

# 朝陽科技大學

## 資訊與通訊系

### 【實務專題成果報告】

#### 心率監測管理系统

指導教授：彭俊澄 卢俊文  
組 員：黃鑫宏 (10430370)  
施养波 (10430378)  
王储君 (10430380)  
许浩望 (10430385)

中 華 民 國 1 0 5 年 1 1 月

# 目录

第一章 摘要.....	5
1.1 选题背景.....	5
1.2 项目背景.....	6
1.3 目的 .....	7
第二章 可行性研究的前提.....	8
2.1 心脏疾病.....	8
2.2 Microsoft Bank2 的介绍 .....	9
第三章 市场分析与风险评估.....	16
3.1 宏观环境分析.....	16
3.2 竞争对手分析.....	16
3.3 市场局限.....	17
3.4 SWOT 分析.....	17
第四章 技术分析.....	18
4.1 硬件技术分析.....	18
4.2 软件技术分析.....	18
4.3 软件要求.....	18
4.4 生理状况预测（六条预测算法） .....	19
4.5 性能 .....	20
4.6 软件目标.....	20
4.7 条件和限制.....	20
第五章 详细设计及功能实现.....	21
5.1 设计分工与时间甘特图.....	21
5.2 设计流程图.....	22
5.3 数据库设计.....	24
第六章 实际功能界面测试.....	32
6.1 登录界面.....	32
6.2 注册界面.....	32
6.3 会员账户信息界面.....	33
6.4 修改资料界面.....	33
6.5 修改密码界面.....	34
6.6 紧急联络人界面.....	35
6.7 生理状况界面.....	35
6.8 心率阈值界面.....	36
6.9 实时心率界面.....	36
6.10 历史记录界面.....	37
6.11 异常记录界面.....	39
6.12 管理员修改密码界面.....	40
6.13 管理会员界面.....	41
第七章 结论.....	41
参考文献.....	42
致 谢 .....	43
附录 .....	44

## 图目录

图 1.1 朝阳银发园.....	6
图 1.2 台湾人口比例图.....	7
图 2.1 LED 无光 .....	10
图 2.2 LED 发光.....	10
图 2.3 光信号转换过程.....	11
图 2.4 光谱.....	11
图 2.5 光信号核心原理.....	12
图 2.6 PPG 信号和 ECG 信号的对比.....	12
图 2.7 实际测量的 PPG 信号 .....	13
图 2.8 滤波处理.....	13
图 2.9 两种获取心率的方法.....	14
图 2.10 PPG 信号的 FFT 转变频域.....	14
图 2.11 测量血氧的原理.....	15
图 2.12 红光和 IR 透过皮肤的原始信号.....	15
图 2.13 红光和 IR 透过皮肤处理后的信号.....	15
图 3.1 SWOT 分析.....	18
图 5.1 甘特图 .....	21
图 5.2 整体架构.....	22
图 5.3 Android 端-Web 端连接 .....	22
图 5.4 管理员界面.....	23
图 5.5 用户界面.....	23
图 5.6 数据写入数据库.....	24
图 5.7 数据读取.....	25
图 5.8 管理员 ms_admin.....	25
图 5.9 用户数据 ms_user.....	26
图 5.10 紧急联系人 ms_contact.....	28
图 5.11 实时心率 ms_warn.....	29
图 5.12 病史-验证码-心率预值 .....	30
图 6.1 界面.....	32
图 6.2 注册界面.....	33
图 6.3 会员账户界面.....	33
图 6.4 修改会员资料界面.....	34
图 6.5 修改密码界面.....	34
图 6.6 紧急联络人界面.....	35
图 6.7 生理状况界面.....	36
图 6.8 心率阈值管理界面.....	36
图 6.9 实时心率界面.....	37
图 6.10 历史纪录界面.....	37
图 6.11 按周查询.....	38
图 6.12 按天查询.....	38
图 6.13 按时查询.....	38
图 6.14 异常记录界面.....	39

图 6.15 按 30 分钟查询.....	39
图 6.16 按一天内查询.....	40
图 6.17 按一周内查询.....	40
图 6.18 管理员修改密码.....	41
图 6.19 管理会员管理界面.....	41

## 表目录

表 5.1 admin .....	25
表 5.2 ms_user .....	27
表 5.3 ms_contact .....	28
表 5.4 ms_warn.....	29
表 5.5 ms_code .....	31
表 5.6 ms_heartrate.....	31
表 5.7 ms_sickness.....	31

# 心率监测管理系统

黄鑫宏 施养波 许浩望 王储君

朝阳科技大学资讯工程系

[cfyu@mail.cyut.edu.tw](mailto:cfyu@mail.cyut.edu.tw)

## 第一章 摘要

随着人口老龄化的发展，健康成为我们的首要目标，人体机能的下降，需要我们除了运动外更能监测到老人身体各项指标的变化。心率监测管理系统基于 Microsoft Band2 手环与智能手机 APP 之间无缝连接，在云服务和软硬件性能上追求极致。手环适合用户佩戴，时时反馈用户的心率状况，预测用户可能发生的心理疾病，成为用户必不可少的健康小助手。

关键字：心率监测，Microsoft Band2，APP，云服务，心率状况

## Abstract

With the development of the aging population, health has become our primary goal, the decline of the human body function, we need to be more in addition to exercise to monitor the changes in the indicators of the elderly. Heart rate monitoring management system based on Microsoft Band2 bracelet and smart phone APP seamless connection between the pursuit of the ultimate in the cloud services and hardware and software performance. Bracelet for the user to wear, the user's heart rate feedback from time to time, to predict the user's mental illness may occur, become essential to the health of the user assistant.

Keywords: heart rate monitor, Microsoft Band2, APP, cloud service, heart rate condition

### 1.1 选题背景

由于台湾社会人口年龄结构日趋高龄化，为了提供银发族一个「安全、安心」的起居环境与「愉快、尊严」的生活氛围，朝阳科技大学提出一个校级重点特色计划“发展银发乐活”。其中的子计划三为「建构银发族的生活照顾安全通报系统」，其中主要内容为建构银发园的行动安全通报系统，监控长者之生理状况（如：心跳，体温等）及长者之状态辨识（如：行走姿态，睡眠情况），若有发生异常状况，联结保全系统及监视系统立即预警通报，以应对长者所有可能发生的意外状况、防患于未然。基于这一缘由，我们小组在导师的建议之下我们参与了这一重点校级项目，并且选择心率预警这一方面的课题。



图 1.1 朝阳银发园

## 1.2 项目背景

目前，市场上关于可穿戴的智能设备越来越多，而且功能越来越炫酷。很多互联网公司都设计出了各种各样的智能手环，智能手表等等。其中：普通智能手环品种很多，具有代表型的品牌有：微软，小米，dido 等等，基本都偏向于运动的方向，都是通过定位，探测等一些方式实现一些监测功能。

现今较为高端的手环设备：iwatch、song、三星等，手环功能大同小异，除了拥有一般手环的功能外，还拥有独立通话，接收信息，照相，听歌一些手机常用的功能

然而这些可穿戴的智能设备主要都是针对年轻人来设计，忽略了老人这一块市场。但是往往老人是最需要检测自身的健康状况的。根据调查 1951 年新出生人口占全国人口的 20%，而 30 岁以下人口占到了全国人口的 70%-75%，随着时间的推算到 2031 年预计 40 岁以上人口将占到全国人口的 50%。如图一为台湾人口比例图。

生活水平的提高也使得的老年人的身体健康得到更多关注。调查显示，72% 的子女希望有一定的智能产品能够帮助父母省去每日主动测量各种数据的烦恼，86% 的子女希望能够借助云服务平台更好地监测父母的身体状况，而市场上的主流智

能手环并不适用于老人群体。

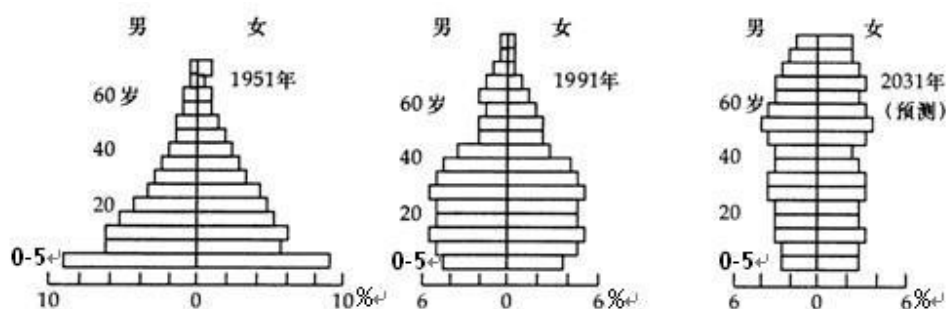


图 1.2 台湾人口比例图

### 1.3 目的

随着人口老龄化的发展，健康成为我们的首要目标，人体机能的下降，需要我们除了运动外更能监测到老人身体各项指标的变化。

因此我们预想将手环的功能可以运用到老人身体监测的方面来：例如通过心跳，计步，卡路里的计算预测心脏病，晕厥等病情的发生，可以让家属或者医院的工作人员更快速的预测到发病信息，得到及时的治疗。所以本小组成员针对这一缘由，基于 Microsoft Bank2 手环制作了适用于老人群体的心率监测系统。通过手环抓取佩戴者的心率实现了对手环佩戴者的实时心率监测，当佩戴者心率出现紧急情况，手环将对佩戴者的绑定亲属手机发出紧急提醒，并提供佩戴者的实时地理位置，从而达到了对佩戴者的第一时间救治。健康手环与智能手机 APP 之间无缝连接，在云服务和软硬件性能上追求极致。手环适合用户佩戴，时时反馈用户的心率状况，预测用户可能发生的心理疾病，成为用户必不可少的健康小助手。

## 第二章 可行性研究的前提

### 2.1 心脏疾病

#### 2.1.1 心脏疾病的治疗

英国人马太-格林现年 40 岁，患上慢性心脏病后，病情一直非常危险，但是他又苦于无法找到合适的捐献者。由于他的健康状况每况愈下，医生决定给他安装一个全套的人工心脏。心脏在背包携带的一个泵和一个电池的帮助下，把血液输送到格林的全身各处。

可惜的是，这项装置，并不是一个可以彻底治疗心脏病过的仪器，全人造心脏的部分零件的使用寿命可达 50 年，不过患者通常只会佩戴 3 年，在这期间这个人造心脏会“跳动”超过 2 亿次。而且这项仪器并不便宜，这项手术费用达 88 万人民币，而每年的维护费高达 18 万人民币，三年下来，总费用过百万，这绝不是一般家庭可以承受得起的范围。现今医学水平，心脏病的彻底根治对于一般收入水平来说，还是天方夜谭。

#### 2.1.2 心脏疾病的预防

既然我们打不过，那我们面对心脏病也只好是躲着，尽量不要让自己沾上。要躲着心脏病，我们就不能躲着连续心率监测了，这时候，面对连续心率监测，我们不得不重视起来了。

什么是心率的连续监测呢？所谓心率的连续监测就是指，在连续的一段时间内，对心率进行准确的监测。而通过对心脏进行连续心率监测，就可以对心率变异性进行预估。心率变异性是指逐次心跳周期差异的变化情况，可以反映自主神经系统活性和定量评估心脏交感神经与迷走神经张力及其平衡性，从而判断其对心血管疾病的病情及预防，是预测心脏性猝死和心律失常性事件的一个有价值的指标。

心率变异性反映了心脏交感神经和迷走神经活动的紧张性和均衡性。凡副交感神经张力降低的冠心病病人，其心室颤动的阈值低，容易发生心脏性猝死，而且心肌梗塞后的死亡率也增高。而原发性高血压患者则会出现交感神经张力增高，而副交感神经的张力降低的特征，而这些特征与高血压的严重程度呈明显的正相关。

它不仅可以是预测心脏性猝死最有价值的独立指标，而且还可以预测急性心肌梗塞后患者发生恶性心律失常危险，无心力衰竭的扩张型心肌病患者，植物神经功能普遍受损。通过心率功率谱分析，可以了解心肌病患者的病情。而对于糖尿病患者来说，它的自主神经系统损伤的评估又是及其有用的。

所以通过连续心率监测，我们就可以测出自身的心率变异性，从而预测多种致命性慢性病的发生及恶化。

#### 2.1.3 心率变异性

心率变异性（heart rate variability, HRV）是指连续心跳间 R-R 间期的微小涨落。HRV 反映了心脏交感神经和迷走神经活动的紧张性和均衡性，是一种检测自主神经性活动的非侵入性指标。近十年来的大量研究已充分肯定了自主神经活动与多种疾病有关系，特别是与某些心血管疾病的死亡率，尤其是猝死率有关。



通过心电图（ECG）对心率的微小涨落的变换和处理来获得心血管系统、自主神经系统等有关信息的信号分析过程即 HRV 分析，是近年来的研究热点之一。

#### 2.1.4 心率研究的意义

心率的测定在以下疾病的诊查中具有一定的作用：

1. 检测冠心病病人猝死的发生。凡副交感神经张力降低的冠心病病人，其心室颤动的阈值低，容易发生心脏性猝死，而且心肌梗塞后的死亡率也增高。
2. 了解副交感神经的受损情况。充血性心力衰竭患者植物神经机能普遍受损害而降低，但副交感神经受损更显著，可运用频域分析法进行监测。
3. 探讨高血压的始动机制。现已查明，原发性高血压患者的交感神经张力增高，而副交感神经的张力降低，且与高血压的严重程度呈明显的正相关。但是，老年高血压患者的上述变化不甚明显。通过植物神经功能测定，可以了解高血压的始动机制。
4. 监测和评价糖尿病患者的植物神经机能状况。糖尿病患者的植物神经和周围神经均受损害，且两者呈平行关系。通过心率变异性检查，可以了解植物神经机能的受损害程度。
5. 通过评价交感神经张力亢进情况，用于诊断年轻患者的血管迷走性晕厥。
6. 监测心肌病患者病情。无心力衰竭的扩张型心肌病患者，植物神经功能普遍受损。通过心率功率谱分析，可以了解心肌病患者的病情。
7. 监测心脏移植术后的排斥反应。心脏移植术后，患者心脏心率变异性显著降低甚至消失，一旦发生排斥反应，心率变异性则明显增高，因此，心脏移植术后应定期检查心率变异性，以了解和预防心脏排斥反应的发生，及时采取相应措施。
8. 对胎儿发育及产程进行监测。
9. 判断吸烟者植物神经功能受损害的程度。长期吸烟者，其植物神经功能均可受到损害，受损的严重程度与烟量及烟龄呈显著正相关。一般表现为交感神经张力增加和副交感神经张力降低。对嗜烟者监测心率变异性，可以对多种相关疾病进行预测。

## 2.2 Microsoft Band2 的介绍

微软于 2015 年 12 月 15 日发布的据称全球最好的智能可穿戴之一的微软 Microsoft Band2 智能手环。

### 2.2.1 外观特点

新 Microsoft Band2 采用了全新的曲面屏设计，同时采用大猩猩玻璃作为屏幕材料增加抗磨损能力。

### 2.2.2 主要功能

可以识别一些简单的语音命令，内置了 Global Positioning System、UltraViolet 监测、训练指导、睡眠追踪、卡路里追踪、通知等功能。微软 Band2 共计新增了 11 个传感器，包括气压计可以检测天气情况。另外，新 Microsoft Band 还增加了对高尔夫的支持，新 Band 可以判断用户是在练习还是在真的击球，从而记录相应的运动数据。

### 2.2.3 Microsoft Band2 光学心率传感器

### 2.2.3.1 心率传感器

1. 心电图 ECG
2. 生物阻抗 bioimpedance
3. 摄像头 Camera RGB 、wifi 等技术

在上述的几种方法中，最被人们熟悉的应该就是心电图。在医疗领域，通常使用心电图（ECG）测量生理电信号来实现心率和心脏活动的检测。但是由于测量 ECG 信号，常常要在身体多个部位连接传感器电极，在胸部和四肢之间最多可以连接 10 个电极。ECG 信号虽然精准并且信息丰富，但是考虑到穿戴设备的便携性和功能简单所以并没有在穿戴设备上广泛采用 ECG 技术。目前情况 ECG 还是在一些专业领域里面使用例如医院、体育等方面的研究。

生物阻抗传感器测量方法，目前市场上看到的好像只有 Jawbone 的 UP3 了，对于此技术网上的资料特别的少。不过通过亲身体会试戴 JawboneUP3，感觉这种技术应该比光学测心率的技术难度大但是应该更精准更可靠。最后的 Camera RGB 和 wifi 都是 MIT 研究出来的新技术，看起来都非常高端。其中一个是通过我们手机的摄像头就能检测出人体的体征变化。另一个则是通过我们家里的 wifi 信号就能测，也甚是高端。当然这两种还没有看到上市的产品所以就不多说了。

### 2.1.3.2 光学心率传感器

别名为光电容积脉搏波描记法，这个名字读起来实在是高端，其实说简单点就是利用光测量脉搏的一种技术。这种技术目前被广泛应用，目前市场上能看到采用这种技术的穿戴设备就有：AppleWatch、三星 Galaxy Gear S2、Moto 360、Microsoft Band2 等。

就拿 Microsoft Band2 来说，测量心率时底部的表盘会发出绿色的灯光，并且测量的时候手腕最好保持不动否则会影响测量结果。接下来将详细介绍光学心率测量的原理。

如下两张图是光学心率传感器。图 2.1 是 LED 没有发光的时候中间是一个光敏二极管，图 2.2 是传感器的 LED 发光的时候。



图 2.1 LED 无光



图 2.2 LED 发光

那么为什么通过 LED 灯发光就能测量心率呢？

当 LED 光射向皮肤，透过皮肤组织反射回的光被光敏传感器接受并转换成电信号再经过 AD 转换成数字信号，简化过程：光--> 电 --> 数字信号

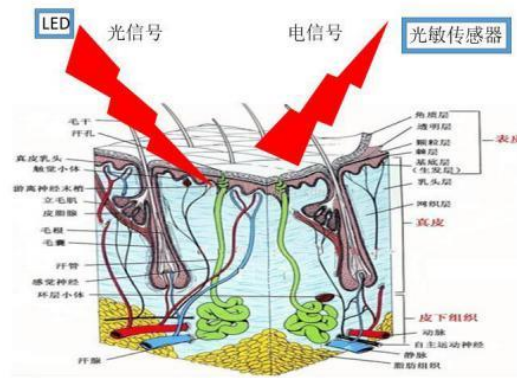


图 2.3 光信号转换过程

为什么大多数传感器都是采用的绿光呢？

我们先看看光谱的特点，从紫外线到红外线的波长是越来越长的。

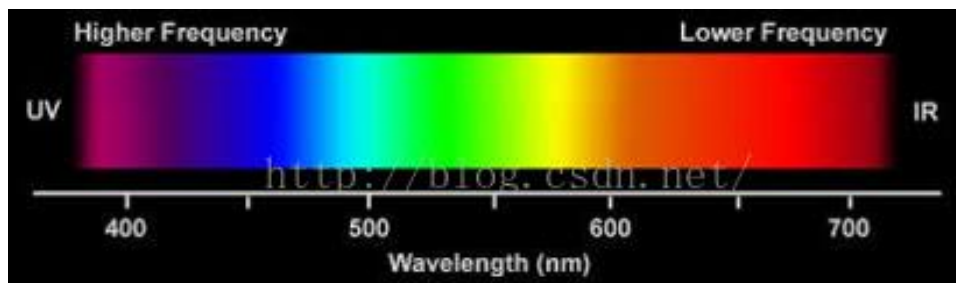


图 2.4 光谱

之所以选择绿光作为光源是考虑到一下几个特点：

1. 皮肤的黑色素会吸收大量波长较短的波
2. 皮肤上的水份也会吸收大量的 UV（紫外线）和 IR（红外线）部分的光
3. 进入皮肤组织的绿光(500nm)-- 黄光(600nm)大部分会被红细胞吸收
4. 红光和接近 IR 的光相比其他波长的光更容易穿过皮肤组织
5. 血液要比其他组织吸收更多的光
6. 相比红光，绿（绿-黄）光能被氧合血红蛋白和脱氧血红蛋白吸收

总体来说，绿光-- 红光能作为测量光源。早起多数采用红光为光源，随着进一步的研究和对比，绿光作为光源得到的信号更好，信噪比也比其他光源好些，所以现在大部分穿戴设备采用绿光为光源。但是考虑到皮肤情况的不用（肤色、汗水），高端产品会根据情况自动使用换绿光、红光和 IR 多种光源。虽然知道了上面的几个特点，但是还不足以弄清楚为什么通过光照就能测出心率、血氧等参数呢？

下图就解释了核心原理

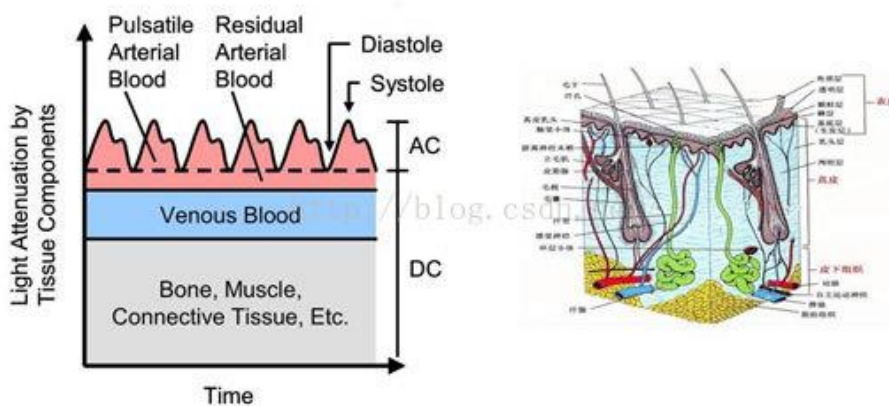


图 2.5 光信号核心原理

当光照透过皮肤组织然后再反射到光敏传感器时光照有一定的衰减的。像肌肉、骨骼、静脉和其他连接组织等等对光的吸收是基本不变的（前提是测量部位没有大幅度的运动），但是血液不同，由于动脉里有血液的流动，那么对光的吸收自然也有所变化。当我们把光转换成电信号时，正是由于动脉对光的吸收有变化而其他组织对光的吸收基本不变，得到的信号就可以分为直流 DC 信号和交流 AC 信号。提取其中的 AC 信号，就能反应出血液流动的特点。我们把这种技术叫做光电容积脉搏波描记法 PPG。

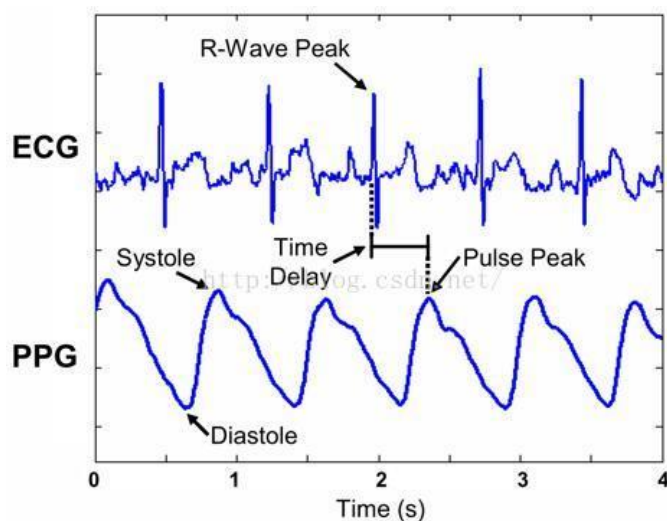


图 2.6 PPG 信号和 ECG 信号的对比

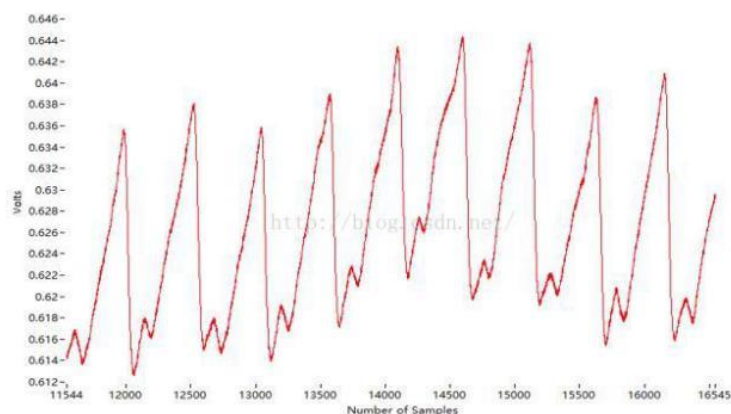


图 2.7 实际测量的 PPG 信号

所以，只要测得到的 PPG 信号比较理想算出心率也不算什么难事。但是事实总是残酷的，由于测量部位的移动、自然光、日光灯等等其他的干扰，最终测到的信号可能是下面的这种，所以要通过很多方法进行滤波处理

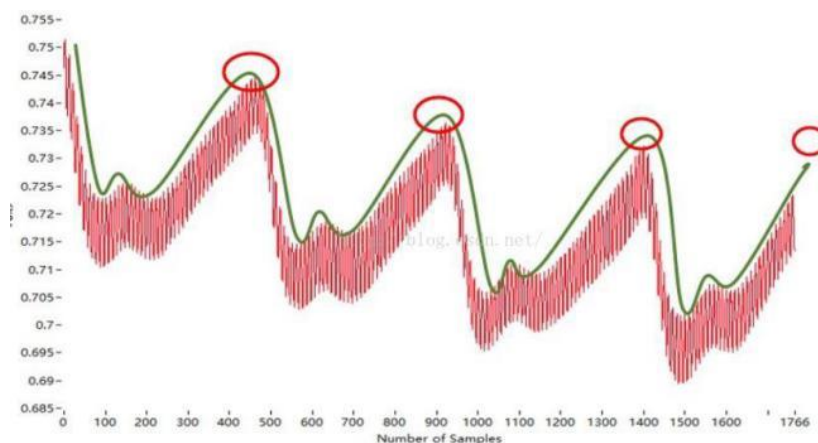


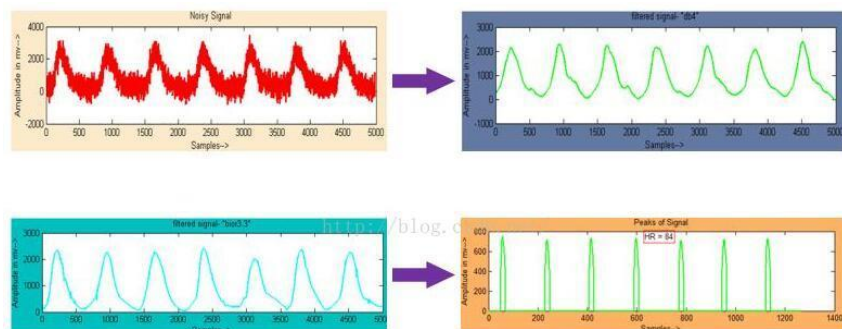
图 2.8 滤波处理

对于 PPG 信号的处理，目前已知道的有两种方法。一种是时域分析，即算出一定时间内 PPG 信号的波峰个数，另一种是通过对 PPG 信号进行 FFT（快速傅氏变换）变换得到频域的特点。

时域方法：

通过对原始的 PPG 信号进行滤波处理，得到一定时间内的波峰个数，然后既可算出心率值





- Heart Rate = No. of peaks x 12
- Respiratory rate = Heart Rate ÷ 2

图 2.9 两种获取心率的方法

假设连续采样 5 秒的时间，在 5s 内的波峰个数为 N，那么心率就是  $N \times 12$  就跟把脉一样。

频域分析：

上面分析过，我们把血液流动对光吸收转变成了 AC 信号，如果对于进行 FFT 变换，那么就能看到频域的特点。如下图就是对 PPG 信号的 FFT 转变

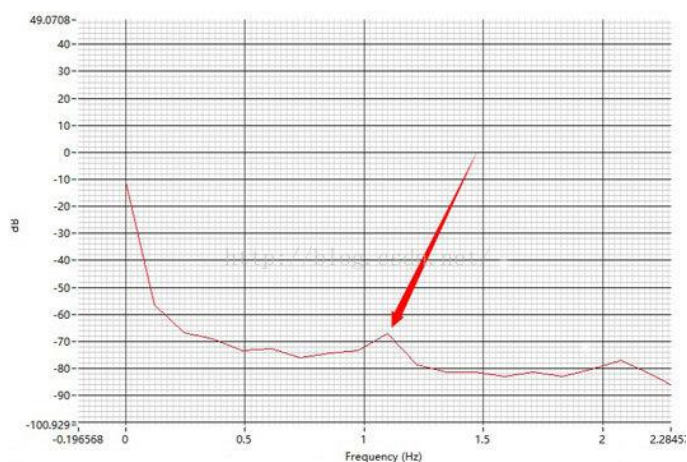


图 2.10 PPG 信号的 FFT 转变频域

上图中的频域图，0Hz 的信号很强，这部分是骨骼、肌肉等组织的 DC 信号，在 1Hz 附近有个相对比较突出的信号就是血液流动转变的 AC 信号。假设测得到的频率  $f = 1.2\text{Hz}$

那么心率 HeartRate  $HR = f \times 60 = 1.2 \times 60 = 72$

最后再简单提一下血氧的测量，相比心率血氧测量难度较大而且精度不算太高。测量血氧的原理图下图所示

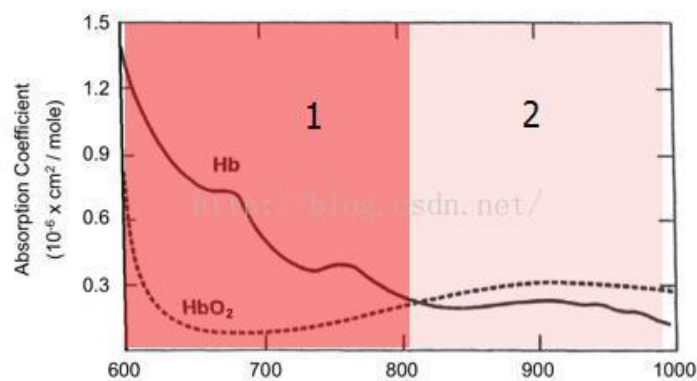


图 2.11 测量血氧的原理

由于血液中含有的氧合血红蛋白  $\text{HbO}_2$  和血红蛋白  $\text{Hb}$  存在一定的比例，简单说也就是含氧量吧。上面的图表示了氧合血红蛋白  $\text{HbO}_2$  和血红蛋白  $\text{Hb}$  对波长 600~1000nm 的光吸收特性，从图中可以看出上 600~800nm 间  $\text{Hb}$  的吸收系数更高，800~1000 之间  $\text{HbO}_2$  的吸收系数更高。所以可以利用红光（600~800nm）和接近 IR（800~1000nm）的光分别检测  $\text{HbO}_2$  和  $\text{Hb}$  的 PPG 信号，然后通过程序处理算出相应的比值，这样就得到了血氧值。

但是由于光源不同，直接利用红光和接近 IR 的光进行信号对比是不可靠的，因为红光和 IR 透过皮肤组织也会产生不同的吸收。下图是红光和 IR 透过皮肤的原始信号示意图

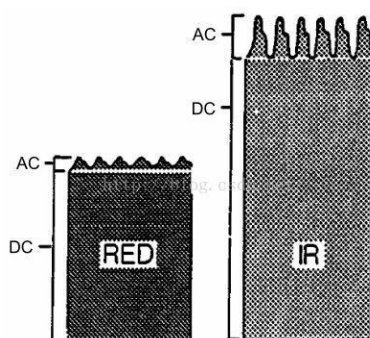


图 2.12 红光和 IR 透过皮肤的原始信号

上面分析说过，DC 部分是光透过皮肤组织转换成的直流信号，AC 是血液流动产生转换成的交流信号。由于皮肤组织对红光和 IR 的吸收程度不同，DC 部分自然也就不一样。为了能共“公平对待”两种光源的 PPG 信号，所以需要处理原始信号一下。下图示意了处理后的信号（DC 部分相等）

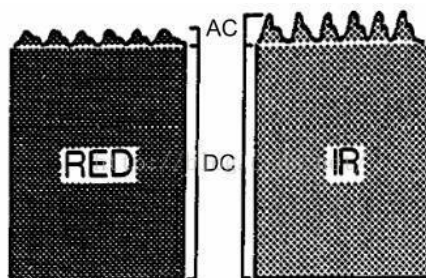


图 2.13 红光和 IR 透过皮肤处理后的信号

通过一定的比例计算，公平对待 Red 和 IR 的 PPG 信号。这样计算出来的 Hb 和 HbO<sub>2</sub> 比例才可靠。以上就是针对光学传感器进行的介绍。

## 第三章 市场分析与风险评估

### 3.1 宏观环境分析

#### a) 目前市场上的手环

当前环境下，智能可穿戴设备已经开始慢慢深入人们的生活。在专家看来，可穿戴设备有着很大的市场。而现今已有多种此类设备研发成功并且投入使用，在这当中有一类被称为手环，显而易见，就是戴在手腕上的智能设备。手环由于携带方便，重量轻巧且操作简单，被应用于多个领域，不仅在国内，在国际上都有着广泛的运用。医疗领域中被应用最多。中研普华公司研制的智能手环 JawboneUP，这款手环是目前在中国最热销的一款运动手环设备。此外，雷蛇公司作为一家计算机外设厂商推出了一款名为 Nabu 的智能穿戴手环。Nabu 除了拥有与其它智能穿戴设备相似的健身追踪和提示通知功能之外，采用了双显示屏设计，并且外形非常小巧。早在 12 年美国 Lark 公司就推出一款名为“larklife”的智能手环，该手环可以利用 iOS 软件监测如睡眠、饮食等的健康状况的细微变化。小米公司也在近日发布了小米手环，可以实现多种功能。在运动领域，NIKE 推出 Nike FuelBand，支持计步监控及蓝牙 4.0 等众多功能，但不支持心率监测和定位功能。

就目前来看，智能手环市场尚属初级阶段，虽经近几年从无到有，从国外专属，到国内纷推，手环业也从当初的新兴期转为发展期，但整体而言，还比较不成熟，就国内市场而言，呈现出一种发展乱象，各大小厂商都在生产，手环市场也处于质量不稳定状态，而产品本身没有在准确度和创新性上做出很大的突破。

#### b) Microsoft Band2 的选择

Microsoft band2 在功能上集合了所有手环的功能，新增了 11 种传感器，堪称目前功能最强大的手环。在应用上，Microsoft Health 提供了传感器功能应用的下载，方便使用者在手环的功能上进行更加强化的设计。

### 3.2 竞争对手分析

#### （一）国内竞争对手

国内手环市场相对国外来说比较不稳定。首先，国内市场优点是售价低。这使得人人都可以买手环来用，但是售价低的负面影响也是很明显的，在低售价下怎么会有高端的产品，于是，健康手环的效果不明显，导致大众对于手环失去信心，产生抵触情绪，国内市场面临售价低，同质化严重的问题。其次，手环遍布大街小巷，非常平民化，制造手环的厂商门槛高低不齐，由于我们的主要客户群体是平民老百姓，那么占据这个市场的产品极有可能是这些低价品。

#### （二）国外竞争对手



国外竞争力相对较强，欧美市场比国内成熟不少，用户接受度高。

### （三）Microsoft 手环

我们的软件是基于微软的手环而开发的应用，在局限性上，微软手环的普及给我们造成了很大的难度。所以我们只能用我们应用的功能打动使用者，从而使他们选择微软手环，进一步使我们的应用得到更广的普及。

## 3.3 市场局限

### （一）低端产品占据市场

低端产品打响了口号，却没有实质性的进展，却让大众对此类健康手环很是疲劳，没有耐心去倾听或者购买。这使得我们在推行新软件的时候在进入大众视野从而进入大众关注范围中有些难度。

### （二）手环功能应用的创新

每款手环功能类似，是消费者在购买时没有选择甚至盲目选择，面对手环的功能也是知之甚少，造成了盲目使用。

### （三）没有可靠推动力

手环作为比较新进的产品，很少有品牌公司对它关注，市面上的手环知名度不高，没有可靠的推动力，最终走向消亡的产品也是很多。所以需要有一双坚实的双手进行推动。

## 3.4 SWOT 分析

软件的拓展性：此类产品的拓展性依赖于硬件的功能性，所以拓展性较差，难以延伸新的功能

软件的易用性：此产品是针对老年人的产品，在产品的易用性和方便性上难以很好的满足老年人的使用习惯。尤其在在安装方便可能难以简单化，所以应制定好详细的说明让老年人的家人或监护人也可以方便的安装使用。

下图为 SWOT 分析图：



图 3.1 SWOT 分析

## 第四章 技术分析

### 4.1 硬件技术分析

微软 Microsoft Band2 在应用上，Microsoft Health 提供了心率传感器功能应用的下载，这使我们在硬件技术上省去了设计的麻烦。

### 4.2 软件技术分析

#### 4.2.1 手环上的数据传输系统

该部分先经由手环与手机的蓝牙连接，手环向手机不断传输心率数据，再凭借手机的数据传输部分通过移动网络上传由传感器传来的各项数据并上传到 WEB 端进行数据整合。

#### 4.2.2 手机上的应用

##### 4.2.2.1 突发情况提示系统

通过手环经由手机传给 WEB 端的实时信息进行全天候监控，在各项指标超过设定的警戒值时给予手机端推送危险提示以提示老人的身体可能出现状况。

##### 4.2.2.2 心率的显示

与手环连接的手机能得到第一时间的心率数据。

#### 4.2.3 网页上的应用

HTML5 开发前端交互页面，PHP 开发后台管理与数据库连接，使用 json 进行数据交换及相关操作。

#### 4.2.4 数据库

使用 MYSQL 进行数据库的表单和数据的建立。

### 4.3 软件要求

#### 4.3.1 功能

##### 4.3.1.1 基本功能

##### (1) 网页端会员登入系统

##### a) 基本信息

##### ● 账户信息

- 修改资料
- 修改密码
- 紧急联络人管理
- 生理状况管理
- 心率阈值管理

b) 手环管理

- 实时心率
- 历史记录
- 异常管理

(2) 网页端管理员管理系统

a) 会员管理

- 实时心率
- 设置心率阈值
- 编辑
- 历史记录
- 异常记录

b) 账户管理

(3) 基于 Microsoft Band2 实现心率的读取上传，并以曲线的形式在网页端得到实时状况

(4) 心率警戒值的设定

(5) 生理状况预测

(6) 手环地理位置的实时定位

每五分钟上传一次位置定位信息，在发送预警短信时发送当前地理位置信息

#### 4.3.1.2 紧急情况

人心率突发状况，而无人身边时，通过手环传给子女手机端的信息，在心率超过设定的警戒值时给予手机端危险提示以提示子女老人的身体可能出现状况。

### 4.4 生理状况预测 ( 六条预测算法 )

目前世界上公认的心电数据库有三个，分别是美国麻省理工学院提供的

MIT-BIH 心电数据库，和美国心脏学会提供的 AHA 以及欧洲的 ST-T 心电数据库。其中 MIT-BIH 数据库近几年应用比较广泛。为了方便研究，该数据库的所有者也将越来越多的数据放到了互联网上。

MIT-BIH 数据库近年来在心电图产品认证、心电图教学等方面应用比较广泛，尤其是 IEC60601-2-47、AAMI EC57 等标准中明确规定使用该数据库作为设备(软件)心律失常自动分析功能的性能测试数据库。

通过 MIT-BIH 心电数据库和有关心率预测资料的调查和整理，本小组归纳总结出了以下这六条预测心脏疾病的六条预测算法：

1. 取最近 1000 次的心率计算平均心率在 80-85 时，可能引发冠心病。
2. 取 100 次平均心率（每 1000 次心率的平均值），每次平均心率增加 5bpm 时，可能引发心肌梗塞。
3. 取 100 次平均心率（每 1000 次心率的平均值），每次平均心率减少 5bpm 时并且小于 40，可能引起血糖升高，导致糖尿病。
4. 取最近 1000 次的心率计算平均心率在 80-85 时，大于 112 时提示可能患有心脏病。
5. 取最近 1000 次的心率计算平均心率，大于 80bpm 在生理建议中提示考虑治疗。
6. 当患者填写病史中存在急性心肌梗塞，取 1000 次心率的平均值，当平均小于 64 提示没有危险，当大于 75 时，提示可能存在病发危险。

## 4.5 性能

本软件手环基于 Microsoft Band2 在样式设计上美观，符合人体工学。在功能上，针对老人群体特点，操作极简，功能稳定适用。APP 的设计交互性更加人性化，方便简单。

## 4.6 软件目标

基于 Microsoft band2 手环与智能手机 APP 之间无缝连接，在云服务和软件性能上追求极致。手环适合用户佩戴，时时反馈用户的心率状况，预测用户可能发生的心理疾病成为用户必不可少的健康小助手。

## 4.7 条件和限制

心率监测管理软件的设计需要掌握软件方面的知识，同时也需要了解云服务以实现数据的时时上传和共享。以现已设计出的智能手环的功能为基础进行设计，因为是在微软手环的基础上进行的设计，所以在使用方面，需要先下载 Microsoft Health 进行手环的连接，在连接到 APP，使用起来较为不方便。

第五章 详细设计及功能实现

硬件设备：Microsoft Band 2，智能手机，搭载 windows 系统的计算机  
本地测试环境：XAMPP，Android Studio(App)，Phpstorm(Web)  
PHP 框架：ThinkPHP

5.1 设计分工与时间甘特图

详细设计分工：

- 黄鑫宏：主要 Web 端的框架设计，前端与后端页面开发，及实现主要功能，协助 app 端的开发。
- 施养波：Android App 的设计和开发，处理传感器接收的数据，协助 Web 端的开发。
- 王储君：实现数据库表单设计和实现，协助 Web 与 App 端处理数据
- 许浩望：搜集心率信息资料，整理编写生理预测规则，分析预测算法，需求分析、可行性分析等文档编辑。

甘特图：

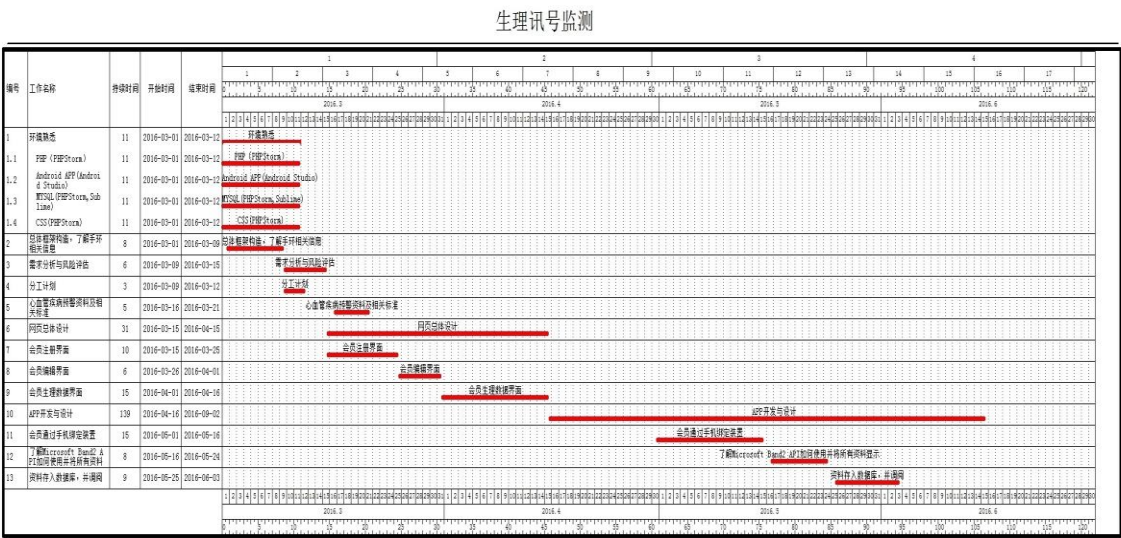


图 5.1 甘特图

## 5.2 设计流程图

### (1) 整体架构

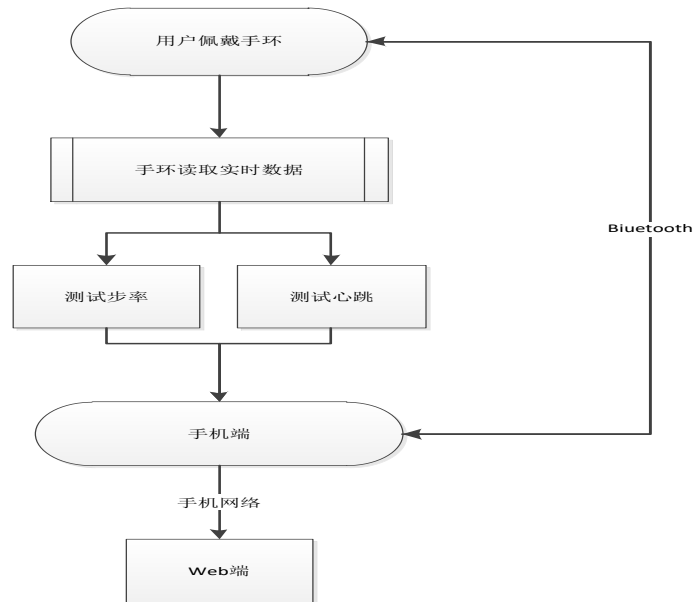


图 5.2 整体架构

### (2) android 端-WEB 端连接流程图：

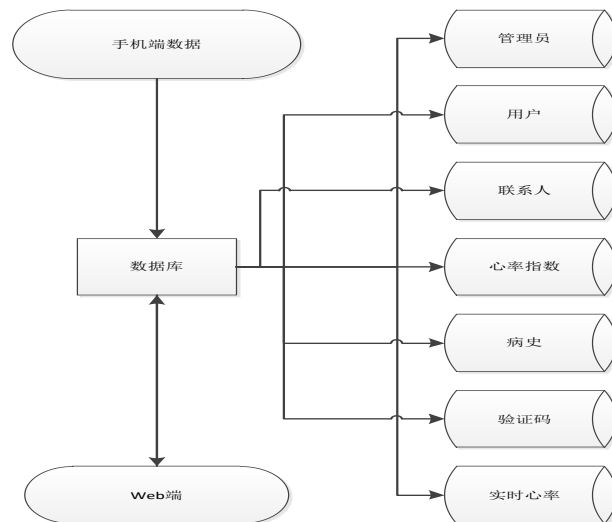


图 5.3 Android 端-Web 端连接

(3) 管理员界面流程图

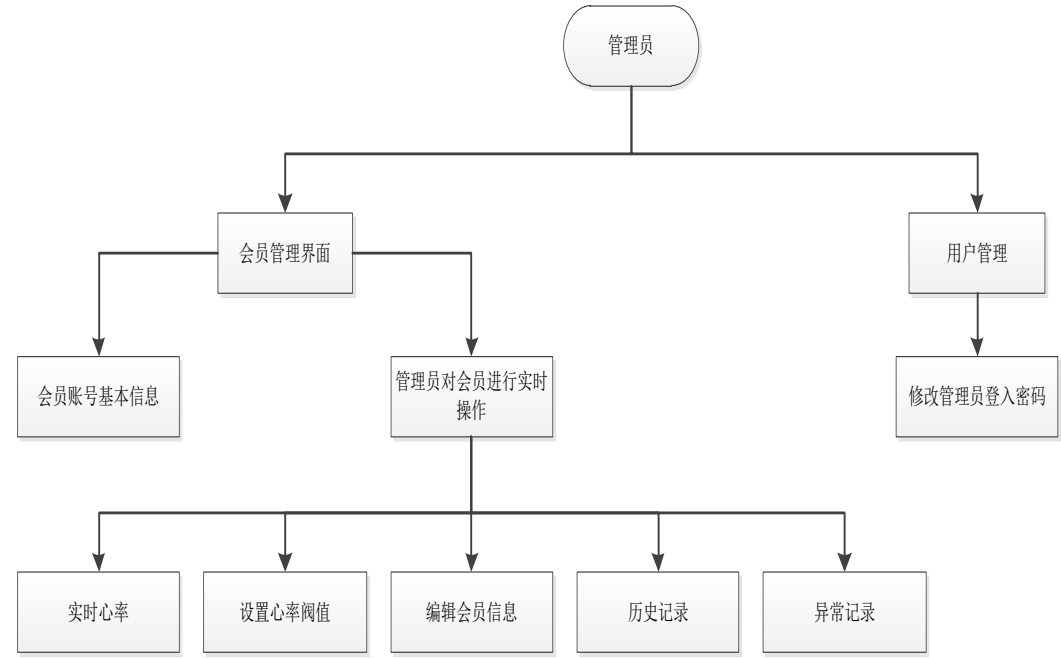


图 5.4 管理员界面

(4) 用户界面流程图

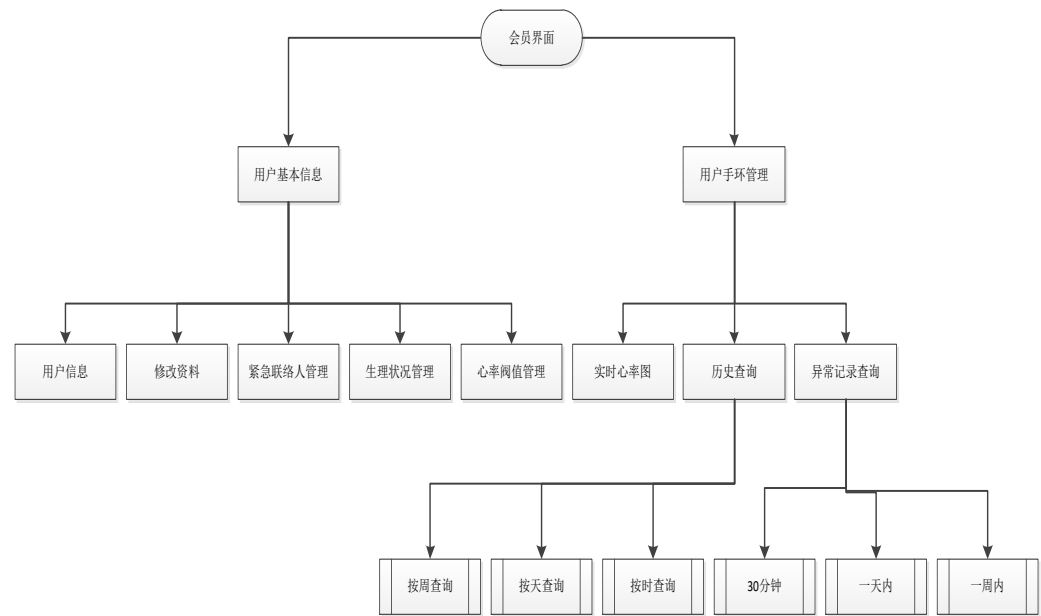


图 5.5 用户界面

### 5.3 数据库设计

(1) 数据写入与读取数据流程：

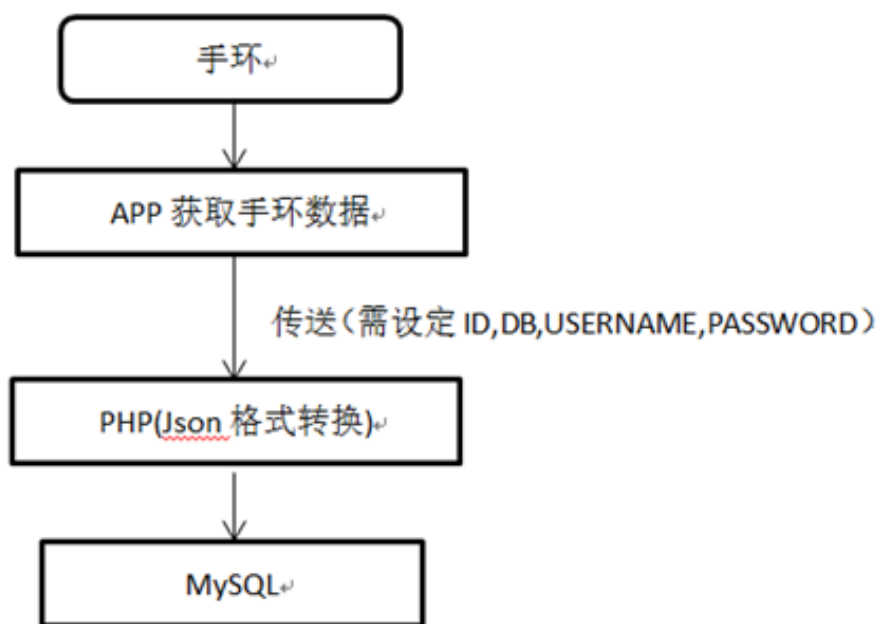


图 5.6 数据写入数据库

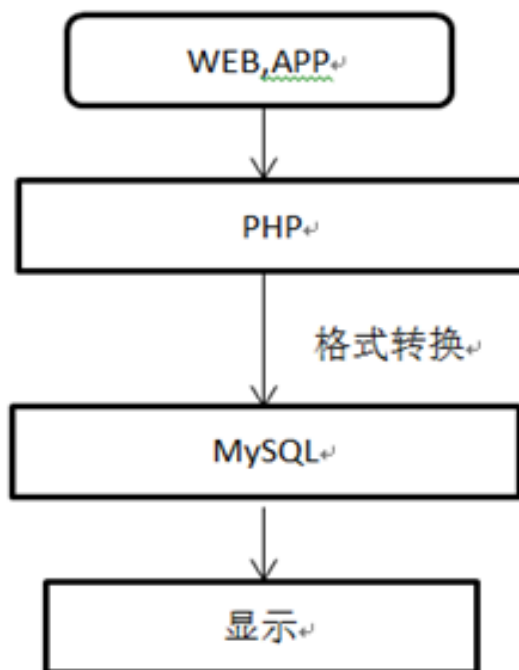




图 5.7 数据读取

(2) 管理员 ms\_admin 表

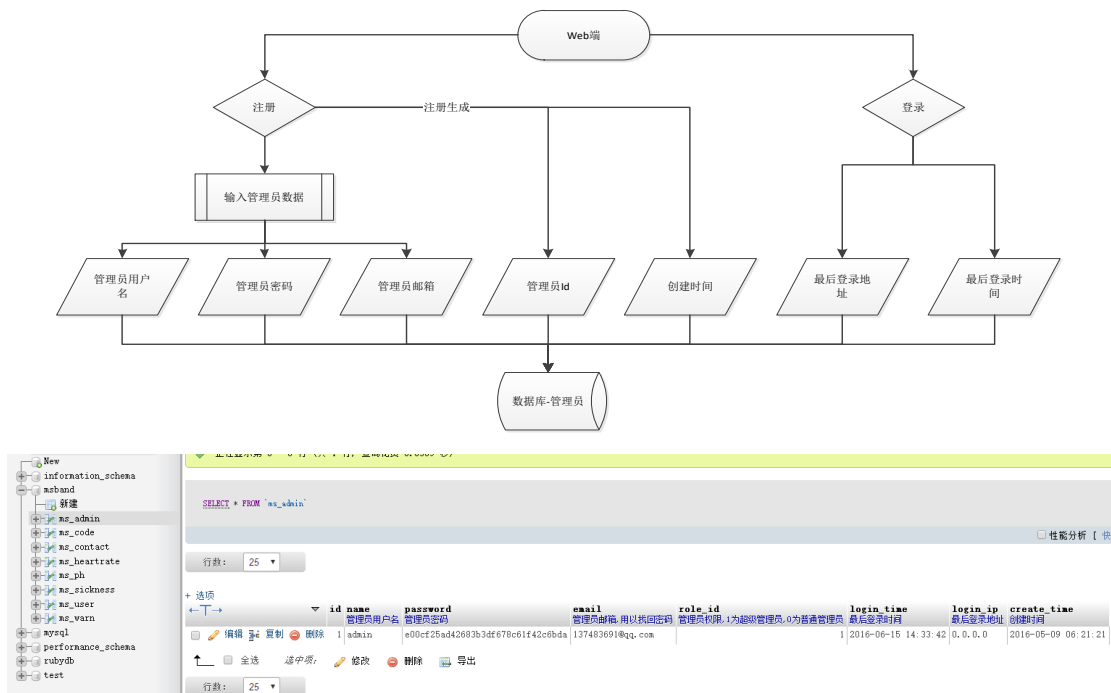


图 5.8 管理员 ms\_admin

中文名称	代码	数据类型	长度	非空	备注
编号	Id	Int	11	否	
管理员姓名	Name	varchar	20	否	
管理员密码	Password	varchar	32	否	
邮箱	Email	varchar	100	否	
管理员权限	Role_id	int	11	否	
登录时间	Login_time	timestamp		是	
登录地址	Login_ip	varchar	20	是	
创建时间	Create_yime	timestamp		是	

表 5.1 admin

(3) 用户数据 ms\_user 表:

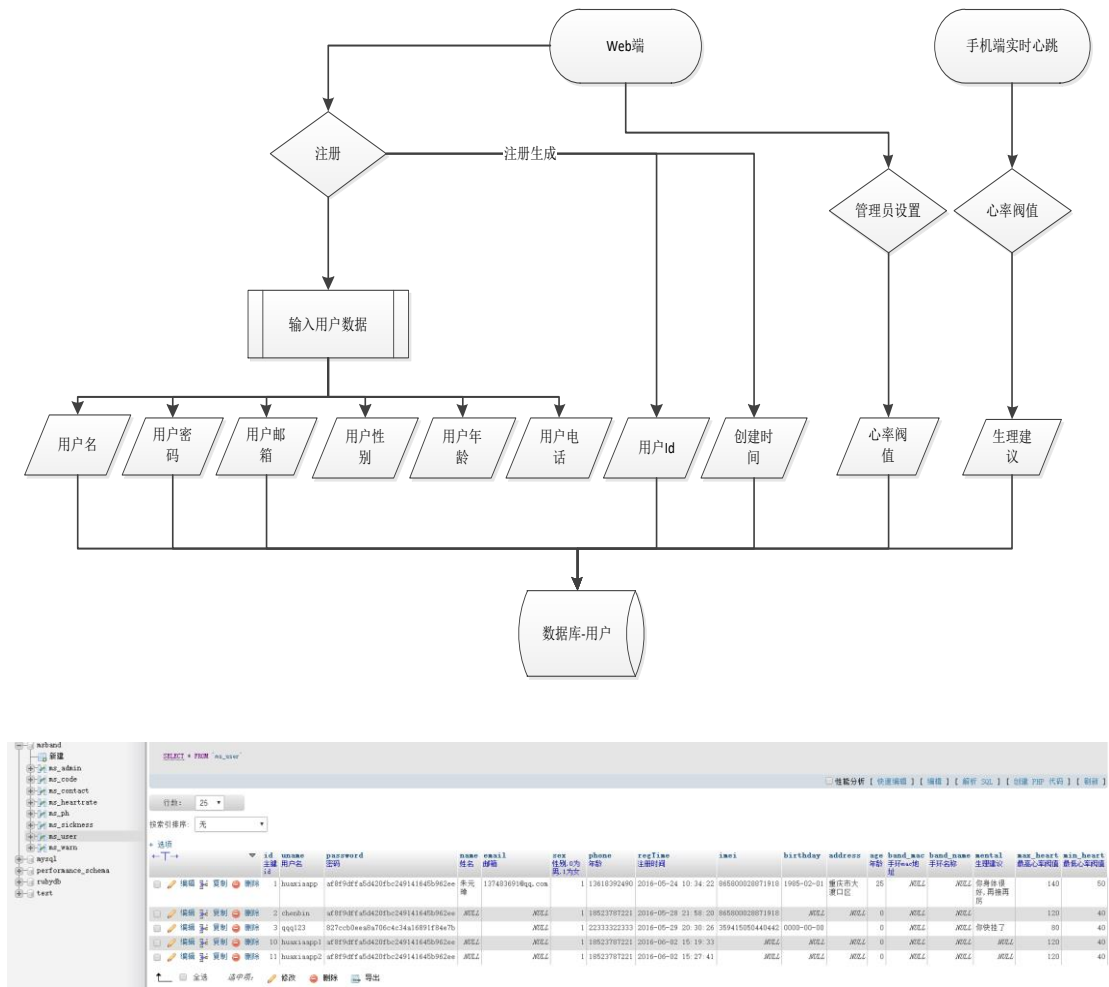


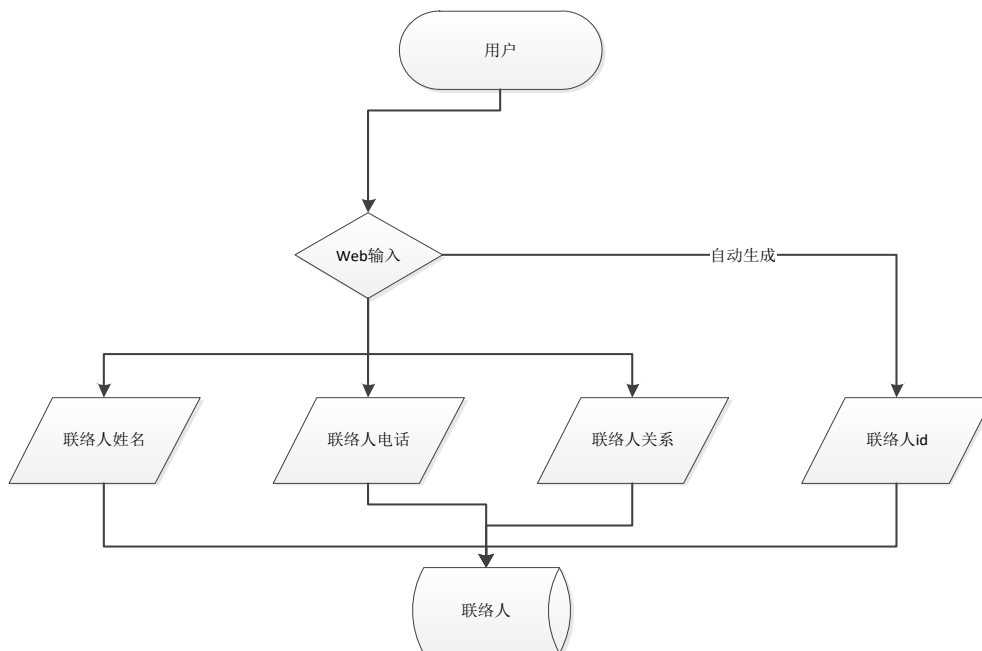
图 5.9 用户数据 ms\_user

中文名称	代码	数据类型	长度	非空	备注
编号	Id	Int	11	否	
用户账号	uname	varchar	20	否	
密码	Password	varchar	32	否	
姓名	name	varchar	45	是	
邮箱	Email	varchar	100	是	
性别	sex	Tinyint	1	否	
电话	phone	varchar	11	是	
注册时间	regtime	timestamp		否	

IMEI	imei	varchar	15	是	
生日	birthday	date		是	
住址	address	varchar	255	是	
年龄	age	int	11	是	
手环 mac	Band mac	varchar	50	是	
手环名称	Band name	varchar	100	是	
生理建议	mental	text		是	
最高心率阀	Max heart	int	11	否	
最低心率阀	Min heart	int	11	否	

表 5.2 ms\_user

(4) 紧急联系人 ms\_contact 表:



msband
新建
ms\_admin
ms\_code
ms\_contact
ms\_hearttrate
ms\_ph
ms\_sickness
ms\_user
ms\_warn
mysql
performance\_schema
rubymb
test

```
SELECT * FROM `ms_contact`
```

行数: 25

按索引排序: 无

+ 选项

id	name 姓名	phone 手机	relation 关系	uid 用户id
4	陈尧	18523787221	父亲	1
5	李香兰	18523789195	母亲	1
6	hongzai	0903663430	父亲	3

编辑
复制
删除

编辑
复制
删除

编辑
复制
删除

↑

全选

选中项:

修改

删除

导出

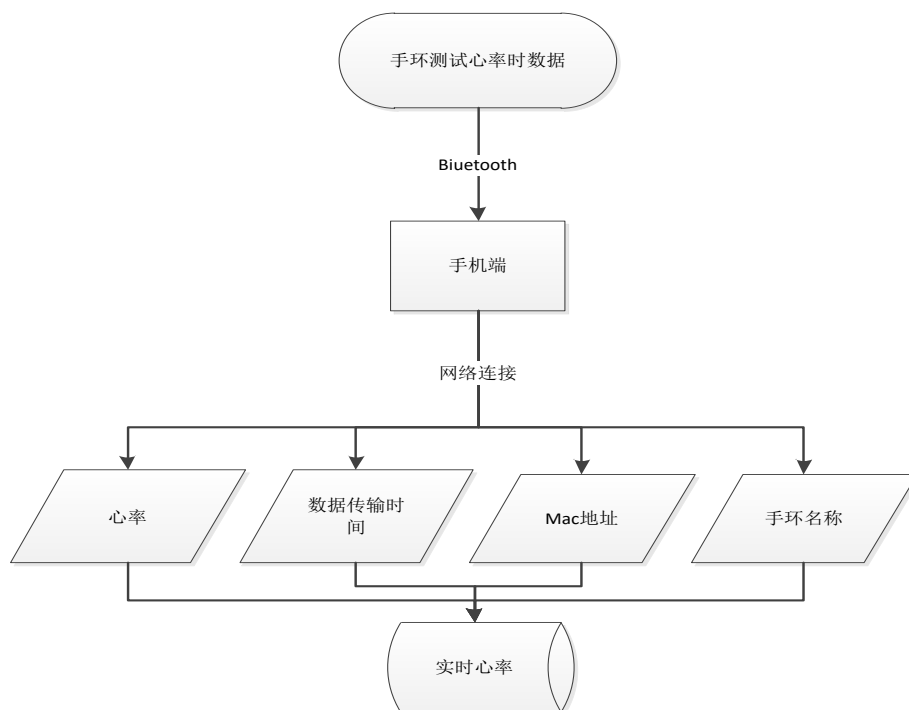
行数: 25

图 5.10 紧急联系人 ms\_contact

中文名称	代码	数据类型	长度	非空	备注
编号	Id	Int	11	否	
联络人姓名	Name	varchar	20	否	
联络人手机	phone	varchar	11	否	
关系	relation	varchar	10	否	
用户 id	u_id	int	11	否	

表 5.3 ms\_contact

(5) 实时心率 ms\_warn 表:



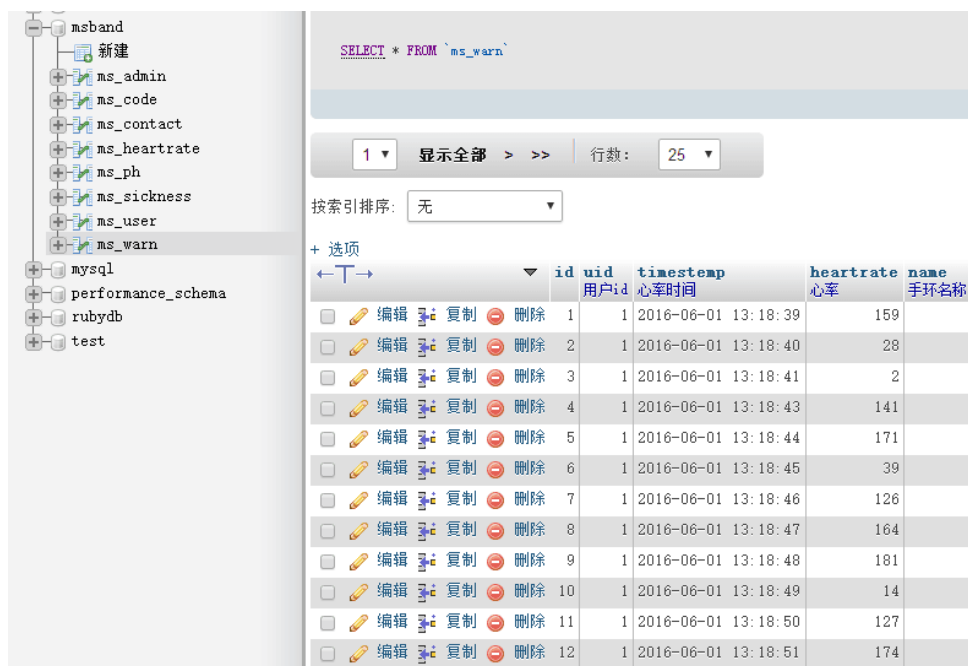
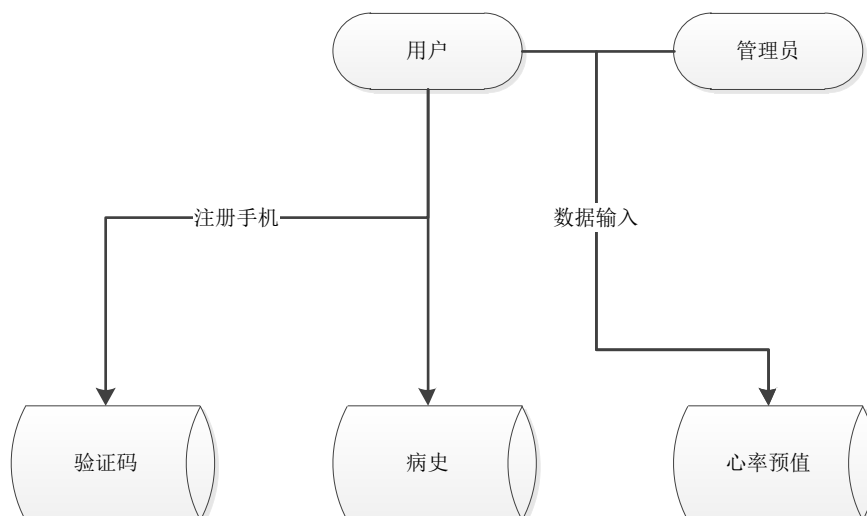


图 5.11 实时心率 ms\_warn

中文名称	代码	数据类型	长度	非空	备注
编号	Id	Int	11	否	
用户 id	uid	int	11	否	
心率时间	timestamp	timestamp	32	否	
心率	heartrate	int	11	否	
手环名称	name	varchar	100	否	

表 5.4 ms\_warn

(6) 病史 ms\_sickness、验证码 ms\_code、心率预值 ms\_hearttrate



msband	新建	ms_admin	ms_code	ms_contact	ms_hearttrate	ms_ph	ms_sickness	ms_user	ms_warn	mysql	performance_schema	rubydb	test
SELECT * FROM 'ms_code'													
行数: 25													
按索引排序: 无													
+ 选项													
id code validDate targetPhone statu sendDate validstate													
1 024486 2016-06-02 14:35:18 18523787221 发送状态:1为成功,0为失败 1 2016-05-28 18:16:26 1													
2 397272 2016-06-02 14:35:20 18523787221 1 2016-05-28 18:20:33 1													
3 710299 2016-06-02 14:35:23 15823789195 1 2016-05-28 21:13:07 1													
4 336632 2016-06-02 14:35:25 15823789195 1 2016-05-28 21:13:20 1													
5 214602 2016-06-02 14:35:27 18523787221 1 2016-05-28 21:18:24 1													
6 373435 2016-06-02 14:35:29 18523787221 1 2016-05-28 21:28:33 1													
7 489052 2016-06-02 14:35:31 18523787221 1 2016-05-28 21:43:20 1													
8 215470 2016-06-02 14:35:33 18523787221 1 2016-05-29 10:10:52 1													
9 043920 2016-06-02 14:35:34 00886903663 1 2016-05-29 20:26:31 1													
10 159983 2016-06-02 14:35:36 130 0785 11 1 2016-05-29 20:28:09 1													
11 800718 2016-06-02 14:35:38 2233322333 1 2016-05-29 20:30:09 1													
12 989520 2016-06-02 15:27:41 18523787221 1 2016-06-02 14:53:37 1													

msband	新建	ms_admin	ms_code	ms_contact	ms_hearttrate	ms_ph	ms_sickness	ms_user	ms_warn	mysql	performance_schema	rubydb	test
SELECT * FROM 'ms_hearttrate'													
1 > >> 行数: 25													
按索引排序: 无													
+ 选项													
id mac name rates addTime uid commit_timestamp													
1 58:82:A8:CE:C7:D5 Hx.Band c7:d5 [{"timestamp":1464615091,"heartRate":70}, {"timestamp":1464615095,"heartRate":70}, {"timestamp":1464615101,"heartRate":70}, {"timestamp":1464615105,"heartRate":70}, {"timestamp":1464615110,"heartRate":70}, {"timestamp":1464615115,"heartRate":70}, {"timestamp":1464615120,"heartRate":70}] 2016-05-30 21:31:32 3 2016-05-30 21:31:35													
2 58:82:A8:CE:C7:D5 Hx.Band c7:d5 [{"timestamp":1464615095,"heartRate":70}, {"timestamp":1464615101,"heartRate":70}, {"timestamp":1464615105,"heartRate":70}, {"timestamp":1464615110,"heartRate":70}, {"timestamp":1464615115,"heartRate":70}, {"timestamp":1464615120,"heartRate":70}] 2016-05-30 21:31:37 3 2016-05-30 21:31:40													
3 58:82:A8:CE:C7:D5 Hx.Band c7:d5 [{"timestamp":1464615101,"heartRate":70}, {"timestamp":1464615105,"heartRate":70}, {"timestamp":1464615110,"heartRate":70}, {"timestamp":1464615115,"heartRate":70}, {"timestamp":1464615120,"heartRate":70}] 2016-05-30 21:31:42 3 2016-05-30 21:31:45													
4 58:82:A8:CE:C7:D5 Hx.Band c7:d5 [{"timestamp":1464615105,"heartRate":70}, {"timestamp":1464615110,"heartRate":70}, {"timestamp":1464615115,"heartRate":70}, {"timestamp":1464615120,"heartRate":70}] 2016-05-30 21:31:47 3 2016-05-30 21:31:50													
5 58:82:A8:CE:C7:D5 Hx.Band c7:d5 [{"timestamp":1464615110,"heartRate":70}, {"timestamp":1464615115,"heartRate":70}, {"timestamp":1464615120,"heartRate":70}] 2016-05-30 21:31:52 3 2016-05-30 21:31:55													
6 58:82:A8:CE:C7:D5 Hx.Band c7:d5 [{"timestamp":1464615115,"heartRate":70}, {"timestamp":1464615120,"heartRate":70}] 2016-05-30 21:31:57 3 2016-05-30 21:32:00													
7 58:82:A8:CE:C7:D5 Hx.Band c7:d5 [{"timestamp":1464615120,"heartRate":70}] 2016-05-30 21:32:02 3 2016-05-30 21:32:05													

msband	新建	ms_admin	ms_code	ms_contact	ms_hearttrate	ms_ph	ms_sickness	ms_user	ms_warn	mysql	performance_schema	rubydb	test
SELECT * FROM 'ms_sickness'													
行数: 25													
按索引排序: 无													
+ 选项													
id name addtime type uid													
2 冠心病 2016-05-24 13:02:12 1 2 1													
3 胃病 2016-05-24 13:37:06 1 1 1													
4 头疼病 2016-05-24 13:37:35 2 1 1													
5 心脏病 2016-05-24 13:41:18 1 1 1													

图 5.12 病史-验证码-心率预值

中文名称	代码	数据类型	长度	非空	备注
Id	Id	Int	11	否	
手机验证码	code	varchar	10	否	
有效日期	validdate	timestamp		否	

目标手机号码	targetphone	varchar	11	否	
发送状态	statu	int	11	否	
发送日期	senddate	timestamp		否	
验证状态	validstate	Int	11	否	

表 5.5 ms\_code

中文名称	代码	数据类型	长度	非空	备注
ID	Id	Int	11	否	
mac 地址	mac	varchar	100	否	
手环名称	name	varchar	100	否	
心率数据	rates	text	50	否	
保存到库时间	addtime	timestamp	50	否	
用户 id	uid	int	11	否	
提交时间	Commit timestemp	timestamp		是	

表 5.6 ms\_heartrate

中文名称	代码	数据类型	长度	非空	备注
ID	Id	Int	11	否	
疾病名称	name	varchar	100	否	
添加时间	addtime	timestamp		否	
疾病类型	type	int	11	否	
用户 id	uid	int	11	否	

表 5.7 ms\_sickness

## 第六章 实际功能界面测试

### 6.1 登录界面

- 功能：管理员与用户登录界面，注册界面跳转
- 主要数据项：用户名、密码
- 实测截图

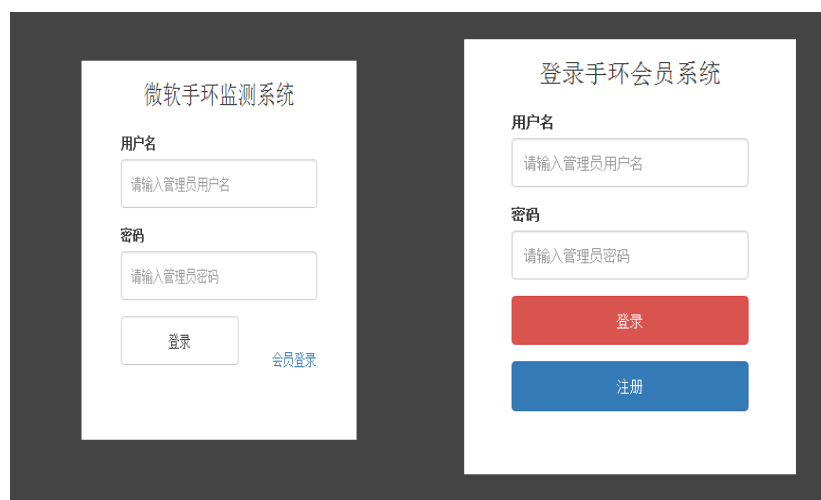


图 6.1 界面

### 6.2 注册界面

- 功能：用户注册界面



- 主要数据项：用户名、密码、手机号码、验证码、性别选择
- 实测截图

注册会员系统

用户名  
请输入用户名

密码  
请输入密码

重复密码  
请重复输入密码

手机号码  
请输入手机号码

手机验证码  
请输入手机验证码  
[发送验证码](#)

性别  
☒ 男 ☐ 女

[注册](#)

[登录](#)

图 6.2 注册界面

### 6.3 会员账户信息界面

- 功能：查看会员当前信息，帐号、姓名、注册时间等
- 主要数据项：
- 实测截图

心率监测会员系统    基本信息    手环管理

账户信息

修改资料

修改密码

紧急联络人管理

生理状况管理

心率阈值管理

账户信息

账户: qqq123

姓名:

电话: 0903663430

性别: 男

年龄: 10

生日: 1970年01月01日

地址:

注册时间: 2016-11-06 14:03:59

[编辑资料](#)    [管理紧急联系人](#)

图 6.3 会员账户界面

### 6.4 修改资料界面

- 功能：修改会员各项账户资料
- 主要数据项：姓名、手机号码、年龄、性别选择、出生年月、地址、手机 IMEI
- 实测截图

心率监测会员系统

基本信息

手环管理

账户信息

修改资料

修改密码

紧急联络人管理

生理状况管理

心率阈值管理

修改资料

账户名

qqq123

姓名

电话

0903663430

年龄

10

性别

☒ 男 ☐ 女

出生年月

0000-00-00

地址

手机IMEI

保存

图 6.4 修改会员资料界面

## 6.5 修改密码界面

- 功能：修改会员密码
- 主要数据项：旧密码、新密码
- 实测截图

心率监测会员系统

基本信息

手环管理

账户信息

修改资料

修改密码

紧急联络人管理

生理状况管理

心率阈值管理

修改密码

原密码

请输入原密码

新密码 (密码必须以字母开头,长度6-20位)

请输入新密码

重复新密码

请重复输入新密码

保存

图 6.5 修改密码界面

6.6 紧急联络人界面

- 功能：查看紧急联络人、增加紧急联络人
- 主要数据项：姓名、手机号码、关系
- 实测截图

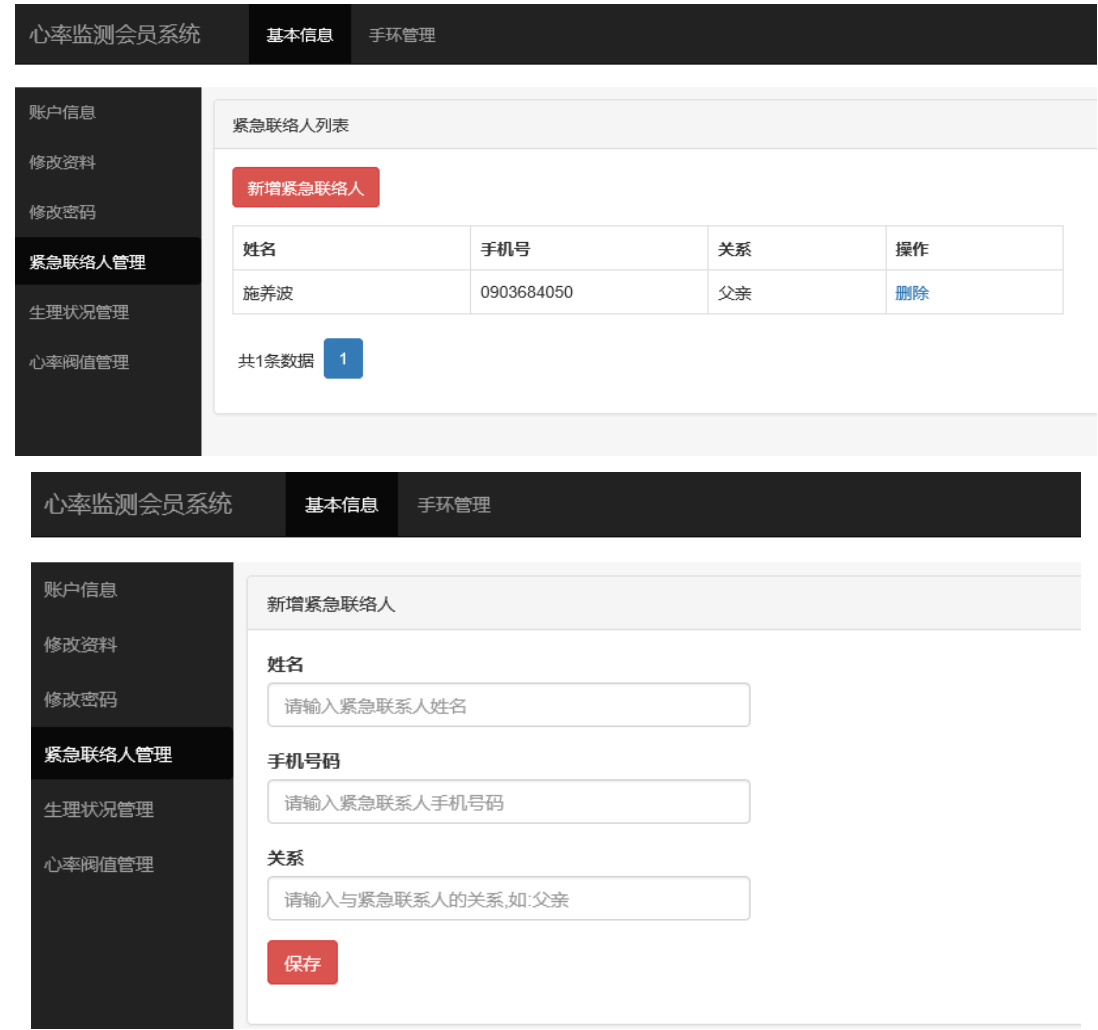


图 6.6 紧急联络人界面

6.7 生理状况界面

- 功能：查看现病史、既往病史
- 主要数据项：现病史、既往病史
- 实测截图

心率监测会员系统

基本信息

手环管理

账户信息

修改资料

修改密码

紧急联络人管理

生理状况管理

心率阈值管理

生理状况预测

生理建议

您最近的平均心率大于80bpm,您可以考虑进行治疗

现病史

新增现病史

病史名称	增加时间	操作
冠心病	2016-11-09 10:17:36	删除

既往病史

新增既往病史

病史名称	增加时间	操作
冠心病	2016-11-09 10:25:55	删除

图 6.7 生理状况界面

## 6.8 心率阈值界面

- 功能：更改心率阈值（警戒值）
- 主要数据项：最高、最低阈值
- 实测截图

心率监测会员系统

基本信息

手环管理

账户信息

修改资料

修改密码

紧急联络人管理

生理状况管理

心率阈值管理

心率阈值管理

最高阈值

120

最低阈值

40

保存

图 6.8 心率阈值管理界面

## 6.9 实时心率界面

- 功能：查看实时心率与相关建议
- 主要数据项：实时心率、生理建议
- 实测截图

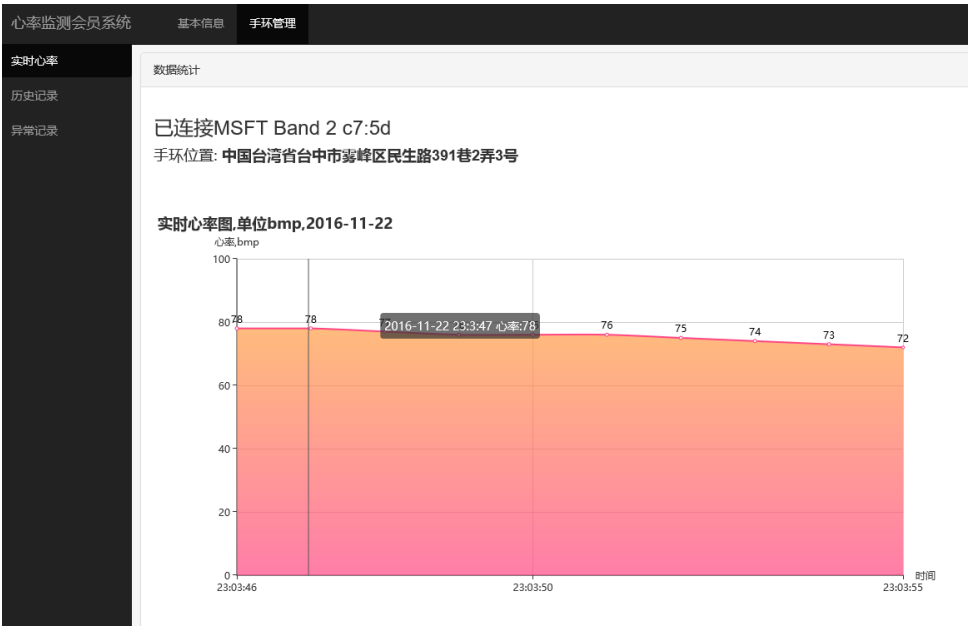


图 6.9 实时心率界面

6.10 历史记录界面

- 功能：可选择查看某一时段的历史心率
- 主要数据项：历史心率
- 实测截图

(1) 历史记录界面

选择查询时间

☐ 按周查询 ☐ 按天查询 ☒ 按时查询

本周 [选择时间] [日历图标] [选择时间] [日历图标]

查询

图 6.10 历史纪录界面

## (2) 按周查询

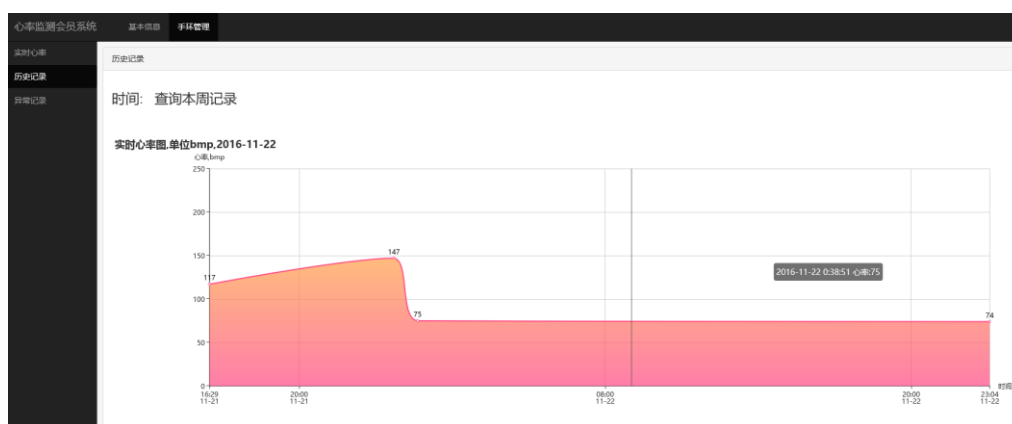


图 6.11 按周查询

## (3) 按天查询

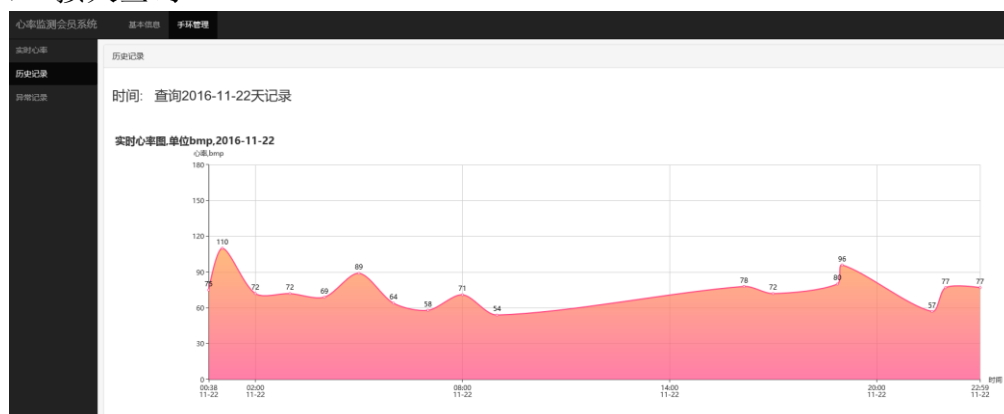


图 6.12 按天查询

## (4) 按时查询

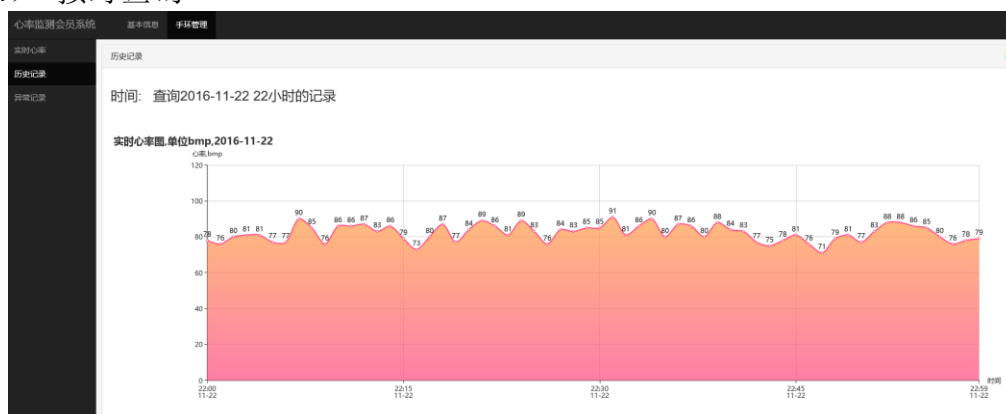


图 6.13 按时查询

## 6.11 异常记录界面

- 功能：可选择查看最近一周内的异常记录（超过或低于心率阈值）
- 主要数据项：异常心率记录
- 实测截图

### （1）异常查询记录界面

心率监测会员系统    基本信息    手环管理

实时心率  
历史记录  
异常记录

异常记录

选择查询时间

☐ 30分钟 ☐ 1天内 ☒ 1周内

查询

图 6.14 异常记录界面

### （2）按 30 分钟查询

心率监测会员系统    基本信息    手环管理		
实时心率 历史记录 异常记录	异常记录	
	时间: 30分钟内	
	编号	异常心率    异常时间
	没有查询到数据	

图 6.15 按 30 分钟查询

### （3）按一天内查询

心率监测会员系统

基本信息

手环管理

实时心率

历史记录

异常记录

异常记录

时间: 1天内

编号	异常心率	异常时间
3101	93	2016-11-22 18:58:13
3100	94	2016-11-22 18:58:12
3099	92	2016-11-22 18:58:11
3098	92	2016-11-22 18:58:04
3097	92	2016-11-22 18:58:03
3096	92	2016-11-22 18:58:02
3095	92	2016-11-22 18:58:01
3094	92	2016-11-22 18:58:00
3093	92	2016-11-22 18:57:59
3092	92	2016-11-22 18:57:58
3091	92	2016-11-22 18:57:57
3090	92	2016-11-22 18:57:56
3089	92	2016-11-22 18:57:55

1

图 6.16 按一天内查询

#### (5) 按一周内查询

心率监测会员系统

实时心率

历史记录

异常记录

基本信息

手环管理

异常记录

时间: 1周内

编号	异常心率	异常时间
3101	93	2016-11-22 18:58:13
3100	94	2016-11-22 18:58:12
3099	92	2016-11-22 18:58:11
3098	92	2016-11-22 18:58:04
3097	92	2016-11-22 18:58:03
3096	92	2016-11-22 18:58:02
3095	92	2016-11-22 18:58:01
3094	92	2016-11-22 18:58:00
3093	92	2016-11-22 18:57:59
3092	92	2016-11-22 18:57:58
3091	92	2016-11-22 18:57:57
3090	92	2016-11-22 18:57:56
3089	92	2016-11-22 18:57:55
3088	159	2016-11-21 22:51:49
3087	165	2016-11-21 22:51:48

1

2

3

4

5

6

7

8

9

10

11

>>

82

图 6.17 按一周内查询

## 6.12 管理员修改密码界面

- 功能：修改当前密码
- 主要数据项：当前密码、新密码
- 实测截图



微软心率监测系统

基本信息

会员管理

账户管理

修改密码

原密码

请输入原密码

新密码 (密码必须以字母开头,长度6-20位)

请输入新密码

重复新密码

请重复输入新密码

保存

图 6.18 管理员修改密码

6.13 管理会员界面

- 功能：查看或编辑所有用户的各项资料
- 主要数据项：用户各项资料
- 实测截图

微软心率监测系统

基本信息

会员管理

账户管理

修改密码

序号	用户名	姓名	手机	性别	年龄	imei	操作
12	tingyu		0988007521	男	0	353187060892120	实时心率 设置心率阈值 编辑 历史记录 异常记录
11	huaxiaapp2		18523787221	男	0		实时心率 设置心率阈值 编辑 历史记录 异常记录
10	huaxiaapp1		18523787221	男	0		实时心率 设置心率阈值 编辑 历史记录 异常记录
3	qqq123		0903663430	男	25	359415050440442	实时心率 设置心率阈值 编辑 历史记录 异常记录
2	chenbin		18523787221	男	0	865800028871918	实时心率 设置心率阈值 编辑 历史记录 异常记录
1	huaxiaapp	朱元璋	13618392490	男	25	865800028871918	实时心率 设置心率阈值 编辑 历史记录 异常记录

1

图 6.19 管理会员管理界面

第七章 结论

专题完成的功能为 Web 端登录注册会员并填写会员信息，并可在紧急联系人界面添加联系人以便后续检测出异常心率通知紧急联系人，在生理状况界面可查看现病史与过往病史，并给出生理预测建议，在心率阈值界面可以更改心率阈值（警戒值），设置最高阈值与最低阈值并在超过与低于阈值时会给出警告提醒。

佩戴手环可实时抓取获取心率，记录历史心率记录与某一段时间内的异常记录，并可向紧急联系人发送警报通知。在 App 上可实现会员的登录，并直接查看实时心率。在项目选题之初没有明确的开发方向，试想是用心率与步数传感器开发一个健身减肥相关项目。后来在跟老师协商，得知朝阳大学内的银发园有老人看护项目，我们才明确方向，选择了心率监测系统这一专题。专题完成的过程我们小组成员分区块讨论后综合完成，过程中遭遇到数据库无法连接，在 App 开发阶段的 UI 界面设计的分歧，App 开发阶段程序出现无法运行的一系列问题，微软给的传感器 API 接口教程和范例都是英文，英文程度不佳的我们起初开发 App 遇到的许多困难。后来在导师和助教学长的指导下，再加之查找了翻译一些翻译教材，才顺利完成 App 开发。后来我们设想的短信发送预警信息，所以我们找了许多大陆的短信接口供应商，但是基本上都不支持台湾地区的号码段，最后求教学长，才找到台湾本地的短信供应商，顺利完成短信验证码和预警短信发送。在完成专题的过程中，我们一步步掌握了 Andriod App 程序开发的方法，应用 Php 设计网页，数据库的搭建与传输等。对于以后完成类似的专题可采用分组讨论并综合完成的方式，遇到问题可多上网搜寻资料，或从书上查找资料解决问题。然而目前我们的软件也还存在不足的地方比如心率预测功能还不够完善，规则较少，心率数据传输的延时还有待降低，手机和手环连接必须保持，手机还必须持续连接 Internet，稳定性不够，使用流程比较繁琐等等。在未来的时间我们将进一步完善我们的软件，进一步完善实时心率的传输延迟与心率获取数据处理并且提高 App 稳定性。使用更多的手环传感器接口获取更多生理数据进行综合分析。在现有心率预测规则基础上，收集学术资料结合人工智能技术和大数据分析，将预测心血管疾病的功能更加完善，或者加入现病史和既往病史的综合分析，帮助使用者能够实际准确地了解自己的身体状况。

## 参考文献

- [1] 《Android App 程序设计教本之无痛起步》施威铭 编著 -----旗標出版股份有限公司

- [2] 《数据库系统原理》 苗雪兰 等编著 -----机械工业出版社
- [3] 《PHP 与 MySQL 5 程序设计》 (W. Jason Gilmore、 朱涛江)
- [4] 《开发高质量 PHP 框架与应用的实际案例解析》 (Sebastian Bergmann Stefan Pribsch , 译者:刘文瀚 刘海燕 )
- [5] Dae-Geun Jang, Sangjun Park, and Minsoo Hahn A Real-Time Pulse Peak Detection Algorithm for the Photoplethysmogram
- [6] Development of a Signal Processing Library for Extraction of SpO<sub>2</sub> , HR, HRV, and RR from Photoplethysmographic Waveforms
- [7]Microsoft band SDK 开发说明书  
<https://developer.microsoftband.com/Content/docs/Microsoft%20Band%20SDK.pdf>
- [8] S. K. Mitra, Digital Signal Processing: A Computer-Based Approach, 3rd ed., New York, USA: McGraw-Hill
- [9] Y. K. Qawqzeh, M. B. I. Reaz, M. A. M. Ali The analysis of PPG contour in the assessment of atherosclerosis for erectile dysfunction subjects
- [10] Telecommunications Institute, Military University of Technology
- [11] Kaliskiego 2, 00-908 Warsaw, Poland ISSN 2079-9292 Wearable
- [12] Photoplethysmographic Sensors—Past and Present
- [13] MIT—BIH 心率失常数据库的识读及应用, 宋喜国, 中国医学物理学杂志, 2004 年 7 月第 21 卷第 4 期
- [14] 刘晓芳, 叶志前. 心率变异性的分析方法和应用. 国外医学生物医学工程分册, 2001,24(1): 42-48.
- [14] 王步青, 王卫东. 心率变异性分析方法的研究进展. 北京生物医学工程, 2007,26(5): 551-554.

## 致 谢

临近毕业, 非常感谢朝阳科技大学老师的栽培, 在专题制作的过程中, 谢谢

老师的悉心教导，在以后的工作和生活中，学生一定会铭记老师的辛勤教导。也感谢专题制作期间在我身边的每一个同学，在学习和生活中我们一起成长，携手走过了这段珍贵的大学时光。

## 附录

注册相关功能函数

```
/**  
 * 注册页面  
 */  
public function reg(){
```

```

        $this->meta_title = '会员注册';
        $this->display();
    }
    /**
     * 发送验证码
     */
    public function c_sendCode($phone = ""){
        if($phone == ""){
            $this->ajaxReturn(ajax_data(ERROR,'手机号码必须填写'));
            exit;
        }
        //发送验证码
        $code = M('code');
        $data['targetPhone'] = $phone;
        $now = time();
        $data['sendDate'] = date("Y-m-d H:i:s",$now);
        //有效期 20 分钟
        $data['validDate'] = date("Y-m-d H:i:s",strtotime($data['sendDate']."+20 minute"));
        $data['statu'] = 1;//发送短信
        $data['code'] = randCode(6,"NUMBER");
        if($this->sendTWSms('【Microsoft band2 心率监测】您注册的验证码为'.$data['code'].'有效期 20 分钟',$phone) !== true){
            $this->ajaxReturn(ajax_data(ERROR,'短信发送失败,请稍后重试'));
            exit;
        }
        $re = $code->data($data)->add();
        if($re === false || $re <= 0){
            $this->ajaxReturn(ajax_data(ERROR,'数据库错误,请稍后重试或者联系管理员'));
        }else{
            $ret = array();
            $ret['id'] = $re;
            $ret['statu'] = $data['statu'];
            // $ret['code'] = $data['code'];
            $ret['sendDate'] = $data['sendDate'];
            $ret['validDate'] = $data['validDate'];
            $this->ajaxReturn(ajax_data(SUCCESS,'发送成功',$ret));
        }
    }
    private function sendTWSms($msg = "", $phone = ""){
        $url
        =
        "http://sms-get.com/api_send.php?username=qq2211888&password=5201314&method=1&sms_msg=".
        urlencode($msg).'&phone='.$phone;
        $html = file_get_contents($url);
        if(strpos($html,"000") === false){

```

```

        return $html;
    }else{
        return true;
    }
}
/**
 * 注册实现
 */
public function c_reg($code_id = 0,$code = ""){
    if($code_id <= 0 || $code == ""){
        $this->error('参数错误');
    }
    $user = D('Reg');
    //先验证参数
    if($user->create()){
        //判断验证码
        $c = M('code');
        $re = $c->where(array('id'=>$code_id))->find();
        if($re !== false && $re != null){
            if($re['validstate'] == 0 && $re['code'] == $code && $re['validdate'] >= date("Y-m-d
H:i:s")){//验证成功
                //注册成功,保存数据
                //设置验证码已经验证过
                $c->where(array('id'=>$code_id))->setField('validstate',1);
                $re = $user->add();
                if($re !== false && $re > 0){
                    $this->success("注册成功,请登录",'loginPage');
                }else{
                    $this->error('数据库错误,请稍后重试');
                }
            }else{
                if($re['validstate'] != 0){
                    $this->error('手机验证码已经验证过了,请重新获取验证码');
                }else{
                    $this->error('手机验证码错误或者已经无效,请重新获取验证码');
                }
            }
        }else{
            $this->error('数据库错误,注册失败,请稍后重试');
        }
    }else{
        $this->error('参数错误,请重试',$user->getError());
    }
}

```

```

}
public function forgetVerifyCode(){
    ob_clean();
    $Verify = new \Think\Verify();
    $Verify->fontSize = 40;
    $Verify->length = 4;
    $Verify->useNoise = true;
    $Verify->entry(FORGET_VERIFY_ID);
}
public function forgetPasswordPage(){
    $this->meta_title = "找回管理员密码";
    $this->display();
}
/**
 * 通过后台更新密码
 * @param string $oldPassword 旧密码
 * @param string $password 新密码
 * @param string $rePassword 重复新密码
 */
public function c_rePassword($oldPassword = "", $password = "", $rePassword = "){

    if(!is_allAdminLogin()){
        $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    if($oldPassword == "" || $password == "" || $rePassword == "){
        $this->error("旧密码,新密码,重复新密码不能为空");
        exit;
    }
    //判断新密码是否和旧密码一致
    if($oldPassword == $password){
        $this->error("新密码不能和旧密码相同");
        exit;
    }
    //验证密码
    if(!preg_match('/^[a-zA-Z][\w~!@#%&*&]{5,19}$/', $password)){
        $this->error("新密码必须以字母开头,长度 6-20 位,并且不能有空白字符");
        exit;
    }
    //先判断 rePassword 和 password 是否一致
    if($password != $rePassword){
        $this->error("新密码和重复新密码不一致");
        exit;
    }
}

```

```

//先判断旧密码
$user = session(SESSION_ADMIN);
$id = $user['id'];
//查询旧密码
$admin = M('admin');
$re = $admin->where(array('id'=>$id,'password'=>md5($oldPassword)))->find();
if($re == null || $re == false){
    $this->error("旧密码错误,请重试");
    exit;
}
//验证成功,那么重置密码
$re = $admin->where(array('id'=>$id))->data(array('password'=>md5($password)))->save();
if($re){//更新成功
    //注销
    session(SESSION_ADMIN,null);
    $this->success("更新密码成功,请重新登录",u(LOGIN_PAGE));
    exit;
}else{
    $this->error("数据库错误,请稍后重试");
}
}

```

#### WEB 相关功能函数

```

<?php
/**
 * 网站信息
 */
namespace Admin\Controller;
use Think\Controller;
use DateTime;
use Think\Page;
class IndexController extends Controller {
    public function index(){
        if(!is_AdminLogin()){
            $this->redirect(LOGIN_PAGE);
            //$this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
            exit;
        }
        //跳转到账户信息页
        $this->redirect('user');
    }
    public function user($p = 1){
        if(!is_AdminLogin()){
            $this->redirect(LOGIN_PAGE);
            //$this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
            exit;
        }
    }
}

```



```

    }
    //查询用户信息
    $user = M('user');
    $count = $user->count();
    $page = new Page($count,10);
    $list = $user->page($p.','.$page->listRows)->order('regtime desc')->select();
    if($list === false || $list == null){
        $this->error('没有查询到该账户信息,可能已经被删除');
        exit;
    }
    //替换数据
    for($i = 0;$i < count($list);$i++){
        if($list[$i]['sex'] == 1){
            $list[$i]['sex'] = '男';
        }else{
            $list[$i]['sex'] = '女';
        }
        $list[$i]['birthday'] = date('Y 年 m 月 d 日',strtotime($list[$i]['birthday']));
    }
    $this->list = $list;
    $this->page = $page->bootshow();
    $this->meta_title = '用户管理';
    $this->display();
}

public function editUser($id = 0){
    if(!is_AdminLogin()){
        $this->redirect(LOGIN_PAGE);
        //$this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    if($id == "" || $id == 0){
        $this->error('用户不存在或者已经删除');
    }
    //查询用户信息
    $user = M('user');
    $data = $user->where(array('id'=>$id))->find();
    if($data === false || $data == null){
        $this->error('没有查询到该账户信息,可能已经被删除');
        exit;
    }
    $this->data = $data;
    $this->meta_title = '修改用户资料';
    $this->display();
}

```

```

public function c_editUser(){
    if(!is_AdminLogin()){
        $this->redirect(LOGIN_PAGE);
        //$this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    $account = D('Account');
    if($account->create()){
        //保存数据
        $re = $account->save();
        if($re === false || $re <= 0){
            $this->error('保存用户资料失败,数据库错误或者没有更改资料');
        }else{
            $this->success('保存用户资料成功','user');
        }
    }else{
        $this->error('修改数据失败,原因如下:'.$account->getLastSql());
    }
}
/**
 * 设置心率阈值页面
 */
public function setThreshold($id = 0){
    if(!is_adminLogin()){
        $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    $user = M('user');
    $this->data = $user->where(array('id'=>$id))->field('max_heart,min_heart')->find();
    $this->meta_title = '设置心率阈值';
    $this->id = $id;
    $this->display();
}
/**
 * 设置心率阈值
 */
public function c_setThreshold($id = 0,$min_heart = 0,$max_heart = 0){
    if($max_heart < 0 || $min_heart < 0){
        $this->error('最高阈值和最低阈值必须为正整数');
    }
    $user = M('user');
    $re = $user->where(array('id'=>$id))
    ->data(array('max_heart'=>$max_heart,'min_heart'=>$min_heart))->save();
    if($re !== false){

```

```

        $this->success('保存成功,请刷新实时心率页面以生效');
    }else{
        $this->error('保存失败,数据库错误');
    }
}

public function account(){
    if(!is_adminLogin()){
        $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    $this->meta_title = '账户管理';
    $this->display();
}

/**
 * 生理建议
 */
public function mental(){
    if(!is_adminLogin()){
        $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    $this->meta_title = '生理建议';
    $this->data = getMental();
    $this->display();
}

/**
 * 修改生理建议
 */
public function c_mental($mental = ""){
    if(!is_adminLogin()){
        $this->ajaxReturn(ajax_data(ERROR,'您还没有登录'));
        exit;
    }
    $this->meta_title = '生理建议';
    if(putMental($mental) > 0)
        $this->success('保存成功');
    else
        $this->error('保存失败,请稍后重试');
}

public function realHeart($id= 0){
    if($id <= 0){
        $this->error('参数错误');
        exit;
    }
}

```

```

if(!is_adminLogin()){
    $this->ajaxReturn(ajax_data(ERROR,'您还没有登录'));
    exit;
}
//以一个时间点开始查询
$start = date('Y-m-d H:i:s',strtotime("-5 seconds"));
$heart = M('heartrate');
$re = $heart->where(array('uid'=>$id,'addTime'=>array('EGT',$start)))->find();
//查询第一条数据
if($re !== false && $re != null){
    $retes = json_decode($re['rates'],true);
    $re['rates'] = $retes;
    $this->heart = json_encode($re);
    $this->id = $re['id'];
}else{
}
$this->start = $start;
//查询生理建议和阈值
$u = M('user');
$this->info = $u->where(array('id'=>$id))->field('mental,max_heart,min_heart')->find();
$this->meta_title = '实时心率';
$this->uid = $id;
$this->name = $u->where(array('id'=>$id))->getField('uname');
$this->display();
}
/**
 * 获取心率数据,根据最后一次提取的时间
 */
public function c_getHeart($uid = 0,$id = 0){
    if(!is_adminLogin()){
        $this->ajaxReturn(ajax_data(ERROR,'您还没有登录'));
        exit;
    }
    if($id <= 0 || $uid <= 0){
        $this->ajaxReturn(ajax_data(ERROR,"参数错误"));
        exit;
    }
    //只显示当期时间
    //以一个时间点开始查询
    $start = date('Y-m-d H:i:s',strtotime("-5 seconds"));
    $heart = M('heartrate');
    $re = $heart->where(array('uid'=>$uid,'addTime'=>array('EGT',$start)))->find();
    //查询第一条数据
    if($re !== false && $re != null){

```

```

        $retes = json_decode($re['rates'],true);
        $re['rates'] = $retes;
        $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
    }else{
        $this->ajaxReturn(ajax_data(ERROR,'查询失败',$re));
    }
    exit;
    $heart = M('heartrate');
    if($id == -1){
        //根据当前时间查询
        //以一个时间点开始查询
        $start = date('Y-m-d H:i:s',strtotime("-3 seconds"));
        $heart = M('heartrate');
        $re = $heart->where(array('uid'=>$uid,'addTime'=>array('EGT',$start)))->find();
        //查询第一条数据
        if($re !== false && $re != null){
            $retes = json_decode($re['rates'],true);
            $re['rates'] = $retes;
            $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
        }else{
            $this->ajaxReturn(ajax_data(ERROR,'查询失败',$re));
        }
        exit;
    }
    //直接查询 id 下的
    $re = $heart->where(array('uid'=>$uid,'id'=>array('GT',$id)))->order('id asc')->find();
    //判断时间间隔
    if($re !== false && $re != null){
        //time() - $re['h_timestemp']-5) > 10
        if((time() - $re['h_timestemp']-5) >= 10){//如果差距太大,修正一下数据
            $start = date('Y-m-d H:i:s',strtotime("-3 seconds"));
            $re = $heart->where(array('uid'=>$uid,'addTime'=>array('GT',$start)))->find();
            if($re !== false && $re != null){
                $retes = json_decode($re['rates'],true);
                $re['rates'] = $retes;
                $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
            }else{
                $this->ajaxReturn(ajax_data(ERROR,'查询失败',$re));
            }
        }
        $retes = json_decode($re['rates'],true);
        $re['rates'] = $retes;
        $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
    }else{

```

```

        $this->ajaxReturn(ajax_data(ERROR,'没有更多数据'));
    }
}
/**
 * 历史记录
 */
public function history($time = 1,$sid = 0){
    if(!is_adminLogin()){
        $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    $this->id = $sid;
    $this->meta_title = '历史记录';
    $this->display();
}
public function c_history($sid = 0,$type = ""){
    if(!is_adminLogin()){
        $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    $heart = M('heartrate');
    $user = session(SESSION_USER);
    switch ($type){
        case 'week'://周查询
            $time = I('time_week');
            //本周
            $heartRate = M('heartrate');
            $list = array();
            if($time == 0){//本周
                //获取本周每一天的日期
                $date=new DateTime();
                $date->modify('this week');
                for ($i = 0;$i < 7;$i++){
                    //计算每天第一条和结束的一条数据
                    $start = $date->format('Y-m-d 00:00:00');
                    $end = $date->format('Y-m-d 23:59:59');
                    //查询数据
                    $count = $heartRate->
                        where(array('uid'=>$sid,
                            'commit_timestemp'=>array('BETWEEN',array($start,$end))))->count();
                    //当天数量
                    if($count >= 2){//查询第一条和最后一条
                        $st = $heartRate->
                            where(array('uid'=>$sid,

```

```

        'commit_timestamp'=>array('BETWEEN',array($start,$end)))
        ->order('commit_timestamp asc')->find();
    $en = $heartRate->
    where(array('uid'=>$id,
        'commit_timestamp'=>array('BETWEEN',array($start,$end)))
        ->order('commit_timestamp desc')->find();
    $list[] = $st;
    $list[] = $en;
} else {
    //查询第一条
    $st = $heartRate->
    where(array('uid'=>$id,
        'commit_timestamp'=>array('BETWEEN',array($start,$end)))
        ->order('commit_timestamp asc')->find();
    if($st !== false && $st !== null){
        $list[] = $st;
    }
}
$date->modify('+1 day');
}
for($i = 0;$i < count($list);$i++){
    $list[$i]['rates'] = json_decode($list[$i]['rates'],true);
    $list[$i]['rates'] = array($list[$i]['rates'][0]);
}
} else { //上周
    $date=new DateTime();
    $date->modify('this week -7 day');
    for ($i = 0;$i < 7;$i++){
        //计算每天第一条和结束的一条数据
        $start = $date->format('Y-m-d 00:00:00');
        $end = $date->format('Y-m-d 23:59:59');
        //查询数据
        $count = $heartRate->
        where(array('uid'=>$id,
            'commit_timestamp'=>array('BETWEEN',array($start,$end)))>count();
        //当天数量
        if($count >= 2){ //查询第一条和最后一条
            $st = $heartRate->
            where(array('uid'=>$id,
                'commit_timestamp'=>array('BETWEEN',array($start,$end)))
                ->order('commit_timestamp asc')->find();
            $en = $heartRate->
            where(array('uid'=>$id,
                'commit_timestamp'=>array('BETWEEN',array($start,$end)))

```

```

        ->order('commit_timestamp desc')->find();
        $list[] = $st;
        $list[] = $en;
    }else{
        //查询第一条
        $st = $heartRate->
        where(array('uid'=>$id,
            'commit_timestamp'=>array('BETWEEN',array($start,$end))))
        ->order('commit_timestamp asc')->find();
        if($st !== false && $st !== null){
            $list[] = $st;
        }
    }
    $date->modify('+1 day');
}
for($i = 0;$i < count($list);$i++){
    $list[$i]['rates'] = json_decode($list[$i]['rates'],true);
    $list[$i]['rates'] = array($list[$i]['rates'][0]);
}
}
$this->hearts = json_encode($list);
$this->pro = '查询'.($time == 0?'本周记录':'上周记录');
$this->display();
break;
case 'day':
    $time = I('time_day');
    //查询每小时第一条数据
    $list = array();
    $heartRate = M('heartrate');
    for ($i = 0;$i < 24;$i++){
        if($i < 10){
            $start = $time.'0'.$i.':00:00';
            $end = $time.'0'.$i.':59:59';
        }else{
            $start = $time.' '.$i.':00:00';
            $end = $time.' '.$i.':59:59';
        }
        //echo $start.'---'.$end.'<br/>';
        //过滤合法的数据
        $hearts = $heartRate->
        where(array('uid'=>$id,
            'commit_timestamp'=>array('BETWEEN',array($start,$end))))->find();
        if($hearts !== false && $hearts !== null){
            $hearts['rates'] = json_decode($hearts['rates'],true);

```



```

        for($k = 0;$k < count($hearts['rates']);$k++){
            $hearts['rates'] = array($hearts['rates'][$k]);
            $list[] = $hearts;
            break;
        }
    }else{
    }
}
$this->hearts = json_encode($list);
$this->pro = '查询'.time.'天记录';
$this->display();
break;
case 'hour'://天查询
    $time = I('time_hour');
    //查询每天每个小时第一条数据
    $fTime = $time.'00:00';
    $fTime = strtotime($fTime);
    $start = date('Y-m-d H:00:00',$fTime);
    $end = date('Y-m-d H:59:59',$fTime);
    //查找这个小时的数据
    $hearts = $heart->
    where(array('uid'=>$id,
        'commit_timestemp'=>array('BETWEEN',array($start,$end))))->select();
    $list = array();
    for($i = 0;$i < count($hearts);$i++){
        $hearts[$i]['rates'] = json_decode($hearts[$i]['rates'],true);
        //判断是否有整点数据
        for($k = 0;$k < count($hearts[$i]['rates']);$k++){
            $heart = $hearts[$i]['rates'][$k];
            $his = date('s',$heart['timestemp']);
            if($his == '00'){//说明是整数
                $hearts[$i]['rates'] = array($heart);
                $list[] = $hearts[$i];
                break;
            }
        }
    }
}
$this->hearts = json_encode($list);
$this->pro = '查询'.time.'小时的记录';
$this->display();
break;
}
exit;
$pro = "

```

```

switch ($time){
    case 1:
        $selTime = strtotime('-10 minute');
        $pro = '10 分钟内';
        break;
    case 2:
        $selTime = strtotime('-20 minute');
        $pro = '20 分钟内';
        break;
    case 3:
        $selTime = strtotime('-30 minute');
        $pro = '30 分钟内';
        break;
    default:
        $pro = '不支持此时间段';
        $this->pro = $pro;
        $this->hearts = "";
        $this->display();
        break;
}
$this->pro = $pro;
//echo ;
// exit;
//判断时间戳
$heart = M('heartrate');
$user = session(SESSION_USER);
$hearts    =    $heart->where(array('uid'=>$user['id'],'commit_timestemp'=>array('EGT',date('Y-m-d
H:i:s',$selTime))))->select();
for($i = 0;$i < count($hearts);$i++){
    $hearts[$i]['rates'] = json_decode($hearts[$i]['rates'],true);
}
$this->hearts = json_encode($hearts);
$this->display();
}
public function warn($id = 0){
    if(!is_adminLogin()){
        $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
        exit;
    }
    $this->id = $id;
    $this->meta_title = '异常记录';
    $this->display();
}
public function c_warn($time = 1,$p = 1,$id = 0){

```

```

if(!is_adminLogin()){
    $this->error(NOT_LOGIN,u(LOGIN_PAGE),NOT_LOGIN_TIME);
    exit;
}
$pro = "";
switch ($time){
    case 1:
        $selTime = strtotime('-30 minute');
        $pro = '30 分钟内';
        break;
    case 2:
        $selTime = strtotime('-1 day');
        $pro = '1 天内';
        break;
    case 3:
        $selTime = strtotime('-7 day');
        $pro = '1 周内';
        break;
    default:
        $pro = '不支持此时间段';
        $this->pro = $pro;
        $this->hearts = "";
        $this->display();
        break;
}
$this->pro = $pro;
//echo ;
// exit;
//判断时间戳
$warn = M('warn');
$count = $warn->where(array('uid'=>$id,'timestemp'=>array('EGT',date('Y-m-d
H:i:s',$selTime))))->count();
$page = new Page($count,15);
$list = $warn->where(array('uid'=>$id,'timestemp'=>array('EGT',date('Y-m-d
H:i:s',$selTime))))->page($p.','.$page->listRows->order('timestemp desc')->select();
$this->list = $list;
$this->page = $page->bootshow();
$this->display();
}
}
?>
<?php
namespace Home\Controller;
use DateTime;

```

```

use Think\Controller;
use Think\Page;
class BandsController extends Controller{

    public function index(){
        if(!is_UserLogin()){
            $this->redirect('User/loginPage');
            exit;
        }
        //以一个时间点开始查询
        $start = date('Y-m-d H:i:s',strtotime("-5 seconds"));
        $heart = M('heartrate');
        $user = session(SESSION_USER);
        $re = $heart->where(array('uid'=>$user['id'],'addTime'=>array('EGT',$start)))->find();
        //查询第一条数据
        if($re !== false && $re != null){
            $retes = json_decode($re['rates'],true);
            $re['rates'] = $retes;
            $this->heart = json_encode($re);
            $this->id = $re['id'];
        }else{
        }
        $this->start = $start;
        //查询生理建议和阈值
        $u = M('user');
        $this->info = $u->where(array('id'=>$user['id']))->field('mental,max_heart,min_heart')->find();
        $this->meta_title = '手环管理';
        $this->display();
    }
}
/**
 * 获取心率数据,根据最后一次提取的时间
 */
public function c_getHeart($id = -1){
    if(!is_UserLogin()){
        $this->ajaxReturn(ajax_data(ERROR,'您还没有登录'));
        exit;
    }
    if($id == ""){
        $this->ajaxReturn(ajax_data(ERROR,"参数错误"));
        exit;
    }
    //只显示当期时间
    //以一个时间点开始查询
    $start = date('Y-m-d H:i:s',strtotime("-5 seconds"));

```

```

$heart = M('heartrate');
$user = session(SESSION_USER);
$re = $heart->where(array('uid'=>$user['id'],'addTime'=>array('EGT',$start)))->find();
//查询第一条数据
if($re !== false && $re !== null){
    $retes = json_decode($re['rates'],true);
    $re['rates'] = $retes;
    $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
}else{
    $this->ajaxReturn(ajax_data(ERROR,'查询失败',$re));
}
exit;
$heart = M('heartrate');
if($id == -1){
    //根据当前时间查询
    //以一个时间点开始查询
    $start = date('Y-m-d H:i:s',strtotime("-3 seconds"));
    $heart = M('heartrate');
    $user = session(SESSION_USER);
    $re = $heart->where(array('uid'=>$user['id'],'addTime'=>array('EGT',$start)))->find();
    //查询第一条数据
    if($re !== false && $re !== null){
        $retes = json_decode($re['rates'],true);
        $re['rates'] = $retes;
        $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
    }else{
        $this->ajaxReturn(ajax_data(ERROR,'查询失败',$re));
    }
}
exit;
}
//直接查询 id 下的
$user = session(SESSION_USER);
$re = $heart->where(array('uid'=>$user['id'],'id'=>array('GT',$id)))->order('id asc')->find();
//判断时间间隔
if($re !== false && $re !== null){
    //time() - $re['h_timestamp']-5) > 10
    if((time() - $re['h_timestamp']-5) >= 10){//如果差距太大,修正一下数据
        $start = date('Y-m-d H:i:s',strtotime("-3 seconds"));
        $user = session(SESSION_USER);
        $re = $heart->where(array('uid'=>$user['id'],'addTime'=>array('GT',$start)))->find();
        if($re !== false && $re !== null){
            $retes = json_decode($re['rates'],true);
            $re['rates'] = $retes;
            $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
        }
    }
}

```

```

        }else{
            $this->ajaxReturn(ajax_data(ERROR,'查询失败',$re));
        }
    }
    $retes = json_decode($re['rates'],true);
    $re['rates'] = $retes;
    $this->ajaxReturn(ajax_data(SUCCESS,'查询成功',$re));
}
}
}
/**
 * 历史记录
 */
public function history($time = 1){

    if(!is_UserLogin()){
        $this->redirect('User/loginPage');
        exit;
    }
    $this->meta_title = '历史记录';
    $this->display();
}

public function c_history($type = ""){
    if(!is_UserLogin()){
        $this->redirect('User/loginPage');
        exit;
    }
    $heart = M('heartrate');
    $user = session(SESSION_USER);
    switch ($type){
        case 'week'://周查询
            $time = I('time_week');
            //本周
            $heartRate = M('heartrate');
            $list = array();
            if($time == 0){//本周
                //获取本周每一天的日期
                $date=new DateTime();
                $date->modify('this week');
                for ($i = 0;$i < 7;$i++){
                    //计算每天第一条和结束的一条数据
                    $start = $date->format('Y-m-d 00:00:00');
                    $end = $date->format('Y-m-d 23:59:59');
                }
            }
        }
    }

```

```

//查询数据
$count = $heartRate->
where(array('uid'=>$user['id'],
    'commit_timestamp'=>array('BETWEEN',array($start,$end))))->count();
//当天数量
if($count >= 2){//查询第一条和最后一条
    $st = $heartRate->
    where(array('uid'=>$user['id'],
        'commit_timestamp'=>array('BETWEEN',array($start,$end)))
        ->order('commit_timestamp asc')->find();
    $en = $heartRate->
    where(array('uid'=>$user['id'],
        'commit_timestamp'=>array('BETWEEN',array($start,$end)))
        ->order('commit_timestamp desc')->find();
    $list[] = $st;
    $list[] = $en;
}else{
    //查询第一条
    $st = $heartRate->
    where(array('uid'=>$user['id'],
        'commit_timestamp'=>array('BETWEEN',array($start,$end)))
        ->order('commit_timestamp asc')->find();
    if($st !== false && $st !== null){
        $list[] = $st;
    }
}
$date->modify('+1 day');
}
for($i = 0;$i < count($list);$i++){
    $list[$i]['rates'] = json_decode($list[$i]['rates'],true);
    $list[$i]['rates'] = array($list[$i]['rates'][0]);
}
}else{//上周
    $date=new DateTime();
    $date->modify('this week -7 day');
    for ($i = 0;$i < 7;$i++){
        //计算每天第一条和结束的一条数据
        $start = $date->format('Y-m-d 00:00:00');
        $end = $date->format('Y-m-d 23:59:59');
        //查询数据
        $count = $heartRate->
        where(array('uid'=>$user['id'],
            'commit_timestamp'=>array('BETWEEN',array($start,$end))))->count();
        //当天数量

```

```

        if($count >= 2){//查询第一条和最后一条
            $st = $heartRate->
                where(array('uid'=>$user['id'],
                    'commit_timestamp'=>array('BETWEEN',array($start,$end))))
                ->order('commit_timestamp asc')->find();
            $en = $heartRate->
                where(array('uid'=>$user['id'],
                    'commit_timestamp'=>array('BETWEEN',array($start,$end))))
                ->order('commit_timestamp desc')->find();
            $list[] = $st;
            $list[] = $en;
        }else{
            //查询第一条
            $st = $heartRate->
                where(array('uid'=>$user['id'],
                    'commit_timestamp'=>array('BETWEEN',array($start,$end))))
                ->order('commit_timestamp asc')->find();
            if($st !== false && $st !== null){
                $list[] = $st;
            }
        }
        $date->modify('+1 day');
    }
    for($i = 0;$i < count($list);$i++){
        $list[$i]['rates'] = json_decode($list[$i]['rates'],true);
        $list[$i]['rates'] = array($list[$i]['rates'][0]);
    }
}
$this->hearts = json_encode($list);
$this->pro = '查询'.($time == 0?'本周记录':'上周记录');
$this->display();
break;
case 'day':
    $time = I('time_day');
    //查询每小时第一条数据
    $list = array();
    $heartRate = M('heartrate');
    for ($i = 0;$i < 24;$i++){
        if($i < 10){
            $start = $time.' 0'.$i.':00:00';
            $end = $time.' 0'.$i.':59:59';
        }else{
            $start = $time.' '.$i.':00:00';
            $end = $time.' '.$i.':59:59';
        }
    }

```



```

    }
    //echo $start.'---'.Send.'<br/>';
    //过滤合法的数据
    $hearts = $heartRate->
    where(array('uid'=>$user['id'],
        'commit_timestemp'=>array('BETWEEN',array($start,$end))))->find();
    if($hearts !== false && $hearts != null){
        $hearts['rates'] = json_decode($hearts['rates'],true);
        for($k = 0;$k < count($hearts['rates']);$k++){
            $hearts['rates'] = array($hearts['rates'][$k]);
            $list[] = $hearts;
            break;
        }
    }else{
    }
}
$this->hearts = json_encode($list);
$this->pro = '查询'.time.'天记录';
$this->display();
break;
case 'hour'://天查询
    $time = I('time_hour');
    //查询每天每个小时第一条数据
    $fTime = $time.'00:00';
    $fTime = strtotime($fTime);
    $start = date('Y-m-d H:00:00',$fTime);
    $end = date('Y-m-d H:59:59',$fTime);
    //查找这个小时的数据
    $hearts = $heart->
        where(array('uid'=>$user['id'],
            'commit_timestemp'=>array('BETWEEN',array($start,$end))))->select();
    $list = array();
    for($i = 0;$i < count($hearts);$i++){
        $hearts[$i]['rates'] = json_decode($hearts[$i]['rates'],true);
        //判断是否有整点数据
        for($k = 0;$k < count($hearts[$i]['rates']);$k++){
            $heart = $hearts[$i]['rates'][$k];
            $his = date('s',$heart['timestemp']);
            if($his == '00'){//说明是整数
                $hearts[$i]['rates'] = array($heart);
                $list[] = $hearts[$i];
                break;
            }
        }
    }
}

```

```

    }
    $this->hearts = json_encode($list);
    $this->pro = '查询'. $time. '小时的记录';
    $this->display();
    break;
}
exit;
$pro = "";
switch ($time){
    case 1:
        $selTime = strtotime('-10 minute');
        $pro = '10 分钟内';
        break;
    case 2:
        $selTime = strtotime('-20 minute');
        $pro = '20 分钟内';
        break;
    case 3:
        $selTime = strtotime('-30 minute');
        $pro = '30 分钟内';
        break;
    default:
        $pro = '不支持此时间段';
        $this->pro = $pro;
        $this->hearts = "";
        $this->display();
        break;
}
$this->pro = $pro;
//echo ;
// exit;
//判断时间戳
$heart = M('heartrate');
$user = session(SESSION_USER);

$hearts = $heart->where(array('uid'=>$user['id'],'commit_timestemp'=>array('EGT',date('Y-m-d
H:i:s',$selTime))))->select();
for($i = 0;$i < count($hearts);$i++){
    $hearts[$i]['rates'] = json_decode($hearts[$i]['rates'],true);
}
$this->hearts = json_encode($hearts);
$this->display();
}
public function warn(){

```

```

        if(!is_UserLogin()){
            $this->redirect('User/loginPage');
            exit;
        }
        $this->meta_title = '异常记录';
        $this->display();
    }

    public function c_warn($time = 1,$p = 1){
        if(!is_UserLogin()){
            $this->redirect('User/loginPage');
            exit;
        }
        $pro = "";
        switch ($time){
            case 1:
                $selTime = strtotime('-30 minute');
                $pro = '30 分钟内';
                break;
            case 2:
                $selTime = strtotime('-1 day');
                $pro = '1 天内';
                break;
            case 3:
                $selTime = strtotime('-7 day');
                $pro = '1 周内';
                break;
            default:
                $pro = '不支持此时间段';
                $this->pro = $pro;
                $this->hearts = "";
                $this->display();
                break;
        }
        $this->pro = $pro;
        //echo ;
        // exit;
        //判断时间戳
        $warn = M('warn');
        $user = session(SESSION_USER);
        $count = $warn->where(array('uid'=>$user['id'],'timestemp'=>array('EGT',date('Y-m-d
H:i:s',$selTime))))->count();
        $page = new Page($count,15);
        $list = $warn->where(array('uid'=>$user['id'],'timestemp'=>array('EGT',date('Y-m-d
H:i:s',$selTime))))->page($p,'',$page->listRows)->order('timestemp desc')->select();

```

```

        $this->list = $list;
        $this->page = $page->bootshow();
        $this->display();
    }
    /**
     * 保存异常数据
     */
    public function c_save_warn(){
        //保存数据
        if(!is_UserLogin()){
            $this->ajaxReturn(ajax_data(ERROR,'您还没有登录'));
            exit;
        }
        //保存数据
        $warn = D('warn');
        if($warn->create()){
            $user = session(SESSION_USER);
            $warn->uid = $user['id'];
            $warn->timestemp = date('Y-m-d H:i:s',$warn->timestemp);
            $warn->add();
            $this->ajaxReturn(ajax_data(SUCCESS,'保存成功'));
        }else{
            $this->ajaxReturn(ajax_data(ERROR,'参数错误'));
        }
    }
}
?>

```

校系名	朝阳科技大学资通系	朝陽科技大學資訊與通訊系
-----	-----------	--------------

專題名稱	心率监测管 理系统
文件編號	

實務專題

專題名稱

CYUT-ICE-2014-B01