健康吧设计规格说明

Ver:1.0

**目 录**

[1 引言 3](#_Toc301301027)

[1.1 目标 3](#_Toc301301028)

[1.2 文档范围 3](#_Toc301301029)

[1.3 术语和缩略语 3](#_Toc301301030)

[1.4 参考资料 3](#_Toc301301031)

[1.5 系统目标和约束 3](#_Toc301301032)

[2 系统设计 4](#_Toc301301033)

[2.1 系统架构概述 4](#_Toc301301034)

[2.2 对象模型 4](#_Toc301301035)

[2.3 接口 4](#_Toc301301036)

[2.4 特性实现 4](#_Toc301301037)

[2.5 错误代码 9](#_Toc301301038)

[2.6 错误日志 9](#_Toc301301039)

[2.7 部署视图 9](#_Toc301301040)

[3 数据库设计 11](#_Toc301301041)

[3.1 逻辑模型 11](#_Toc301301042)

[3.2 物理模型 11](#_Toc301301043)

[4 质量及其他方面 12](#_Toc301301044)

[4.1 可维护性 12](#_Toc301301045)

[4.2 安全性 12](#_Toc301301046)

[4.3 可扩展性 12](#_Toc301301047)

[4.4 可靠性 12](#_Toc301301048)

[4.5 可用性 12](#_Toc301301049)

[4.6 性能设计 13](#_Toc301301050)

[5 附录 13](#_Toc301301051)

[5.1 附件 13](#_Toc301301052)

[5.2 修过记录 13](#_Toc301301053)

# 引言

## 目标

在完成软件开发前期的准备工作如项目需求等，结合《需求确认书》，，项目小组提出了这份软件设计说明书。

此概要设计说明书对北戴河家庭旅馆系统的功能分配，模块划分，程序的总体结构，输入输出和接口设计，运行设计，数据结构设计及出错设计等方面作了全面的概括性的说明，为软件详细设计奠定了基础，同时作为系统分析员工作的阶段性总结和程序员进行开发及未来测试的重要文档资料。

## 文档范围

本文档包括系统的架构设计、数据库设计、安全性、可靠性、可用性等方面的规划和设计。

## 术语和缩略语

1.心率：也叫安静心率，指正常人安静状态下每分钟心跳的次数，一般为60~100次/分。

2.血压：（blood pressure，BP）是指血液在血管内流动时作用于单位面积和血管壁的侧压力，它是推动血液在血管内流动的动力。

3.脉冲：（pulses per second）是指电子技术中经常运用的一种像脉搏似的短暂起伏的电冲击。

4.BMI：身体质量指数，是目前国际上常用的衡量人体胖瘦程度以及是否健康的一个标准。成人标准值是BMI18.5-23。

## 参考资料

部分内容参考了《健康吧软件需求说明书》

## 系统目标和约束

软件目标：项目需要完成登录注册，查找附近医院，测量心率，相关健康内容推荐，用户和专家交流，计算IBM值等内容

软件的约束：因为时间有限，该软件只实现最基本的功能。。

# 系统设计

## 系统架构概述

该软件的架构采用两层架构的模式：展示层（视图+业务逻辑）和数据访问层。采用面向过程的设计方法。

## 对象模型

该软件采用非面向对象的设计方法。

## 接口

数据通信协议：http协议

接口提供：向用户提供查询、编辑和登录注册等命令选择，对应软件的不同功能实现

外部接口：键盘，摄像头  
内部接口：数据库接口采用SQL链接

MySideBar.OnTouchingLetterChangedListener：按字母顺序显示城市名称，根据点击的城市名进行定位

ViewPager.OnPageChangeListener：处理viewPager页面滑动的事件

## 特性实现

1. 用户注册
   1. 用户注册页面
   2. 注册成功，可选择使用软件全部功能



1. 首页-功能页
   1. 轮播图-展示各方面当下流行信息
   2. 功能模块-健身、食谱、养生、疾病、药箱、护眼、心理、保健
   3. 推荐-根据用户相关数据，为用户进行相关推荐



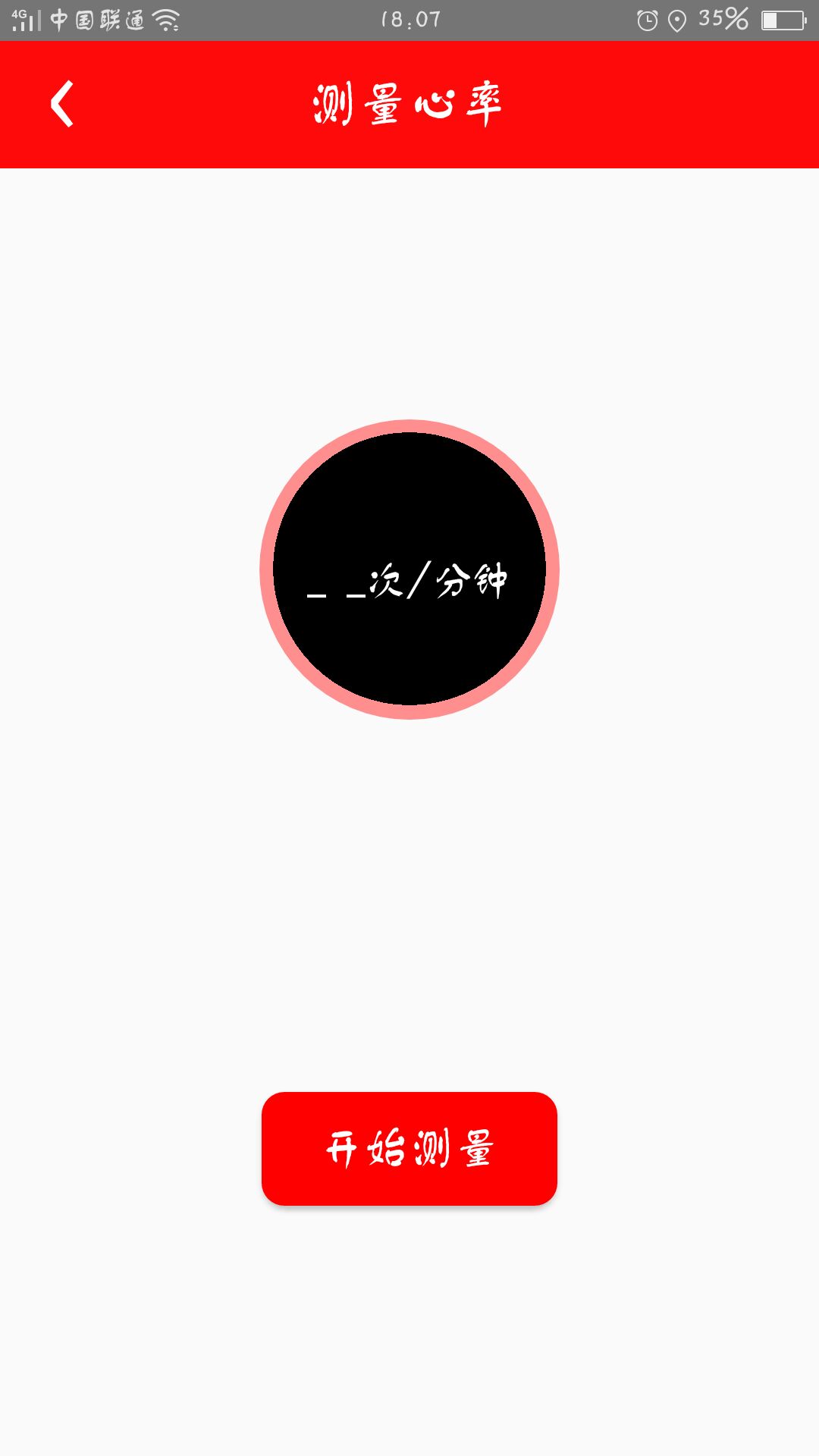
1. 医院
   1. 定位用户当前位置
   2. 根据关键字查找医院或药房，并显示



1. 心率-测量心率



* 1. 手指置于摄像头下，等待测量完成
  2. 保存测量数据，若用户已登录且选择 保存则保存到我的-身体数据，若用户未登录，则提示登录后保存



1. 交流
   1. 发布问题
   2. 回答问题
   3. 评论点赞



1. 我的



* 1. 登录

Ⅰ.普通登录



Ⅱ.快速登录



* 1. 功能栏

Ⅰ.收藏

Ⅱ.菜谱

Ⅲ.药箱

Ⅳ.消息

* 1. 个人信息
  2. 身体数据
  3. 最近阅读
  4. 设置

## 错误代码

列出Java错误处理机制所能处理的所有错误代码及其含义

**抛出异常**：当一个方法出现错误引发异常时，方法创建异常对象并交付运行时系统，异常对象中包含了异常类型和异常出现时的程序状态等异常信息。运行时系统负责寻找处置异常的代码并执行。

捕捉异常：在方法抛出异常之后，运行时系统将转为寻找合适的异常处理器（exception handler）。潜在的异常处理器是异常发生时依次存留在调用栈中的方法的集合。当异常处理器所能处理的异常类型与方法抛出的异常类型相符时，即为合适的异常处理器。运行时系统从发生异常的方法开始，依次回查调用栈中的方法，直至找到含有合适异常处理器的方法并执行。当运行时系统遍历调用栈而未找到合适的异常处理器，则运行时系统终止。同时，意味着Java程序的终止。

抛出异常的方法：throws和throw

throws：通常被用在声明方法时，用来指定方法可能抛出的异常，多个异常可使用逗号分隔。throws关键字将异常抛给上一级，如果不想处理该异常，可以继续向上抛出，但最终要有能够处理该异常的代码。

throw：通常用在方法体中或者用来抛出用户自定义异常，并且抛出一个异常对象。程序在执行到throw语句时立即停止，如果要捕捉throw抛出的异常，则必须使用try-catch语句块或者try-catch-finally语句。

1. throws方法抛出异常

public class Shoot {

static void pop()throws NegativeArraySizeException{

int[] arr = new int[-3];

}

public static void main(String[] args) {

try{

pop();

}catch(NegativeArraySizeException e){

System.out.println("pop()方法抛出的异常");

}

}

1. throw方法抛出异常

public class TestException {

public static void main(String[] args) {

int a = 6;

int b = 0;

try {

if (b == 0) throw new ArithmeticException(); // 通过throw语句抛出异常

System.out.println("a/b的值是：" + a / b);

}

catch (ArithmeticException e) { // catch捕捉异常

System.out.println("程序出现异常，变量b不能为0。");

}

System.out.println("程序正常结束。");

}

}

1. try-catch-finally语句

package Exception;

public class exception4 {

public exception4() {

}

boolean testEx() throws Exception {

boolean ret = true;

try {

ret = testEx1();

} catch (Exception e) {

System.out.println("testEx, catch exception");

ret = false;

throw e;

} finally {

System.out.println("testEx, finally; return value=" + ret);

return ret;

}

}

boolean testEx1() throws Exception {

boolean ret = true;

try {

ret = testEx2();

if (!ret) {

return false;

}

System.out.println("testEx1, at the end of try");

return ret;

} catch (Exception e) {

System.out.println("testEx1, catch exception");

ret = false;

throw e;

} finally {

System.out.println("testEx1, finally; return value=" + ret);

return ret;

}

}

boolean testEx2() throws Exception {

boolean ret = true;

try {

int b = 12;

int c;

for (int i = 2; i >= -2; i--) {

c = b / i;

System.out.println("i=" + i);

}

return true;

} catch (Exception e) {

System.out.println("testEx2, catch exception");

ret = false;

throw e;

} finally {

System.out.println("testEx2, finally; return value=" + ret);

return ret;

}

}

public static void main(String[] args) {

exception4 testException1 = new exception4();

try {

testException1.testEx();

} catch (Exception e) {

e.printStackTrace();

}

}

}

## 错误日志

错误类型：Error、Runtime Exception 运行时异常、Exception、throw 用户自定义异常

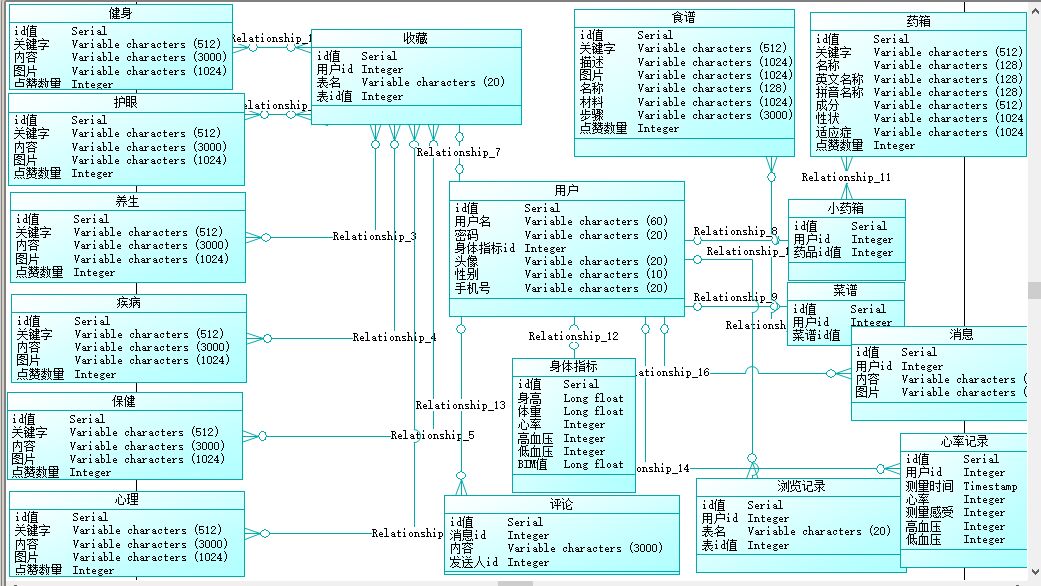
利用java.until.logging工具将错误信息记录到日志中。

## 部署视图

1. 硬件环境：
   1. 互联网、电脑
   2. 至少需要一台服务器
2. 软件：
   1. 需要Windows Server 2003或其他Server版的操作系统以安装SQL SERVER2005企业版。或者其他操作系统，但是使用SQL SERVER就不能使用企业版了，可以换成EXPRESS版。
   2. 需要Apache2.2作为WEB服务器。
   3. 需要安装javascript、jQuery、Ajax、SpringMVC+Hibernate体系。
3. 配置：
   1. 以基于互联网的WEB应用方式提供服务
   2. 前端前端技术需要配置javascript、jQuery、Ajax。
   3. 后端后端技术需要配置SpringMVC+Hibernate体系。
4. 部署：
   1. 配置PHP.ini,关闭错误提示。
   2. 配置PHP连接MS SQL的文件，修改IP地址、数据库的用户名和密码。
   3. 将程序包放在Apache安装目录中得htdocs目录下。

# 数据库设计

## 逻辑模型



## 物理模型

1. 为每个表初始设置300M的存储空间，以20%的大小扩展。
2. 为上传的图片预留空间。

# 质量及其他方面

## 可维护性

1. 通过程序注释等方式增加代码的可读性和可维护性。
2. 将数据访问层分离，做成一个个函数，由其它层调用，以增加代码的可维护性。
3. 数据库有日志记录，系统一旦出现故障有恢复到故障之前的信息和数据的能力

## 安全性

1. 密码使用md5加密
2. 对用户的输入进行验证
3. 对用户的输入的特殊字符进行转义，防止sql注入攻击

## 可扩展性

1. 可以通过增加硬件资源的方式提高系统的响应速度。
2. 可以通过修改代码来添加新功能以及修改完善现有功能。
3. 可以通过软件框架来实现：动态加载的插件、顶端有抽象接口的认真设计的类层次结构、有用的回调函数构造以及功能很有逻辑并且可塑性很强的代码结构。

## 可靠性

1. 数据库按时备份，有日志记录

## 可用性

1. 做好充分的前期工作，系统一旦投入使用，尽量减少宕机的次数和时间
2. 用户不必记忆面向计算机硬件软件的知识。
3. 用户学习操作的时间较短，不必为手的操作分心，操作动作简单重复。
4. 在非正常环境和情景时，用户仍然能够正常进行操作。

## 性能设计

1. 识别约束：预算在10万左右；时间2个月；可选的开发工具以及技术等不约束，框架可自选。
2. 确定功能特性：
3. 可用于大部分成年人的心率测量；
4. 测量数据较为准确，符合实际情况；
5. 身体状况的测量显示需要依从用户输入的身体数据；
6. 精度：

该软件的输入精度：小数点前一位，限制输入特殊字符；  
输出数据的精度：小数点前一位；  
传输过程中的精度：小数点前一位；

1. 时间特性要求：  
   a．响应时间：0.5s；  
   b．更新处理时间：0.5s；  
   c．数据的转换和传送时间：1s；  
   d．等待心率平稳值的时间：8s；
2. 安全性良好。
3. 确定负载：系统的客户端需要用户输入准确的身体数据，正确地进行心率测试来保证系统服务器返回并保存的测量数据的准确性。

# 附录

## 附件

Androidstudio.exe 、 Eclipse.exe 、 Navicat.exe

## 修过记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 | 核准 |
| 1.0 | 董倩 | 2020-4-16 | 传感器测量心率 | 心率数值 |