

工作报告【10.4】

实现内容：

- 1、基本实现了数据统计和历史记录模块
- 2、进入用户管理页面时可定时（**每五分钟**）读取新的行为数据文件并读入数据库

实现形式：

- 1、数据统计模块：



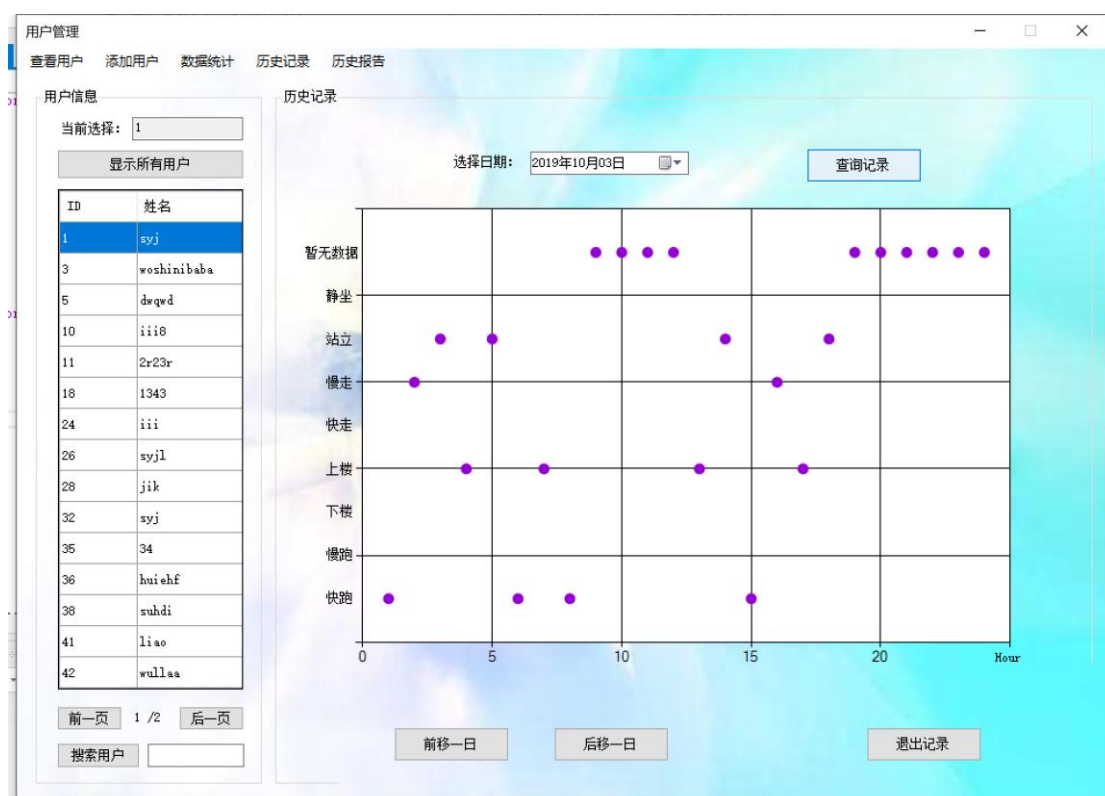
在“数据统计”模块，左侧选择用户，右侧选定开始和结束的时间后点击“时段统计”按钮，可显示这段时间内该用户的各种行为的**总时长和比例**。

注：

a.为了方便选定时间，设置了“快捷选项”，可以选定“日”、“周”、“月”单选按钮来实现对时间长度的精准控制，同时，选定“自定义”的话可以自行选定开始和结束的精确时间。

b.柱状图可以根据时间的具体长度，调整单位为秒、分或小时。

2、 历史记录模块：



历史记录模块按照 ppt 的形式用点状图记录下用户在选定日期每个小时进行的最多的行为，例如最左边第一个点代表 2019 年 10 月 3 日用户在 0-1 小时内进行的最多的行为是快跑。

注：

a.一个点代表的是附近一个小时用户最多的行为。

b.如果某段时间没有行为数据记录，对应的点将位于“暂无数据”一

行。

c.由于 c#图表控件难以支持一张表同时包含点状图和柱状图，因此 ppt 中该模块显示行为总量的柱状图放在“数据统计”模块的柱状图中实现。

个人的一些修改和看法：

1、 csv 文件内容：



```
getdata.py - C:\Users\Song\Desktop\学习和工作\人体行为2\bin\Debug\csvData\getdata.py (3.7.4)
File Edit Format Run Options Window Help
import pandas as pd
import numpy as np
import random

def judgeDay(y,M):
    if M == 1 or M == 3 or M == 5 or M == 7 or M == 8 or M == 10 or M == 12:
        return 31
    elif M == 2:
        if y % 400 == 0 or (y % 4 == 0 and y % 100 != 0):
            return 29
        else:
            return 28
    else:
        return 30

dic = ["sitting", "standing", "slowwalk", "fastwalk", "downstair", "upstair", "slowrun", "fastrun"]

#设置信息:
ID = 1
length = 10000
timestep = 2
y = 2019
M = 10
d = 3
h = 12
m = 0
s = 0

#生成csv代码:
data = []
str1 = str(y)+"_"+str(M)+"_"+str(d)+"_"+str(h)+"_"+str(m)+"_"+str(s)
for i in range(length):
    row = [str(y)+"_"+str(M)+"_"+str(d)+"_"+str(h)+"_"+str(m)+"_"+str(s), y, M, d, h, m, s, dic[random.randint(0, len(dic)-1)]]
    data.append(row)
    s += timestep
    while s > 59:
        s -= 60
        m += 1
    while m > 59:
        m -= 60
        h += 1
    while h > 23:
        h -= 24
        d += 1
    while d > judgeDay(y, M):
        d = 1
        M += 1
    while M > 12:
        M = 1
        y += 1
df = pd.DataFrame(data)
print(df)
df.to_csv(str(ID)+"_"+str1+".csv", index = None, header = ["Time", "Yea", "Mon", "Day", "Hour", "Minute", "Second", "Do"])
```

便于数据的统计，csv 文件加至八列，分别保存行为开始时间（数据库主键）、行为时间的年、月、日、时、分、秒、具体行为（有些是数据库关键字所以英文有些不全）。

如果 csv 只含有一个“yyyy_MM_dd_hh_mm_ss”格式的东西代表时间的话，其实时间范围的判断和统计会很麻烦，所以如果数据录入端录入数据的时候能生成上面八列形式的 csv 会方便很多。

	字段名称	数据类型
🔍	Time	短文本
	Yea	数字
	Mon	数字
	Day	数字
	Hour	数字
	Minute	数字
	Second	数字
	Do	短文本

同时，为了保存每一个用户的行为信息，为每一个用户都新建了一个数据库，表的设计如上图。

2、 行为数据录入问题：

行为数据时间采集间隔较短也产生了一个数据量较大的麻烦，间隔 2s 使得一个用户一天行为数据超过 4 万条，录入数据库时间花费将近 30 秒。如果数据过多的话录入一次一旦超过 5 分钟，录入过程中 timer 时间事件再次触发的话不能确定会不会出事…（没尝试过）

3、 实现的测试数据：

基于应用的测试，部分用户已拥有部分行为数据(通过上述 python 代码生成并已保存至相应数据库)

具体如下表所示：

ID	行为数据时间范围
1	10 月 1、2 号全天; 3、4 号部分时间
3、5	10 月 1 号部分时间
其他	均不含任何数据

因此，在数据统计时，对应 ID 对应时间可统计出正确信息，不含数据的 ID 和时间统计时会显示“暂无数据”的提示。

另外，行为数据录入时若与数据库内原行为数据时间发生冲突（主要是主键的唯一性质），将会录入失败，所以有必要的话可清空上述测试数据。

4、 数据文件的位置：

打开压缩包，打开人体行为 2\bin\Debug 文件夹之后：

bugCsv	2019/10/4 15:28	文件夹	
csvData	2019/10/4 15:28	文件夹	
mdbData	2019/10/4 15:32	文件夹	
photos	2019/9/29 19:06	文件夹	
人体行为	2019/10/3 22:16	Microsoft Access D...	408 KB
人体行为2	2019/10/4 15:31	应用程序	321 KB
人体行为2.exe.config	2019/9/5 21:02	XML Configuration ...	1 KB
人体行为2.pdb	2019/10/4 15:31	Program Debug Da...	124 KB

其中 bugCsv 文件夹是当上文讲到的行为数据文件录入失败时，会将该行为数据文件移到该文件夹，给使用者自行处理。csvData 文件夹保存定时录入的 csv 文件，成功录入后自动删除，以及生成随机行为数据的 python 代码。mdbData 文件夹保存每个用户的行为数据库，命名为 ID+“_data.mdb”，以及一个 basisData.mdb 文件，用于在添加用户时新建行为数据库。