Universe Scientific Publishing
Computer System Network and Telecommunications,计算机系统网络和电信(1) 2019,3
ISSN:2661-3719(Print) ISBN:2661-3727(Online)

学生上网系统的设计与实现

阳子聪[1] 郭晓丹[2]
四川大学锦城学院 计算机与软件学院 四川 成都 611731

DOI:10.18686/jsjxt.v1i3.1280

【摘要】对于数据量日益剧增,数据采集和数据分析都面临巨大的工作量并且效率低下的情况。学生上网系统基于 MVC 架构,使用 JAVA 语言和 MySQL 数据库优化设计,良好的实现数据实时分析。

【关键词】大数据 框架 学生上网系统 传输 方法

1 引言

学生上网系统是用于帮助用户分析学生上网情况,在这个大数据时代,数据采集和分析都要面临巨大的工作量,并且在数据采集完成后会存在一些错误的数据以及格式,便需要进行数据处理达到数据分析的标准。而一个高效的系统,用户可根据自身需求选择分析条件,系统进而选择对应的算法,再经过一系列流程,将具体的分析结果和报告呈现给用户,实现数据的高效分析。

2 学生上网系统分析

2.1 系统需求分析

本平台需求说明书主要面向的读者对象为学校客户和一般互联网用户,说明书的目的是使学校管理和一般互联网用户能够清晰明了的使用本软件,并且能够使其更方便快速的进行有效操作,以达到符合他们标准的需求。我们也希望学校管理和一般互联网用户在进行软件体验的同时也能从软件和这份需求说明书中找出漏洞并进行反馈,让我们更好地升级改进软件,以此优化提高用户体验度。

2.2 系统目标

此系统平台用于在学学生上网信息统计,可模拟自动生成学生上网记录信息,并且根据上网记录信息分析得到:

1. 访问最多的网站
2. 男生访问最多的网站
3. 女生访问最多的网站
4. 指定时间区间内上网次数最多的学生姓名
5. 指定时间区间内上网次数最多的男生学生姓名
6. 指定时间区间内上网次数最多的女生学生姓名
7. 以 10 分钟为区间、上网次数最多的时间区间
8. 指定时间段,查看上网次数最多的学生学号
9. 同时【5s 以内】同时上同一个网站的男生列表
10. 同时【5s 以内】同时上同一个网站的女生列表

2.3 系统结构

系统采用 MVC 框架模式设计,分别搭建各类模块实现其功能。例如将学生类划分为控制器,数据持久层,业务逻辑层,减少代码的重复性,提高开发效率。具体功能结构如下:

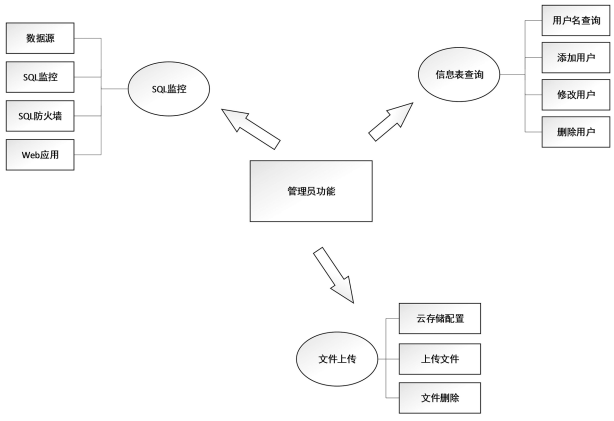

图 1 管理员功能结构图

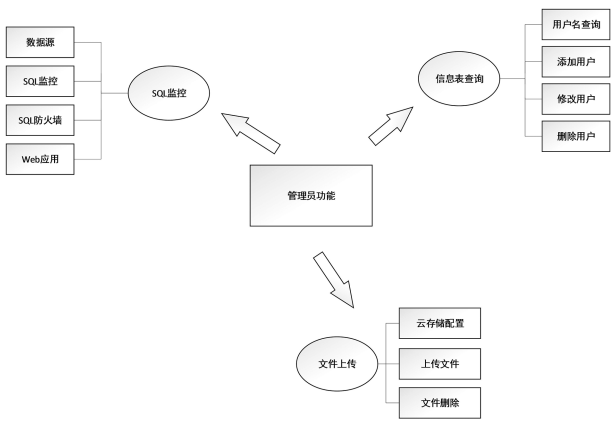

图 2 用户功能结构图

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

图 3 登陆界面

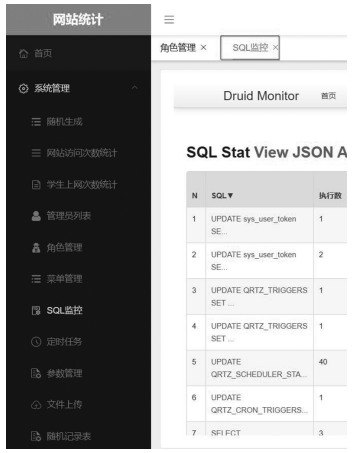

图 4 管理员使用界面

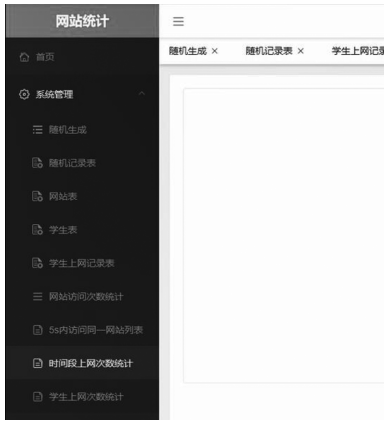

图 5 用户使用界面

2.4 数据库设计

本系统使用的数据库是 MySQL,因为其操作简单,安全性高且可移植性强,在对数据进行获取时,能够较快捷地检索。Spring 连接 MySQL 的方式有很多,例如 JDBC,Spring JPA,Hibeirnate,Mybatis 等,我采用最简单的 JDBC 方式连接 MySQL,主要通过注入 JDBCTemplate 来操作。

2.4.1 数据传输

首先需要生成相关的学生信息及上网信息,再通过 web 响应,将生成数据存入数据库 MySQL,此时经过数据分析处理后返回数据库,并在新的表中存储。当用户使用 web 页面时,可通过 web 请求向数据库端获取相关数据,并通过页面进行可视化展示。

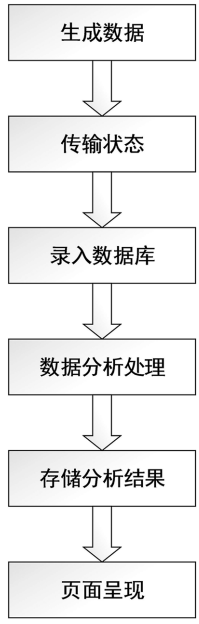

图 6 数据传输关系图

2.4.2 数据库结构设计

根据各个功能模块创建数据库 Internet,下面介绍主要的 4 张数据表:

学生信息表存储上网学生学号、学生姓名、学生性别、学生访问网站的时间、学生访问访问网站的次数,结构如下:

序号	字段	说明	数据类型	允许为空	主键
1	sid	学生学号	int(20)	NOT	P
2	sname	学生姓名	Varcha(255)	NOT	
3	sex	学生性别	char(10)	NOT	
4	stime	访问网站的时间	Timestamp	NOT	
5	snumber	访问网站的次数	int(20)	NOT	

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

网站信息表存储学生所有访问的网页的序号、网页的姓名、网页地址信息、访问该网页的次数、男生访问次数、女生访问次数,结构如下:

序号	字段	说明	数据类型	允许为空	主键
1	wid	网页的序号	int(20)	NOT	P
2	wname	网页的姓名	Varchar(255)	NOT	
3	waddress	网页地址信息	Varchar(255)	NOT	
4	wnumber	访问该网页的次数	int(20)	NOT	
5	w_m_number	男生访问次数	int(20)		
5	w_w_number	女生访问次数	int(20)		
6	w_log	记录用户 id 和时间	Varcha(255)	NOT	

间隔为十分钟的时间段信息表存储学生上网的时间段序号、时间段、时间段内学生学号、时间段内访问次数、时间段内男生访问次数、时间段内女生访问次数。

序号	字段	说明	数据类型	允许为空	主键
1	tid	时间段序号	int(20)	NOT	P
2	t_tq	时间段	Varcha(255)	NOT	
3	t_sid	时间段内学生学号	int(20)	NOT	
4	t_wid	时间段内网站序号	int(20)	NOT	
5	t_wnumber	时间段内访问次数	int(20)	NOT	
6	t_wmn	时间段内男生访问次数	int(20)	NOT	
7	t_wwn	时间段内女生访问次数	int(20)	NOT	

查询结果存储信息表存储用户自定义查询数据的查询结果的序号、访问次数最多的网站序号、男生访问次数最多的网站、女生访问次数最多的网站、上网次数最多的时间区间。

序号	字段	说明	数据类型	允许为空	主键
1	did	查询结果的序号	int(20)	NOT	P
2	d_w	访问次数最多的网站序号	int(20)	NOT	
3	d_m_w	男生访问次数最多的网站	Varchar(255)	NOT	
4	d_w_w	女生访问次数最多的网站	Varchar(255)	NOT	
5	d_t_wnumber	上网次数最多的时间区间	Varchar(255)	NOT	

3 系统功能实现

3.1 系统设计

系统采用三层架构设计,分别是模型层、视图层、控制层。具体基类如下:[1]

(1)路由基类:将用户请求进行处理,并送入相应应用的控制器中。

(2)系统启动类:在 web 服务器接收到请求后,自动载入相应的应用和基础类。

(3)学生基类:主要为接收参数对学生表数据进行刷新操作,并增添对学生表数据库的增删改查操作。

(4)学生记录基类:实现对学生记录表的刷新,并根据前台传入参数调用业务层方法实现对数据的分析处理并返回给前台。

(5)随机记录基类:在前台操作后,根据传入参数生成数据并保存在数据库中。

(6)网站基类:与学生基类相似,对网站表进行刷新并返回给前台和在数据库进行保存。

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

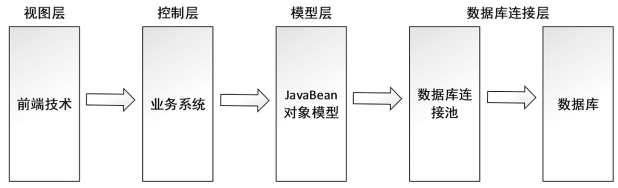

图 7 系统流程图

浏览器会根据用户操作发送请求由系统接收请求并验证请求是否合法。然后由路由器解析请求并分解为对应的模块和参数,将参数传递给对应的模块进行处理,调用改模块的控制器进行实现方法,最终返回结果给前台页面展示。[2]

3.2 功能实现

系统为前后端分离项目,前端使用 node.js,后台设计使用 java 语言和采用 springboot+mybaits(持久层框架),在本文只介绍下系统中主要的代码实现部分。

为达到前后台交互功能,通过 api 引入 axios 并封装在底层工具类文件,设置 ajax 请求方法,并将格式转换为 json 格式实现异步通信。[3][4][5] 代码如下:

```
getDataList () {
  this.dataListLoading = true
  this.$http({
      url: this.$http.adornUrl('/internet/drandomrecord/list'),
      method: 'get',
      params: this.$http.adornParams({
        'page': this.pageIndex,
        'limit': this.pageSize,
        'key': this.dataForm.key
      })
  }).then(({data}) => {
    if (data && data.code === 0) {
      this.dataList = data.page.list
      this.totalPage = data.page.totalCount
    } else {
      this.dataList = []
      this.totalPage = 0
    }
    this.dataListLoading = false
  })},
  // 获取数据列表
  siteList () {
    this.$http({
      url: this.$http.adornUrl('/internet/dsite/listAll'),
      method: 'get',
      }).then(({data}) => {
    if (data && data.code === 0) {
      this.dsiteList = data.data
      } else {
    }
    })},
```

这里展示的是两个 http 请求,采用的是异步通信方式,在执行发送请求代码后紧接着使用回调函数,所以只有在每个请求完成后才会分别执行回调函数。而如果前端页面使用的为阻塞方式即向服务器发送一个请求后,只有当服务器完成该请求后才会执行下一个请求,那么前台页面往往会在提交一个请求后就处于长时间的等待状态。[6]

数据是由用户在前段页面提交随机生成数据请求来生成,然后根据参数设置条件,对每个数据类型随机生成。用户可以在网站中选择姓名如“张三”,亦或是点击随机生成姓名,选择性别为“男”或“女”,浏览网站的时间区间如“2 点至 3 点”,时间间隔如“5 秒”,再通过键盘输入数字来限制生成数据的数量。这些数据不会在当前页面显示出来,而是在用户看不见的后台数据库中生成。会有一个“历史数据模块”负责记录我们生成数据的痕迹,例如在生成数据模块我们选择了一个名字为“张三”,性别为“男”,时间区间为“3 点至 4 点”,数据数量限制为 500 条,那么在“历史数据模块”就会有这么一条操作记录。部分代码如下:

```
Assert.isNullArray(dRandomRecord.getStudentIdList(),"请选择学生生成方式");
Assert.isNullArray(dRandomRecord.getSiteNameList(),"请选择网站生成方式");
Assert.isNull(dRandomRecord.getRandomDate(),"请输入生成日期");
Assert.isBlank(dRandomRecord.getStartTime(),"请输入开始时间段");
Assert.isBlank(dRandomRecord.getEndTime(),"请输入结束时间段");
Assert.isNull(dRandomRecord.getNumber(),"请输入生成条数");
Long[] studentIdList = dRandomRecord.getStudentIdList();
String[] siteNameList = dRandomRecord.
```

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

```
getSiteNameList();
    //dRandomRecord. setSiteName ( ArrayUtil.
contains ( siteNameList," 随 机") ? " 随 机" :
Arrays. stream ( siteNameList ). collect
(Collectors. joining(",")));
    List<DStudentEntity> list = dStudentSer-
vice. list(new QueryWrapper<DStudentEntity>
(). lambda()
    . in ( DStudentEntity:: getStudentId, studen-
tIdList)
    );
    String format = DateUtil. format(dRandom-
Record. getRandomDate(), "yyyy-MM-dd");
    String startTimeStr = format + " " + dRan-
domRecord. getStartTime();
    String endTimeStr = format + " " + dRan-
domRecord. getEndTime();
    /* *
     * 判断是否选择随机
     */
    if ( ArrayUtil. contains ( siteNameList," 随
机")) {
    dRandomRecord. setSiteName("随机");
    List<DSiteEntity> list1 = dSiteService. list
(null);
    List<String> collect = list1. stream(). map
(DSiteEntity:: getSiteName). collect (Collectors.
toList());
    siteNameList = ArrayUtil. toArray (collect,
String. class);
    }else {
    dRandomRecord. setSiteName(Arrays. stream
( siteNameList ). collect ( Collectors. joining
(",")));}
```

根据需求可查询同时上网(5s 内)且为同一个网站的男女生列表,先写一个循环遍历学生上网记录表的上网地址,并获取网站地址转换为字符串形式。三个条件:如果遍历到倒数第二条信息时则退出本层循环;获取的两个网址不一样时则退出本层循环;获取的学生姓名相同时则退出本层循环。再进行获取上网的时间,如果时间间隔小于 5 秒的话则添加信息。最后对同时上同一个网站的人进行计数统计并返回给前台显示。代码如下:

```
    @RequestMapping("/studentAccessList")
    @ RequiresPermissions ( " internet: dstuden-
trecordssite:list")
    public R studentAccessList (int page, int
limit,String key,String gender){
        List<Map> maps = dStudentRecords-
SiteService. studentAccessList(MapUtil. builder(" 
gender", gender). build());
        List<Map> mapList = new ArrayList<
>();
        for (int i = 0; i < maps. size(); i++) {
          if (i == maps. size() - 1) {
            continue;
          }
          Map mapr = new HashMap();
          Map map = maps. get(i);
          Map map1 = maps. get(i + 1);
           String site_name = map. get(" site_
name"). toString();
          String site_name1 = map1. get("site_
name"). toString();
          if (! site_name. equals(site_name1)) {
            continue;
          }
          String name = map. get("name"). toS-
tring();
           String name1 = map1. get ( " 
name"). toString();
          if (name. equals(name1)) {
            continue;
          }
          String online_time = map. get("online_
time"). toString();
           String online_time1 = map1. get("on-
line_time"). toString();
           long time = DateUtil. parseDateTime
(online_time). getTime();
          long time1 = DateUtil. parseDateTime
(online_time1). getTime();
          if (time1 - time <= 5000) {
            mapr. put("key1",map);
            mapr. put("key2",map1);
            mapList. add(mapr);
          }
        }
```

ISSN: 2661-3719(Print) ISBN:2661-3727(Online)

```
List<Map> collect = mapList.stream().skip(page - 1).limit(limit).collect(Collectors.toList());
PageUtils pageUtils = new PageUtils(collect,mapList.size(),limit,page);
return R.ok().put("page",pageUtils);}}
```

4 总结

学生上网系统是为对学生的上网各项情况作出分析所设计的系统且符合项目需求,能够分析大规模数据。具有页面简洁,功能方便,数据传输快捷,分析结果详细,开发流程清晰的优点。用户能够根据自己需求对数据作出分析得出结论,极大减少数据采集和数据分析的过程时间。而项目中也存在不足,在对数据进行处理计算时,使用的是 mybatis 提供的对数据库的查找方法以及在后台中直接编写 java 程序对数据进行计算。因而限制了数据量,当数据量过大时易发生程序崩溃和运行较慢。所以项目应在算法处作出改进,与 spark,hadoop 等大数据技术相结合。

【参考文献】

[1]qq_33257238. Vue+SpringBoot 前后端分离小实验[EB/OL]. (2018-9-25)[2019-8-4]. https://blog.csdn.net/qq_33257238/article/details/82833537

[2] zhoukaishun. 搭建 spring-boot+vue 前后端分离框架并实现登录功能[EB/OL]. (2018-8-12)[2019-8-4]. https://blog.csdn.net/zks_4826/article/details/81603865

[3] 行径行. SpringMVC 映射的前端后台数据交互总结[EB/OL]. (2017-11-19)[2019-8-4]. https://www.jianshu.com/p/fc8793f430f4

[4] mdahly. ajax + springboot + Restful 实现前后端分离项目[EB/OL]. (2018-9-10)[2019-8-4]. https://blog.csdn.net/bat_xu/article/details/82597149

[5] white_fisher. Spring Data REST 入门(三):自定义配置[EB/OL]. (2017-1-6)[2019-8-4]. https://blog.csdn.net/soul_code/article/details/54139709

[6] _明月. IntelliJ Idea SpringBoot 数据库增删改查实例[EB/OL]. (2017-8-27)[2019-8-4]. https://blog.csdn.net/dear_Alice_moon/article/details/77621452