/
for(var i=0; i<8; i++){
    ctx.strokeStyle = randomColor(40,180);
    ctx.beginPath();
    ctx.moveTo( randomNum(0,width), randomNum(0,height) );
    ctx.lineTo( randomNum(0,width), randomNum(0,height) );
    ctx.stroke();
}

/**绘制干扰点*/
for(var i=0; i<100; i++){
    ctx.fillStyle = randomColor(0,255);
    ctx.beginPath();
    ctx.arc(randomNum(0,width),randomNum(0,height), 1, 0, 2*Math.PI);
    ctx.fill();
}

return last_list.join("");
}

```

(2) 信息展示界面，则是主要用的是超链接的方式，来展示这个用水用电量啊，天气情况啊啥的图表，这里画图使用 `php` 写的，通过接收到数据库中的所需要的数据后，来画图

例：

```

<?php
// ****画 用电图*****
require_once ("E:\wamp\jpgraph-4.2.6\jpgraph\jpgraph.php");
require_once ("E:\wamp\jpgraph-4.2.6\jpgraph\jpgraph_line.php");

$graph = new Graph(1000,600);
$graph->SetScale("textlin");
$graph->SetShadow();

$graph->img->SetMargin(60,30,30,70); //设置图像边距

$graph->graph_theme = null; //设置主题为 null, 否则 value->Show(); 无效

```

```

//创建设置 y 轴 晴天 的曲线
$lineplot2=new LinePlot($qing_E_date);
$lineplot2->value->SetColor("red");
$lineplot2->value->Show();

//创建设置 y 轴 阴天 的曲线
$lineplot3=new LinePlot($yin_E_date);
$lineplot3->value->SetColor("green");
$lineplot3->value->Show();

//创建设置 y 轴 雨天 的曲线
$lineplot4=new LinePlot($yu_E_date);
$lineplot4->value->SetColor("black");
$lineplot4->value->Show();

//将曲线放置到图像上
$graph->Add($lineplot2);
$graph->Add($lineplot3);
$graph->Add($lineplot4);

//$/graph->xaxis->SetTickLabels($qing_T_date);

$graph->title->Set("晴天_阴天_雨天_用电图(红-晴天;绿-阴天;黑-雨天)"); //设置
图像标题

$graph->xaxis->title->Set("数据个数"); //设置坐标轴名称
$graph->yaxis->title->Set("度"); //设置坐标轴名称
$graph->title->SetMargin(10);
$graph->xaxis->title->SetMargin(10);
$graph->yaxis->title->SetMargin(10);

$graph->title->SetFont(FF_SIMSUN,FS_BOLD); //设置字体

```

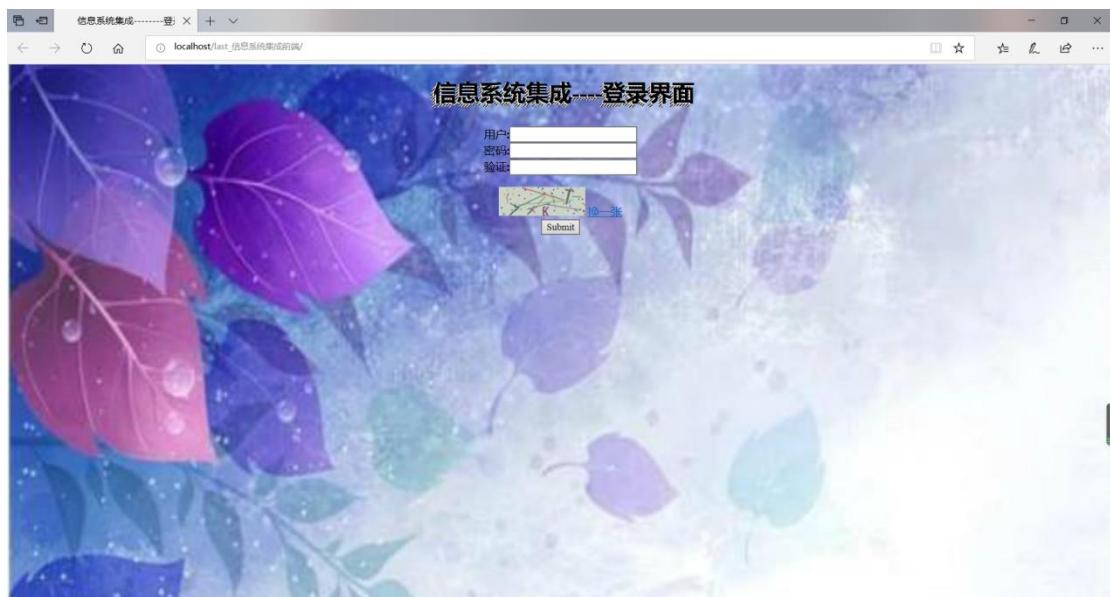
```
$graph->yaxis->title->SetFont(FF_SIMSUN,FS_BOLD);  
$graph->xaxis->title->SetFont(FF_SIMSUN,FS_BOLD);
```

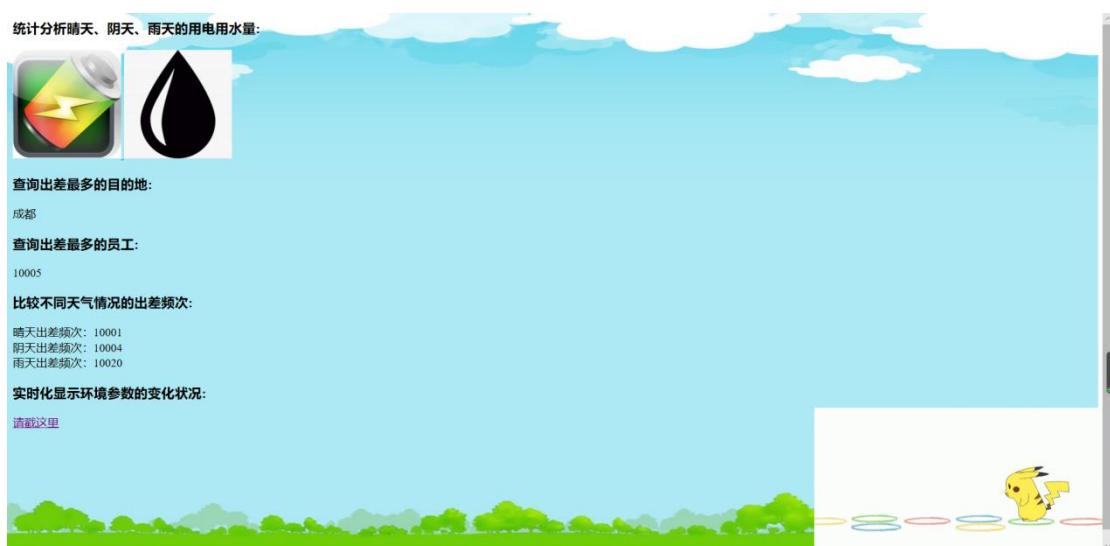
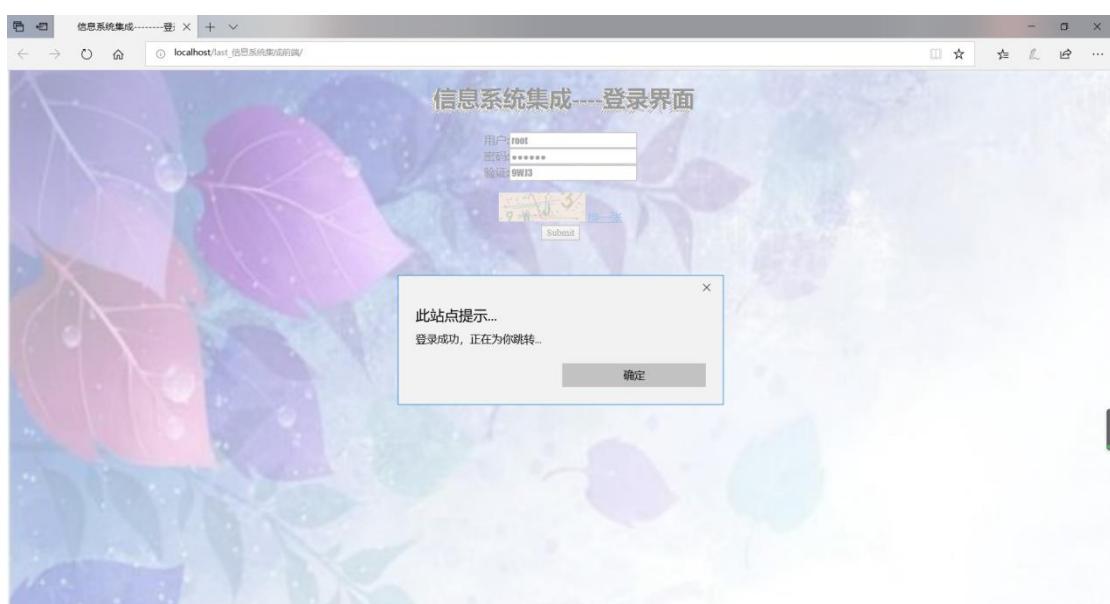
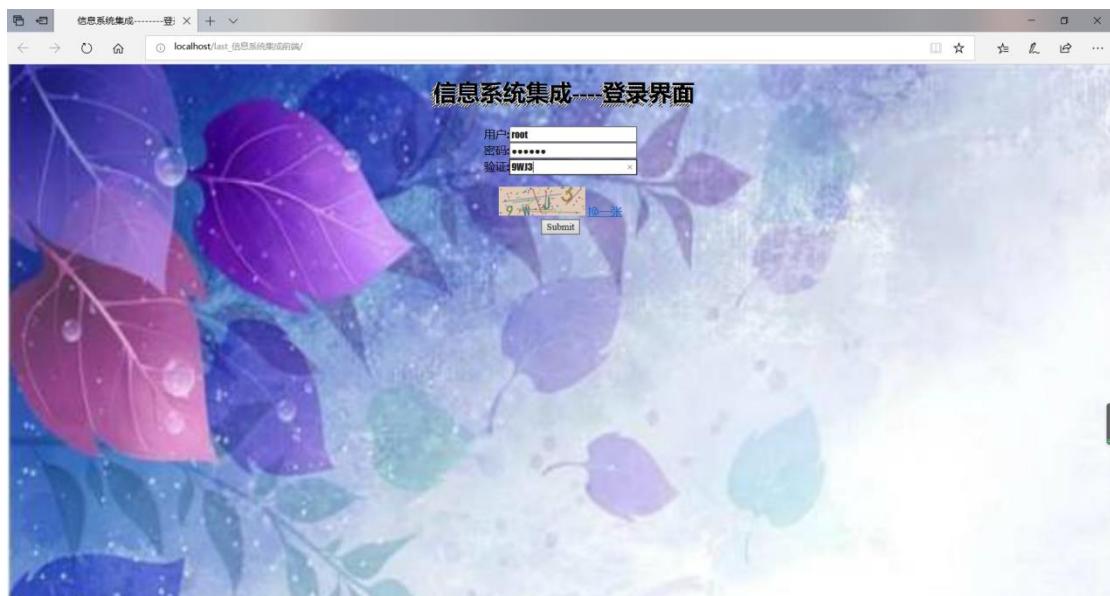
```
$file_1 = "..\\data\\晴天_阴天_雨天_用电图.png";  
if(file_exists($file_1))  
{  
    unlink($file_1);  
}
```

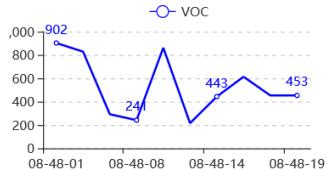
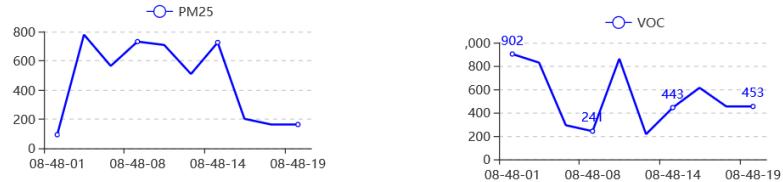
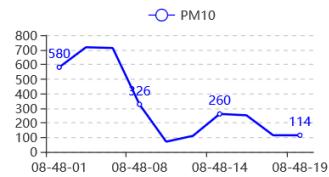
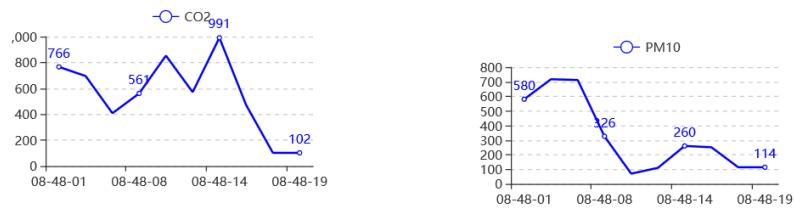
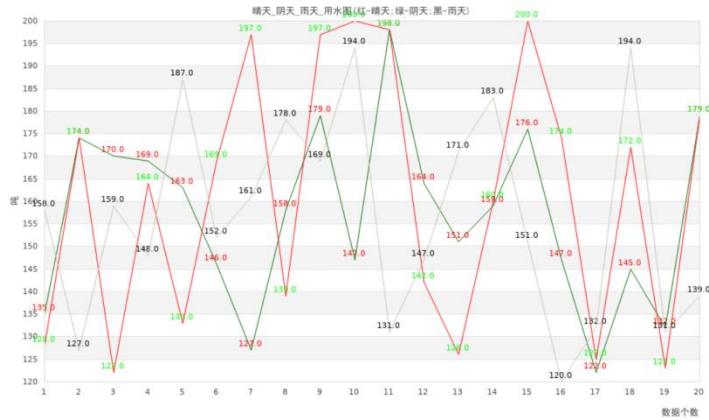
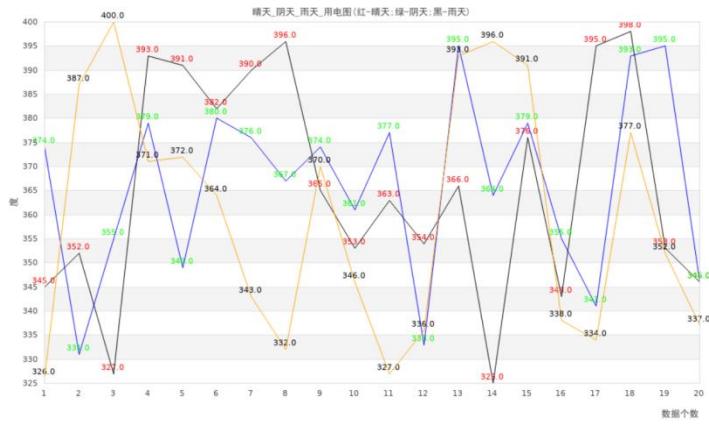
```
$graph->Stroke("..\\data\\晴天_阴天_雨天_用电图.png"); //输出图像
```

(3) 实时显示方面用的则是 onenet 提供的接口，直接调用的

## 四、实验结果







## 五、实验总结

(1) 王勇翔：通过这次大作业，对前端的编写更加熟练，对 html+css+js+php 的组合拳的应用更加熟悉，对数据库与前端的交互有了更加深刻的认知

(2) 李济上：对应用服务器部分的理解更加深刻，对整个实验有更清晰的认识，对服务器的搭建更加了解，对 tomcat 使用有了基本的熟悉

(3) 张志远：对 onenet 平台的使用基本熟悉，对使用脚本与 onenet 进行交互的原理和程序有了更加深刻的认知，对数据库交互方面和网络爬虫上的知识也有了初步了解