[1引言 2](#_Toc521465556)

[1.1编写目的 2](#_Toc521465557)

[1.2背景 2](#_Toc521465558)

[1.3定义 2](#_Toc521465559)

[1.4参考资料 2](#_Toc521465560)

[2程序系统的结构 2](#_Toc521465561)

[3程序1（标识符）设计说明 2](#_Toc521465562)

[3.1程序描述 3](#_Toc521465563)

[3.2功能 3](#_Toc521465564)

[3.3性能 3](#_Toc521465565)

[3.4输人项 3](#_Toc521465566)

[3.5输出项 3](#_Toc521465567)

[3.6算法 3](#_Toc521465568)

[3.7流程逻辑 3](#_Toc521465569)

[3.8接口 3](#_Toc521465570)

[3.9存储分配 4](#_Toc521465571)

[3.10注释设计 4](#_Toc521465572)

[3.11限制条件 4](#_Toc521465573)

[3.12测试计划 4](#_Toc521465574)

[3.13尚未解决的问题 4](#_Toc521465575)

[4程序2（标识符）设计说明 4](#_Toc521465576)

**详细设计说明书**

# 1引言

## 1.1编写目的

本详细设计说明书的编写目的是为了说明程序系统的各个层次中的每个软件对象（包括每个模块和程序）的设计和考虑，以向系统实现（编码和测试）阶段提供关于程序系统实现方式的详细描述，从而成为编码的技术基础。

本详细设计说明书的适用读者为：软件开发者、测试人员

## 1.2背景

待开发软件系统名称：Aqua智慧淋浴系统

开发单位：山东电子技术学院Aqua组

主管部门：山东电子技术学院Aqua组

在XXX大学中，学校浴室的管理消耗大量的人力、物力资源，为了解决此问题，我们开发了此智慧淋浴系统。旨在实现校园浴室的智能化管理与使用节省不必要的人力、物力，提高浴室的服务效率和同学们的使用效率，提高同学和管理者的使用体验。

此客户端是基于使用广泛的微信公众号进行开发，可同时在Android、iPhone、WindowsPhone系统上使用。

## 1.3定义

树莓派：它是一款基于ARM的微型电脑主板，又称卡片式电脑，外形只有信用卡大小，却具有电脑的所有基本功能。

上位机：上位机是指可以直接发出操控命令的计算机，一般是PC/host computer/master computer/upper computer,屏幕上显示各种信号变化

Swoole：PHP的异步、并行、高性能网络通信引擎

## 1.4参考资料

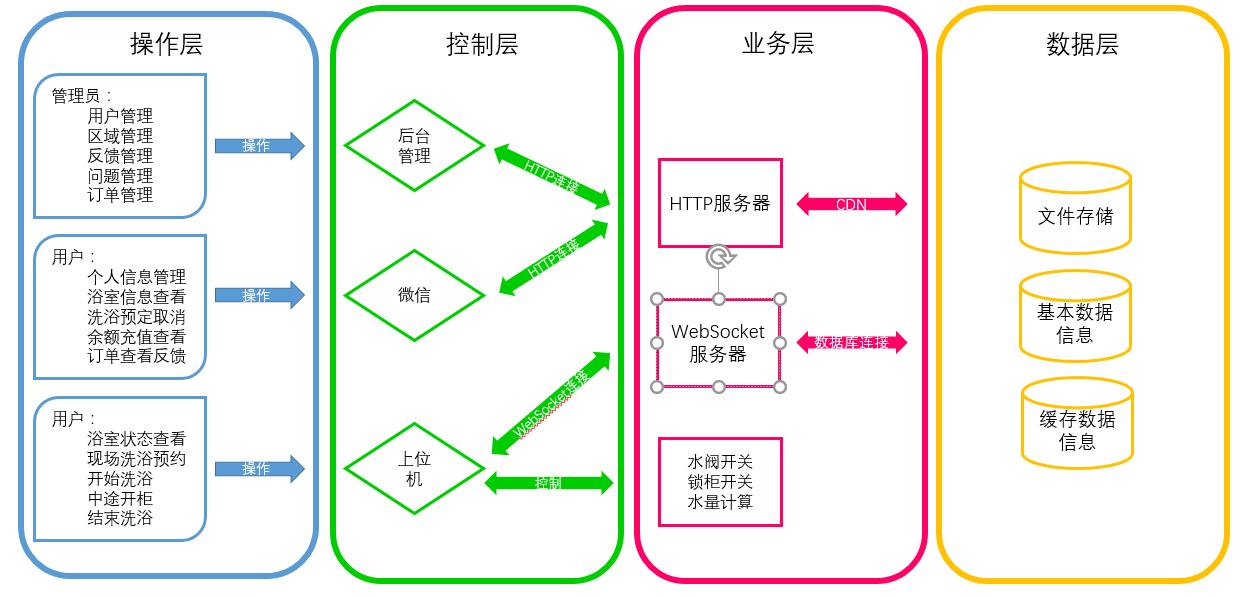
Thinkphp5完全开发手册

当swoole遇上thinkphp5

Redis设计与实现

深入浅出MySQL

# 2程序系统的结构

。

# 3程序

## 3.1程序描述

客户端：集预定今明两天洗浴、实时显示浴室状态、今明两天预订情况显示、问题反馈、优惠券使用与兑换于一体的淋浴预约系统。

上位机：使用树莓派进行开发，集成了开始、结束洗浴，中途开柜、现场预约和实时显示浴室预约情况的功能。

后台管理网站：通过web管理，可以修改客户端的相关配置，对用户及设备等信息进行管理。

## 3.2功能

## 3.3性能

响应时间：2S以内

## 3.4 限制条件

1. 本程序所存放路径不能出现中文。
2. 设备要求

a.客户端：

CPU：Core E8400 3.0Ghz 或更高CPU

内存：4GB 或更高

显卡：1G显存，GeForce 820或以上级别显卡

硬盘：剩余8G

带宽：512Kb或更高

操作系统：Windows7/Windows8(.net4.0环境)

b.移动端:

CPU:四核MT6595或更高CPU

内存:1GB或更高

操作系统:安卓4.0或更高版本

## 3.5注释设计

说明准备在本程序中安排的注释，如：

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等所加的注释；
4. 对使用的逻辑所加的注释等等。

## 3.6算法

哈希256加密算法，对密码进行第一次加密，随机截取2-8位数字，进行第二次加密，配合PHPUNIQID根据毫秒级别加熵生成唯一标志字符串

## 3.7测试计划

对各个接口进行不同请求方式、请求参数等方式进行反复测试

对移动端进行重复提交、重复绑定等方式进行返回测试

对浴室终端进行交叉功能进行反复测试

对服务器进行压力测试

## 3.9接口

<接口文档.docx>

<接口错误码.docx>

[websocket 状态码及参数.docx](websocket%20状态码及参数.docx)

## 3.9尚未解决的问题

无

# 4程序

## 4.1设计说明

部分技术的实现

1. 密码加密

*/\*密码加密存储  
 \*  
 \*加密规则：  
 \* 密码使用sha256进行单项加密  
 \* 根据时间函数uniqid获取到唯一的字符串，截取2-8位作为盐  
 \* 盐+密码  
\*/  
//密码*$password=*hash*(**'sha256'**,$data->**password1**);  
*//盐*$salt=*substr*(*uniqid*(*rand*(),**true**),2,6);  
*//加密后的密码*$password=$salt.$password;

1. 获取最大优惠券

$totalmoney = $this->**cost** \* (*ceil*($data->**use\_time** / 60));  
$coupon1list = [];*//现金券*$coupon2list = [];*//满减券*$coupon3list = [];*//折扣券*$maxmoney = 0;*//最大优惠*$couponid = **''**;*//优惠券id*$costmoney = **''**; *//最终花费金额  
  
  
//判断优惠券类型***foreach** ($coupon **as** $k =>$v) {  
**switch** ($v->**scene\_type**) {  
**case '1'**:  
*array\_push*($coupon1list, $v);  
**break**;  
**case '2'**:  
*array\_push*($coupon2list, $v);  
**break**;  
**case '3'**:  
*array\_push*($coupon3list, $v);  
**break**;  
**default**:  
**break**;  
 }  
}  
*//筛选优惠券（使用场景）  
  
//现金券***foreach** ($coupon1list **as** $k =>$v) {  
**switch** ($v->**scope**) {  
*//全部***case '1'**: {  
*//判断是否是最大优惠***if** ($maxmoney <$v->**limit**) {  
$maxmoney = $v->**limit**;  
$couponid = $v->**coupon\_id**;  
 }  
 };  
**break**;  
*//代理商***case '2'**: {  
*//判断是否是同一个代理商***if** ($v->**agent\_id** == $cell->**agent\_id**) {  
**if** ($maxmoney <$v->**limit**) {  
$maxmoney = $v->**limit**;  
$couponid = $v->**coupon\_id**;  
 }  
 }  
 };  
**break**;  
**default**:  
**break**;  
 }  
}  
*//满减券***foreach** ($coupon2list **as** $k =>$v) {  
**switch** ($v->**scope**) {  
*//全部***case '1'**: {  
*//判断是否够满减需求***if** ($coupon->**fulldown\_full** >= $totalmoney) {  
*//判断是否时最大金额***if** ($coupon->**fulldow\_down** >$maxmoney) {  
$maxmoney = $coupon->**fulldown\_down**;  
$couponid = $v->**coupon\_id**;  
 }  
 }  
 };  
**break**;  
**case '2'**: {  
**if** ($v->**agent\_id** == $cell->**agent\_id**) {  
**if** ($coupon->**fulldown\_full** >= $totalmoney) {  
**if** ($coupon->**fulldow\_down** >$maxmoney) {  
$maxmoney = $coupon->**fulldown\_down**;  
$couponid = $v->**coupon\_id**;  
 }  
 }  
 }  
 };  
**break**;  
**default**:  
**break**;  
 }  
}  
*//折扣券***foreach** ($coupon3list **as** $k =>$v) {  
**switch** ($v->**scope**) {  
**case '1'**: {  
*//折扣的金额*$dismoney = $totalmoney \* $coupon->**discount**;  
**if** ($dismoney >$totalmoney) {  
$maxmoney = $dismoney;  
$couponid = $v->**coupon\_id**;  
 }  
 };  
**break**;  
**case '2'**: {  
**if** ($v->**agent\_id** == $cell->**agent\_id**) {  
$dismoney = $totalmoney \* $coupon->**discount**;  
**if** ($dismoney >$totalmoney) {  
$maxmoney = $dismoney;  
$couponid = $v->**coupon\_id**;  
 }  
 }  
 };  
**break**;  
**default**:  
**break**;  
 }  
}  
  
*//获取出优惠最大的带金券*$maxcoupon = ShowerCoupon::*get*([**"coupon\_id"** =>$couponid]);

1. 用户预约洗浴
2. **if**($request->method()==**'POST'**) {  
   $data = *json\_decode*($request->post(**'data'**));  
   *//检测各参数是否完全***if** (**empty**($data->**begin\_time**) || **empty**($data->**end\_time**) || **empty**($data->**user\_id**) || **empty**($data->**cell\_id**)) {  
   $this->**code** = 15000;  
   $this->**msg** = **'parameter error'**;  
   $this->SendMessage();  
   **die**();  
    }  
   *//检测时间是否符合需求  
    //正则 \d{2}(((0(?=[1-9]))|(1(?=[0-2])))[0-9])((0(?=[1-9]))|(([1-2](?=[0-9]))|(3(?=0)))[0-9])([0-1](?=[0-9])|2(?=[0-4]))[0-9][0-6][0-9]  
     
    //开始时间*$begin\_time = $data->**begin\_time**;  
   *//结束时间*$end\_time = $data->**end\_time**;  
   *//当前时间*$nowtime =*date*(**"ymdHi"**,*time*());  
   *//判断时间是否已过***if**($begin\_time<$nowtime||$end\_time<$nowtime){  
   $this->**code** = 15001;  
   $this->**msg** = **'time rule error'**;  
   $this->SendMessage();  
   **die**();  
    }  
   *//正则*$zzrule = **"#\d{2}(((0(?=[1-9]))|(1(?=[1-2])))[1-9])(((0(?=[1-9]))|([1-2](?=[0-9]))|(3(?=0)))[0-9])([0-1](?=[0-9])|2(?=[0-4]))[0-9][0-6][0-9]#"**;  
   *//匹配开始时间是否符合规则  
   preg\_match*($zzrule, $begin\_time, $begincheckresult);  
   *//匹配结束时间是否符合规则  
   preg\_match*($zzrule, $end\_time, $endcheckresult);  
   *//判断是否符合规则  
    //15001***if** (**empty**(@$begincheckresult[**'0'**]) || **empty**(@$endcheckresult[**'0'**])) {  
   $this->**code** = 15001;  
   $this->**msg** = **'time rule error'**;  
   $this->SendMessage();  
   **die**();  
    }  
   *//1707232125  
    //1707232150  
    //判断预订时间顺序  
    //15002***if** ($data->**end\_time** <= $data->**begin\_time**) {  
   $this->**code** = 15002;  
   $this->**msg** = **'time queue error'**;  
   $this->SendMessage();  
   **die**();  
    }  
   *//判断时间是否准确  
     
    //判断是否跨天预订  
    //15003***if** (*substr*($data->**end\_time**, 4, 2) != *substr*($data->**begin\_time**, 4, 2)) {  
   $this->**code** = 15003;  
   $this->**msg** = **'time cross day'**;  
   $this->SendMessage();  
   **die**();  
    }  
   *//判断是否跨小时预订***if** (*substr*($data->**end\_time**, 6, 2) != *substr*($data->**begin\_time**, 6, 2)) {  
   *//是跨小时预订*$timediffer = (*substr*($data->**end\_time**, 6, 2) \* 60 + *substr*($data->**end\_time**, 8, 2))  
    - (*substr*($data->**begin\_time**, 6, 2) \* 60 + *substr*($data->**begin\_time**, 8, 2));  
    } **else** {  
   *//不是跨小时预订*$timediffer = $data->**end\_time** - $data->**begin\_time**;  
    }  
   *//判断是否超时  
    //15004***if** ($timediffer >60) {  
   $this->**code** = 15004;  
   $this->**msg** = **'time out 60'**;  
   $this->SendMessage();  
   **die**();  
    }  
   *//判断cell\_id是否正确  
    //15005***if** (!ShowerCell::*get*($data->**cell\_id**)) {  
   $this->**code** = 15005;  
   $this->**msg** = **'cell\_id not exist'**;  
   $this->SendMessage();  
   **die**();  
    }  
     
   *//根据cell\_id获取cell room block 信息*$listinfo = ShowerCell::*alias*(**'cell'**)  
    ->join(**'ShowerRoom room'**, **'cell.room\_id=room.room\_id'**)  
    ->join(**'ShowerBlock block'**, **'cell.block\_id=block.block\_id'**)  
    ->where(**"cell\_id"**, $data->**cell\_id**)  
    ->find();  
   $listinfo = *json\_decode*($listinfo);  
     
   *//判断开始时间和结束时间是否大于15分钟***if** ($timediffer <= 15) {  
   $expire\_time = $data->**end\_time**;  
    } **else** {  
   $expire\_time = $data->**begin\_time** + 15;  
    }  
     
   *//判断在指定浴室的指定时间是否冲突  
    //15006*$checktime1 = ShowerPromise::*where*(**"cell\_id"**, $data->**cell\_id**)  
    ->where(**'begin\_time'**, **'<='**, $data->**begin\_time**)  
    ->where(**'end\_time'**, **'>='**, $data->**begin\_time**)  
    ->where(**'status'**,**'1'**)  
    ->select();  
   $checktime2 = ShowerPromise::*where*(**"cell\_id"**, $data->**cell\_id**)  
    ->where(**'begin\_time'**, **'<='**, $data->**end\_time**)  
    ->where(**'end\_time'**, **'>='**, $data->**end\_time**)  
    ->where(**'status'**,**'1'**)  
    ->select();  
   **if** ($checktime1 || $checktime2) {  
   **echo** *json\_encode*([**'1'**=>$checktime1,**'2'**=>$checktime2]);  
   $this->**code** = 15006;  
   $this->**msg** = **'time conflict'**;  
   $this->SendMessage();  
   **die**();  
    }  
   *//检测余额是否充足*$price=1;  
   $totalmoney=$price\*$timediffer;  
   $user=ShowerUser::*get*([**'user\_id'**=>$data->**user\_id**]);  
   **if**($user->**money**<$totalmoney){  
   $this->**code** =16013;  
   $this->**msg**=**'money not enough'**;  
   $this->SendMessage();  
   **die**();  
    }  
     
     
   *//订单创建时间*$create\_time = *date*(**"ymdHi"**, *time*());  
   *//订单信息*$info = [**'begin\_time'** =>$data->**begin\_time**, **'end\_time'** =>$data->**end\_time**, **'expire\_time'** =>$expire\_time,  
   **'status'** =>1, **'create\_time'** =>$create\_time, **'user\_id'** =>$data->**user\_id**, **'block\_id'** =>$listinfo->**block\_id**,  
   **'room\_id'** =>$listinfo->**room\_id**, **'cell\_id'** =>$listinfo->**cell\_id**,  
    ];  
   *//开启事务*Db::**startTrans**();  
   **try** {  
   *//创建订单*$res = ShowerPromise::*create*($info);  
   *//判断是否创建成功  
    //15007***if** ($res) {  
   $nickname=ShowerUser::*where*(**'user\_id'**,$data->**user\_id**)->value(**'nickname'**);  
   $sendhost=[**'begin\_time'**=>$res->**begin\_time**,**'end\_time'**=>$res->**end\_time**,**'cell\_id'**=>$res->**cell\_id**,  
   **'nickname'**=>$nickname,**'status'**=>$res->**status**,**'room\_id'**=>$res->**room\_id**];  
     
   *//向上位机发送数据*$sendhost = *json\_encode*([**'type'** =>**'update'**, **'content'** =>$sendhost]);  
   $webres = $this->SendWebSocket($sendhost);  
   *//判断上位机是否成功接收处理数据*$webres = *json\_decode*($webres);  
   **if** ($webres->**res**) {  
   *//提交事务*Db::*commit*();  
   **echo** *json\_encode*([**"res"** =>1]);  
    } **else** {  
   *//回滚*Db::*rollback*();  
   **echo** *json\_encode*([**"res"** =>0]);  
    }  
    } **else** {  
   **echo** *json\_encode*([**'res'** =>0]);  
    }  
    }**catch** (\Exception $e){  
    Db::*rollback*();  
   **echo** *json\_encode*([**'res'**=>0]);  
    }  
   }