**<体育共同体管理系统>软件开发过程**

**4. 详细设计**

（1）主要模块算法思想描述

1）输入用户登录信息：连接“登录信息”表，根据表中字段输入所有用户登录信息，完成输入后提交用户登录信息。其中，“登录用户uid”根据用户创建帐号日期自动生成，其他表中字段进行取值范围检查。

2）输入小组详细信息：连接“小组信息”表，根据表中字段输入所有小组详细信息，完成输入后提交所有小组详细信息。其中，所有小组详细信息字段均由用户输入得到，需进行取值范围检查。

3）核对小组信息：

4）小组信息合法性检查：

5）显示登录用户信息：

6）显示小组简略信息：

（2）模块算法设计

1）“输入用户登录信息”算法过程设计（）：

Function shuruyhdlinfo()

{

连接“登录信息”表；

根据界面提示输入用户名和密码信息；

int flag = 0；

if 用户名在登录信息表中，then：

if 密码是正确的，then：

printf("登录成功")；

Elseif 密码是错误的，then：

printf("用户名或密码错误")；

Elseif 用户名不在登录信息表中，then：

{

flag = 1；

printf("请先注册账号")；

}

End

if flag，then：

根据界面提示输入注册账号的用户名和密码；

if 用户名已经在登录信息表中，then：

printf("该用户名已被注册")；

Elseif 用户名不在登录信息表中，then：

if 注册的用户名和密码是合法的，then；

printf("注册成功")；

Elseif 注册的用户名和密码是不合法的，then：

printf("请输入合法的用户名和密码")；

End

End

提交新用户信息；

}

2）“输入小组详细信息”算法过程设计：

Function shuruteaminfo()

{

连接“小组信息”表；

根据界面提示输入运动类型、运动时间、运动地点、运动水平和需求人数信息；

if 输入的小组详细信息是合法的，then；

printf("发布小组成功")；

Elseif 输入的小组详细信息是不合法的，then：

printf("请输入合法的小组详细信息")；

End

提交新小组信息；

}

3）“小组信息合法性检查”算法过程设计：

Function checkteaminfo(运动类型 ydlx，运动时间 ydsj，运动地点 yddd，运动水平 ydsp，需求人数 xqrs)

{

运动类型集合 = {“篮球”， “足球”，“排球”，“羽毛球”，“乒乓球”，“跑步”，“游泳”，“健身”}；

运动时间集合 = {“周一”，“周二”，“周三”，“周四”，“周五”，“周六”，“周日”}；

运动地点集合 = {“篮球馆”， “足球场”，“排球场”，“羽毛球场”，“乒乓球馆”，“操场”，“游泳馆”，“健身馆”}；

运动水平集合 = {“High”，“Mid”，“Average”}；

需求人数集合 = {10，22，12，4，4，10，4，10}；

int flag = 1;

连接“小组信息”表；

if 运动类型信息不在运动类型集合中，then：

{

flag = 0;

printf("小组信息中运动类型信息不合法")；

}

End

if 运动时间信息不在运动时间集合中，then：

{

flag = 0;

printf("小组信息中运动时间信息不合法")；

}

End

if 运动地点信息不在运动地点集合中，then：

{

flag = 0;

printf("小组信息中运动地点信息不合法")；

}

End

if 运动水平信息不在运动水平集合中，then：

{

flag = 0;

printf("小组信息中运动水平信息不合法")；

}

End

if 需求人数信息不在需求人数集合中，then：

{

flag = 0;

printf("小组信息中需求人数信息不合法")；

}

End

if flag, then:

printf("小组信息合法")；

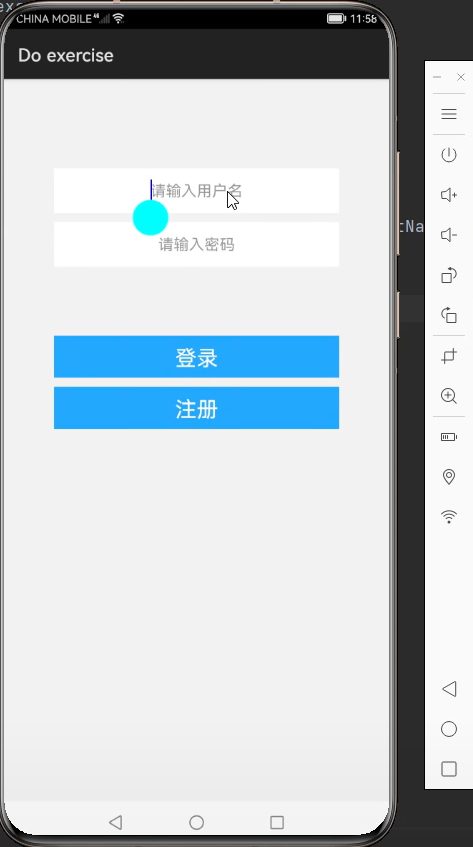
End

}

（3）用户界面设计

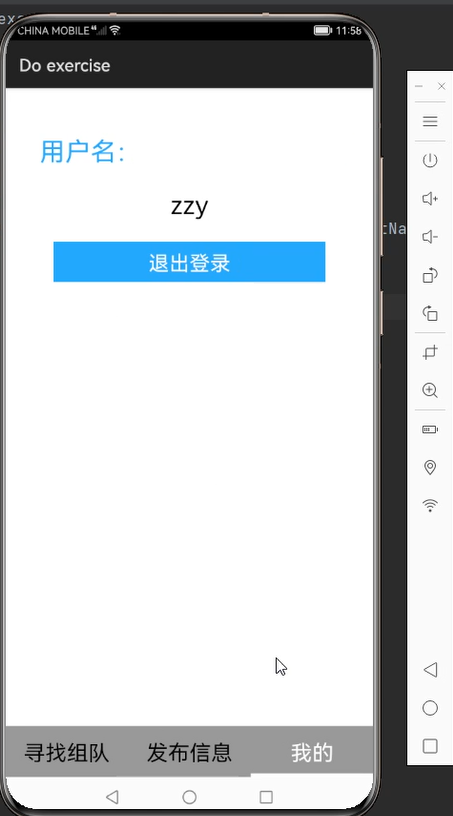
1）小程序界面设计：

a）用户登录界面设计：

b）用户注册界面设计：

用户登录界面 用户注册界面

2）主要流程界面设计：

a）“当前登录用户”界面：

b）“发布信息”界面：



c）“寻求组队”界面：

