**系统概要设计说明书**

**1. 引言**

**1.1 编写目的**

本文档编制的目的是说明对软件系统的设计考虑，包括软件系统的基本处理流程，软件系统的组织架构，接口设计，运行设计，数据库结构设计和难点及其解决方案。

**1.2 背景**

我们团队开发的软件是基于Android的手机拔河APP

**1.3 定义**

Mysql:系统服务器适应的数据库管理系统

SQL：一种用于访问查询数据库的语言

主键：数据库表中的关键域，值互不相同

**1.4 参考资料**

Android第一行代码(第二版) 郭霖著

**2. 系统总体设计**

**2.1 需求规定**

**2.1.1 系统功能**

（1）客户端：采集安卓手机上加速度传感器的数据（左右摇/上下摇），并将其传送到服务器。同时接收服务端返回的计数统计结果，实时在客户端上显示这一结果。

（2）服务器：对收集到的所以客户端数据进行处理，并将结果返回到手机，以可视化形式显示。

（3）详细需求分析：

①客户端要实现注册登录功能，要有注册登录界面并存入服务器数据库中；

②注册登录后进入主页面，需要设计主页面，主页面要有创建房间按钮、加入房间按钮和查询历史战绩按钮；

③在创建房间界面进行房间创建，加入房间界面输出房间号加入房间；

④设计游戏界面，界面上要有开始游戏按钮；

⑤设计结果显示界面，显示游戏结果。

**2.1.2 系统性能（精度，可靠性）**

（1）精度：精准读取加速度传感器的数据，显示手机摇晃的次数。

（2）可靠性：软件使用过程中未出现卡死或者数据获取出错等问题，可持续运行。**2.1.3 条件与限制**

服务器端采用了云服务器，所以在安装APP的前提下必须联网才能进行游戏。

**2.1.4 详细设计方法与工具**

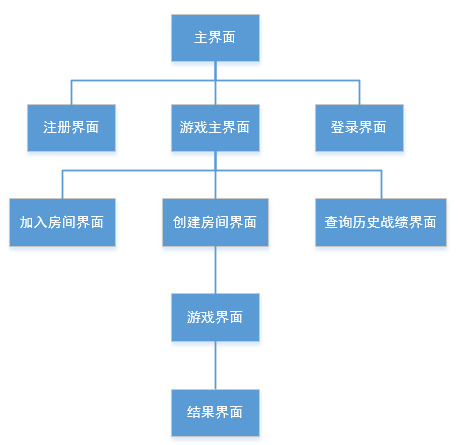
Xshell作为服务器端操作的处理软件，进行数据库操作，jsp编程等。Android studio作为主要开发软件，对界面、http请求、单机实验过程等进行编程。

**2.2 运行环境**

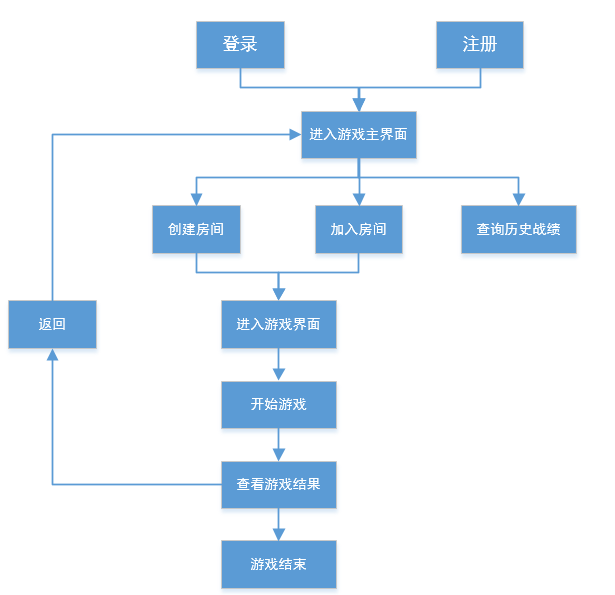
各种可连接互联网的Android设备，如Android手机，装有Android系统的平板电脑

**2.3 基本设计结构和处理流程**

（1）下图描述了整个游戏的界面总体结构



（2）下图描述了整个游戏的运行流程



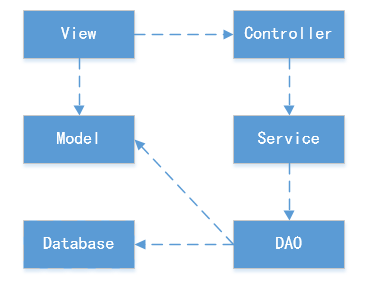
**2.4 系统架构设计**

**2.4.1 系统逻辑架构设计**

软件采用C/S模式，服务器负责对收集到的所以客户端数据进行处理，并将结果返回到手机，以可视化形式显示；客户端负责采集安卓手机上加速度传感器的数据（左右摇/上下摇），并将其传送到服务器，同时接收服务端返回的计数统计结果，实时在客户端上显示这一结果。

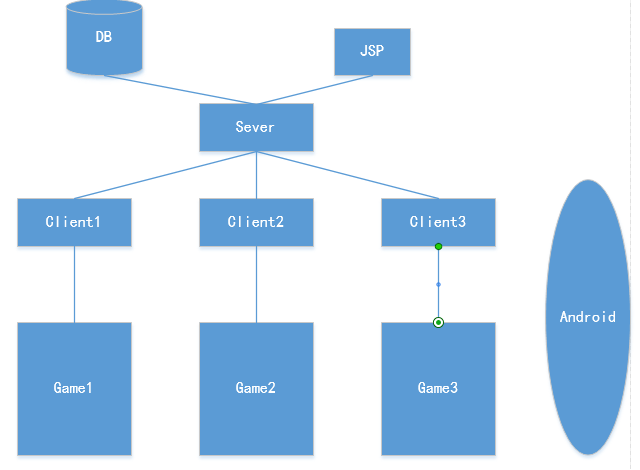
软件整体设计采用MVC的架构模式。系统大致分为五层：控制层（Controller）、业务逻辑层（Service）、数据访问层（DAO）、数据模型层（Model）、视图层（View）。数据访问层主要对数据库进行操作，包括增加数据、修改数据和删除数据。业务逻辑层基于数据访问层之上，它主要针对业务流程来实现系统的功能。控制层又基于业务逻辑层之上，主要控制请求流转，并处理业务逻辑层通过数据访问层得到的数据在视图层进行展示。视图层为软件提供展示界面。

结构依赖如图：



**2.4.2 系统物理架构设计**

物理结构上采用C/S模式，百度云服务器作为服务器端，进行数据操作及客户端发来的用户请求处理；Android手机端APP作为客户端，作为进行软件基本功能实现的平台，作为给用户展示的视图层。



**2.5 系统详细界面划分**

开始界面： 注册界面：

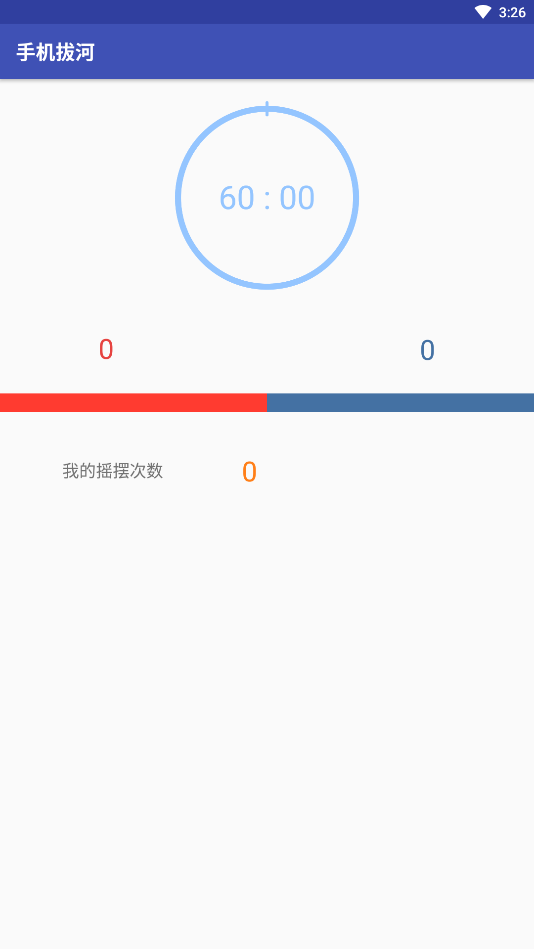
登录界面： 游戏主界面：

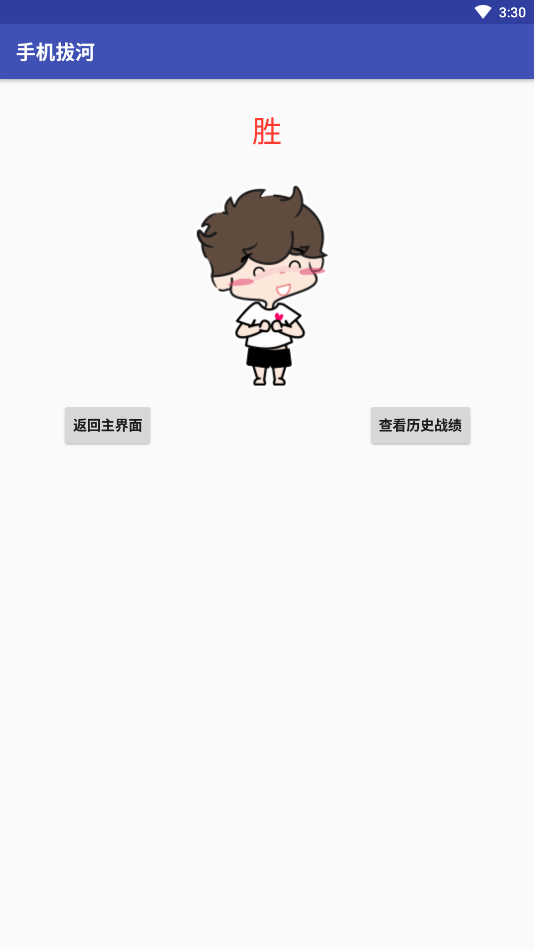
创建房间界面： 加入房间界面：

查询历史战绩界面： 游戏界面：

游戏结果界面：



**3．接口设计**

内部接口：Http回调监听接口。实现客户端服务器之间的Http请求处理。

**HttpCallbackListener.java:**

package com.example.liugang.tugofwar;

public interface HttpCallbackListener {

void onFinish(String response);

void onError(Exception e);

}

1. **系统程序代码详细设计**

Http回调函数

**HttpUtil.java：**

package com.example.liugang.tugofwar;

import java.io.BufferedReader;

import java.io.DataOutputStream;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class HttpUtil {

public static void sendHttpRequest(final String req\_url,final String request\_info,final HttpCallbackListener listener){

new Thread(new Runnable() {

@Override

public void run() {

HttpURLConnection connection = null;

try {

String request\_url = "http://106.13.37.201:8000/" + req\_url;

URL url = new URL(request\_url);

connection = (HttpURLConnection) url.openConnection();

connection.setRequestMethod("POST");

connection.setConnectTimeout(8000);

connection.setReadTimeout(8000);

connection.setDoInput(true);

connection.setDoOutput(true);

DataOutputStream out = new DataOutputStream(connection.getOutputStream());

out.writeBytes(request\_info);

InputStream in = connection.getInputStream();

BufferedReader reader = new BufferedReader((new InputStreamReader(in)));

StringBuilder response = new StringBuilder();

String line;

while ((line = reader.readLine()) != null) {

response.append(line);

}

String result = response.toString();

if(listener != null)

{

listener.onFinish(result);//回调方法

}

} catch (Exception e) {

if(listener != null)

listener.onError(e);

}finally {

if(connection != null)

connection.disconnect();

}

}

}).start();

}

}

**AndroidManifest.xml：**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.liugang.tugofwar">

<uses-permission android:name="android.permission.GET\_ACCOUNTS" />

<uses-permission android:name="android.permission.READ\_PROFILE" />

<uses-permission android:name="android.permission.READ\_CONTACTS" />

<uses-permission android:name="android.permission.INTERNET" />

<application

android:allowBackup="true"

android:icon="@drawable/launcher\_icon"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name=".SigninActivity"

android:label="@string/signin" />

<activity android:name=".GameMainActivity" />

<activity

android:name=".LoginActivity"

android:label="登录" />

<activity android:name=".RoomActivity" />

<activity android:name=".PlayingActivity" />

<activity android:name=".MyHistoryActivity" />

<activity android:name=".JoinRoomActivity" />

<activity android:name=".ResultActivity"></activity>

</application>

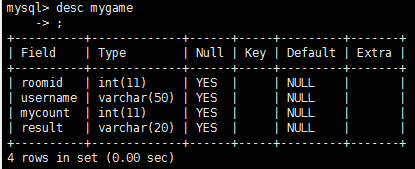
</manifest>

**5. 系统数据库设计**

根据软件需求，设计出能满足用户需求的数据实体，得到数据库支持的数据模型，在服务器端设计Mysql数据库的数据表。共包含五张表：我的游戏表，房间表，房间信息表，开始信息表，用户信息表。

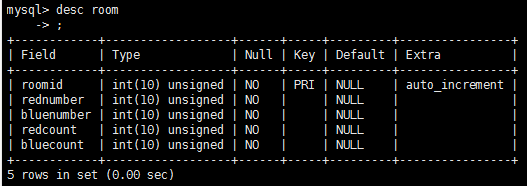
（1）我的游戏表（mygame）

包含四项内容：roomid、username、mycount、result



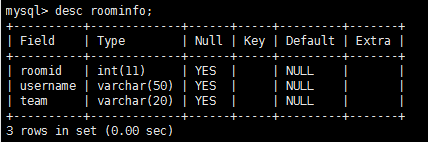
（2）房间表（room）

包含五项内容：roomid、rednumber、bluenumber、redcount、bluecount



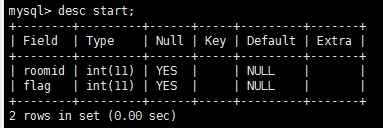
（3）房间信息表（roominfo）

包含三项信息：roomid、username、team



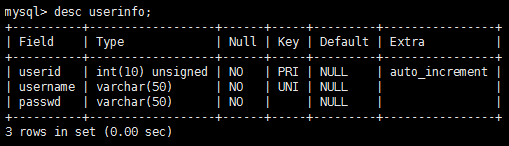
（4）开始信息表（start）

包含两项信息：roomid、flag



（5）用户信息表（userinfo）

包含三项信息：userid、username、passwd



**6. 难点及解决方案**

（1）难点：Http访问

解决方案：编写HttpCallbackListener接口，通过回调函数访问

（2）难点：更新UI

解决方案：异步消息处理机制

**系统详细设计说明书**

**1. 引言**

**1.1 编写目的**

本文档编制的目的是说明对软件系统的设计考虑，包括软件系统的基本处理流程，软件系统的组织架构，接口设计，运行设计，数据库结构设计和难点及其解决方案。

**1.2 背景**

我们团队开发的软件是基于Android的手机拔河APP

**1.3 定义**

Mysql:系统服务器适应的数据库管理系统

SQL：一种用于访问查询数据库的语言

主键：数据库表中的关键域，值互不相同

**1.4 参考资料**

Android第一行代码(第二版) 郭霖著

**2. 设计概述**

**2.1 任务和目标**

**2.1.1 需求概述**

（1）客户端：采集安卓手机上加速度传感器的数据（左右摇/上下摇），并将其传送到服务器。同时接收服务端返回的计数统计结果，实时在客户端上显示这一结果。

（2）服务器：对收集到的所以客户端数据进行处理，并将结果返回到手机，以可视化形式显示。

**2.1.2 运行环境概述**

各种可连接互联网的Android设备，如Android手机，装有Android系统的平板电脑。

**2.1.3 条件与限制**

服务器端采用了云服务器，所以在安装APP的前提下必须联网才能进行游戏。

**2.1.4 详细设计方法与工具**

Xshell作为服务器端操作的处理软件，进行数据库操作，jsp编程等。Android studio作为主要开发软件，对界面、http请求、单机实验过程等进行编程。

**3. 系统详细需求分析**

**3.1 详细需求分析**

（1）客户端要实现注册登录功能，要有注册登录界面并存入服务器数据库中；

（2）注册登录后进入主页面，需要设计主页面，主页面要有创建房间按钮、加入房间按钮和查询历史战绩按钮；

（3）在创建房间界面进行房间创建，加入房间界面输出房间号加入房间；

（4）设计游戏界面，界面上要有开始游戏按钮；

（5）设计结果显示界面，显示游戏结果。

**3.2 详细系统运行环境及限制条件分析接口需求分析**

Android4.0.3

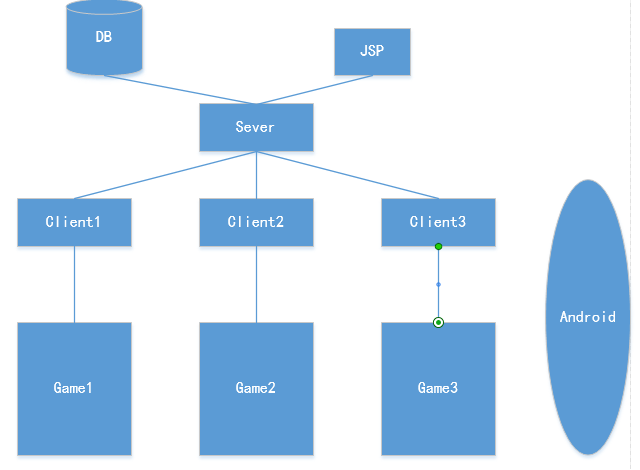
**4．总体方案确认**

**4.1 系统总体结构确认**

**4.2 系统详细界面划分**

**5. 系统详细设计**

**5.1 系统结构设计**



**5.2 系统功能模块详细设计**

**5.3 系统界面详细设计**

**5.4 系统程序代码详细设计**

**6. 数据库系统设计**

**6.1 设计要求**

**6.2 信息模型设计**

**6.3 数据库设计**

**6.3.1 设计依据**

**6.3.2 数据库选型**

**6.3.3 数据库种类及特点**

**6.3.4 数据库逻辑结构**

**6.3.5 物理结构设计**

**6.3.6 数据库安全**

**6.3.7 数据字典**