

Design and Implementation of Social Sports App Based on MQTT

Xue Yuanjie, Wang Tingmei*

Applied Science and Technology College, Beijing Union University, Beijing 100101, China

Abstract: In view of the urgent requirements of sports social communication and the lack of APP in sports social communication, by analyzing the open source and easy-to-use programming characteristics of Android, adopting the modular idea, using the C/S communication model and combining Servlet technology and JDBC database-access technology, a social function with efficient interaction between mobile phone and database based on MQTT(Message Queuing Telemetry Transport) protocol is designed and implemented. The system. The test results show that the system has stable operation, good interaction function, strong scalability, easy to upgrade and maintain, and meets the requirements of mobile software compatibility, real-time and security.

Keywords: Android, Database, MQTT, Servlet, Social sports

1. Introduction

Perhaps you have noticed that there are more and more people running in the stadium and on the street every day. Perhaps you have noticed that there will be new basketball and football fields in every corner of the city. People's love for sports is also increasing year by year. Young people like competitive sports, while the middle-aged and old people are more inclined to keep fit. In a word, more and more people are active in obtaining information of sports events and events. At the same time, this is an era where everyone has strong social needs. For those friends who lack social contact and are fond of sports, they need to use a new carrier to urge people to play sports and make friends. In order to solve this problem, this paper designs and implements a system based on MQTT(Message Queuing Telemetry Transport) protocol, using C/S communication model, Servlet technology and JDBC database-access technology. The design of the system is simple and clear, the operation is convenient, and it provides a platform for users to communicate in detail.

2. Message Queuing Telemetry Transport (MQTT)

2.1 MQTT Introduction

MQTT was proposed by IBM in 1999, and the latest version is 3.1.1. It is a message transmission protocol with publish subscribe mode, which is built on TCP/IP [1]. The initial purpose of the design is to achieve unreliable communication between extremely limited

memory devices and networks with low network bandwidth.

2.2 Characteristics of MQTT

Using Publish/Subscribe Message Mode, One to Many Publishing of Messages Can Be Realized and Application Coupling Can Be Decoupled;

It Can Realize the Message Transmission of Shielding the Load Content;

It Can Provide Three Kinds of Service Quality of Transmission Message, and Users Can Weigh Efficiency and Service Quality According to Actual Demand;

Set Up Small-Scale Transmission with Little Cost, Saving Network Traffic for Users;

Provide a Notification Mechanism of Client Abnormal Interrupt.

2.3. Structure of MQTT

There are three identities in the MQTT protocol: Publisher, Broker and Subscriber. First, the client and the Broker need to establish the connection through the interactive connection request message, then the client Publishes the message to the message Broker, and the subscriber can subscribe the message to the message Broker. MQTT follows a TCP-based connection establishment procedure. The device sends MQTT a request message, CONNECT, to connect with the broker. Once the request is received, the broker will send the acknowledgment, CONNACK, to the sending device [1]. In this protocol model, the Broker acts as a forwarder, and the forwarded messages are distinguished by subject [3]. Figure 1 shows the general MQTT message model. MQTT realizes many to many communication through this message mode, with high flexibility, and the sending device is not directly connected with the receiving device, thus realizing the decoupling between Publisher and Subscriber [5].

3. Analysis of Servlet and Jdbc Technology

3.1. About Servlet

Servlet realizes its functions by using Java Servlet API (Application Programming Interface) and its related classes and methods. The biggest advantage of Java Servlet API is the protocol independence. It does not assume the protocol used by network transmission, how the servlet is loaded and the application server environment in which it runs [2]. These features make Java Servlet API easily embedded in many different

kinds of application servers. In this design, servlet mainly plays the role of control and server-side logic function execution. It can dynamically send information to Android side, playing the role of C in MVC model. Through servlet, it can easily receive and respond to Android side information with the help of HTTP protocol [4]. Servlet inherits a series of advantages of Java, such as automatic memory recycling, good object-oriented features and exception handling mechanism. The end can call Servlet through Hyperlink, Form, Tag and other forms.

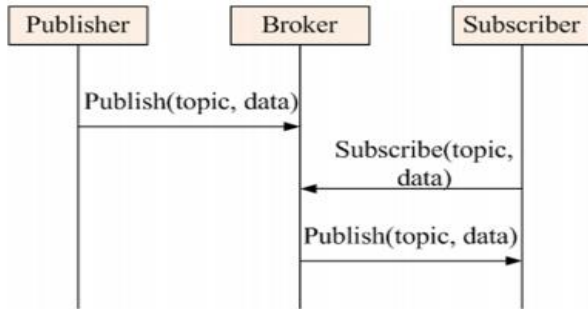


Figure 1. MQTT publish/subscribe topic message model.

3.2. About Jdbc

This system adopts the scheme of Android/servlet + jdbc + MySQL in data access. On the Android side, query data is sent to the server through HTTP; on the server side, after the server response program (Servlet) receives the query request, it accesses and operates MySQL database by executing SQL statements in the program and using a standard API provided by JDBC for tool database developers. The Servlet transmits the queried data to the Android side, which parses the data through the parser and displays the data on the Activity [2]. Figure 2 shows the Process of JDBC Access DataBase.

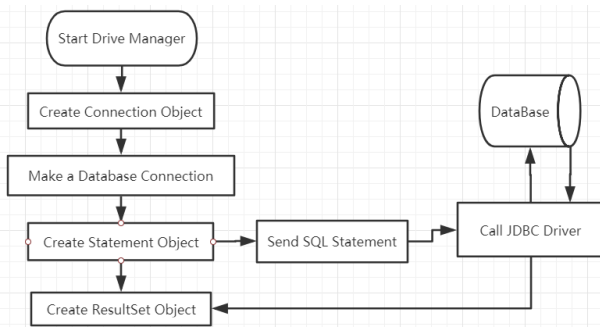


Figure 2. Process of jdbc access database.

There are many advantages in using JDBC to access data:

- ① Providing a standard API for developers, enabling them to write database applications with pure Java API;
- ② Combining java with JDBC, making the program cross platform;
- ③ Using JDBC to realize the connection between applications and various databases;
- ④ JDBC extending Java's Function.

3.3. Design of Interaction

According to the analysis of Servlet and JDBC, a feasible interaction scheme is put forward: using wireless communication interface in Android to submit SQL statement to web server, web server uses request object of Servlet to obtain information and execute SQL statement with JDBC technology, and then sends the information returned by DataBase back to Android in response object [6]. Therefore, the interaction process is composed of two parts: 1. Communication between Android and web server; 2. Interaction between web server and DataBase. The web server of this system is Tomcat 7.0. Figure 3 is a schematic diagram.

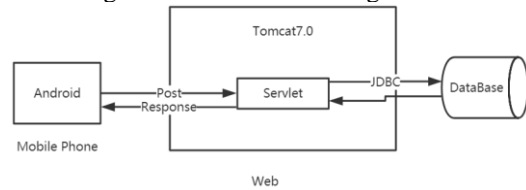


Figure 3 Schematic of Interaction.

4. Design of System Software

Eclipse is an open-source, Java based extensible development platform, through various plug-ins based on the framework platform, users can flexibly and freely create the services they need. This system uses eclipse to build Android development environment: first install Java JDK (Java Development Kit) and configure its environment variables, then install Java JRE (Java Runtime Environment) and eclipse, finally install Android SDK (software development kit) and Android ADT (Android Development Tools). After the successful establishment of the development environment, Android programs can be easily debugged and quick tests can be completed using virtual machines. This system mainly uses Android mobile phones (one Meizu Pro 6 and one Samsung Galaxy S8) as the main testing tools.

4.1 Android Software Structure Design and Implementation

The system consists of two parts, Android and PC. There are five modules in Android, which are News, Schedule, BBS, Agreement of balls and Me. Users can view the latest sports news, understand the schedule of sports events, view their favorite teams, browse the photos of their friends and watch the details of sports events, and sign up for the related activities of surrounding clubs. There are five modules in the PC, which are Integrated Information, Surrounding Clubs, User Management, Communication Platform and System Settings. The PC is mainly in the charge of the administrator. It operates through the CRUD of the Database Table, which is mainly for the management of the sports news displayed by Android, the ball appointment review and other related information. Figure 4 shows the Structure of Social Sports App.

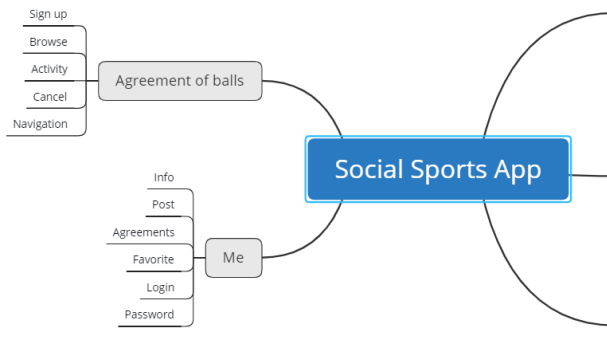


Figure 4 Structure of social sports app.

4.2. Design of Part I

Part I is News. Its main function is to show sports news, including some sports reviews. Figure 5 shows the Process of News. Figure 6 shows the Interface of News.

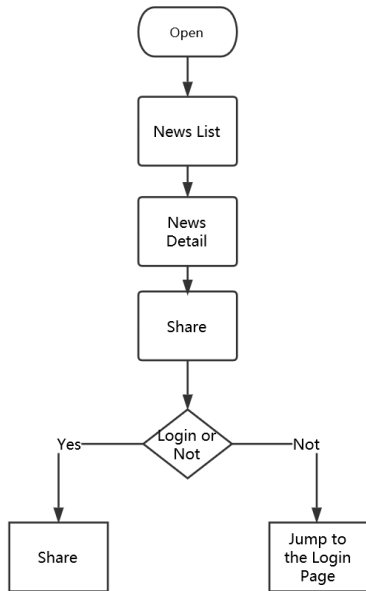


Figure 5 Process of news.

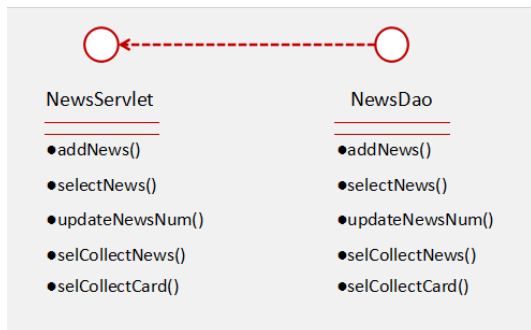


Figure 6 Interface of news.

4.3. Design of Part Ii

Part II is Schedule. Its main function is to list the ongoing matches in the form of Basketball-Football, and display the real-time score statistics and live game by

clicking the game to call the network interface. Here, we preload all team and player information and save it locally. Figure 7 shows the Process of News. Figure 8 shows the Interface of News.

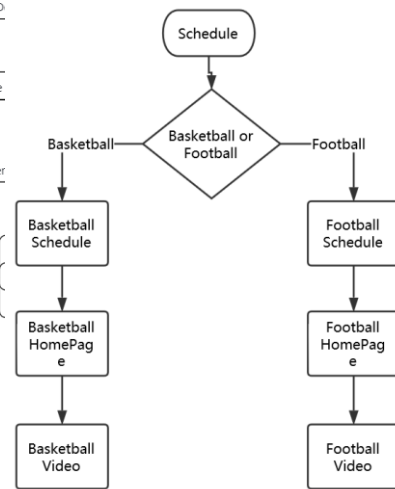


Figure 7 Process of schedule.

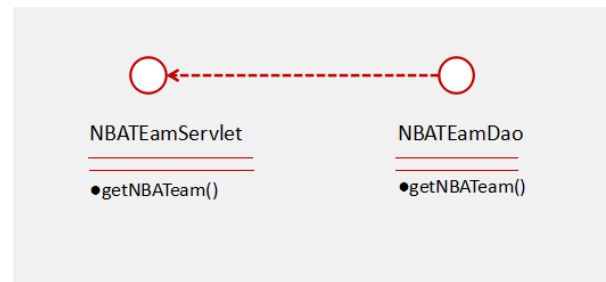


Figure 8 Interface of schedule.

4.4. Design of Part Iii

Part III is BBS. Its main function is to manage and display the posts published by users and administrators. All users can comment, collect and other functions, post publishing, browsing, collecting, commenting, and managers can delete. Posts can be posted with pictures. Figure 9 shows the Process of BBS. Figure 10 shows the Interface of BBS.

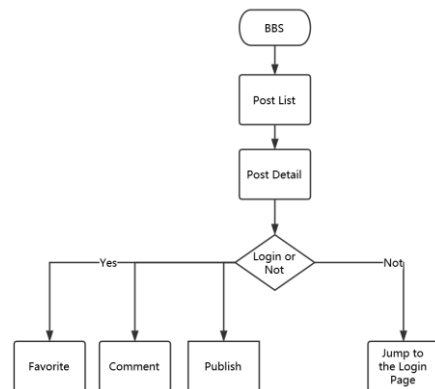


Figure 9 Process of Bbs.

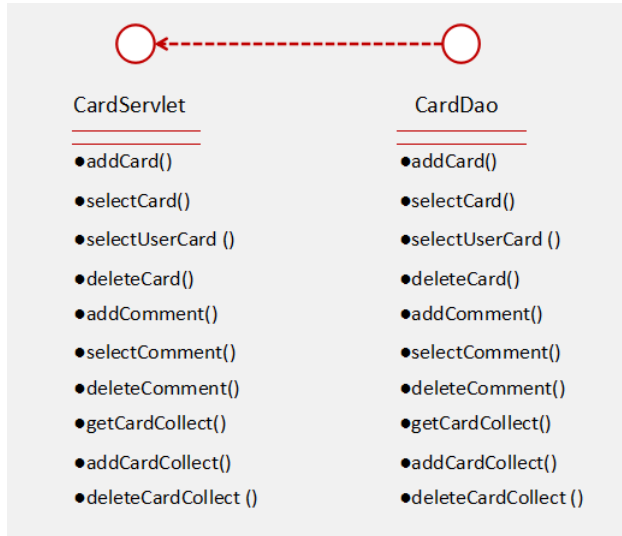


Figure 10 Interface of Bbs.

4.5. Design of Part Iv

Part IV is Agreement of balls. Its main function is to initiate and browse the ball appointment activity. The administrator only plays the role of audit. All users can launch, browse, sign up to participate and other functions. They can click in the internal address bar of the ball appointment and navigate to the appointed place with the help of the third-party SDK. The appointment can be released with pictures. Figure 11 shows the Process of Agreement of balls. Figure 12 shows the Interface of Agreement of balls.

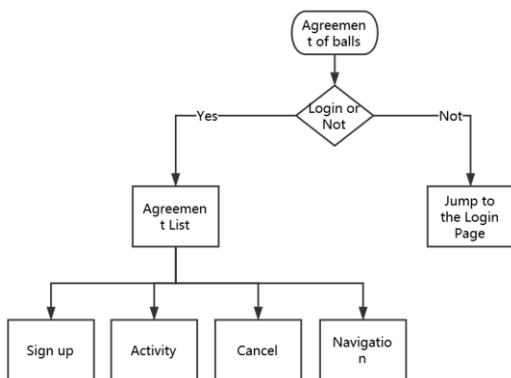


Figure 11 Process of agreement of balls.

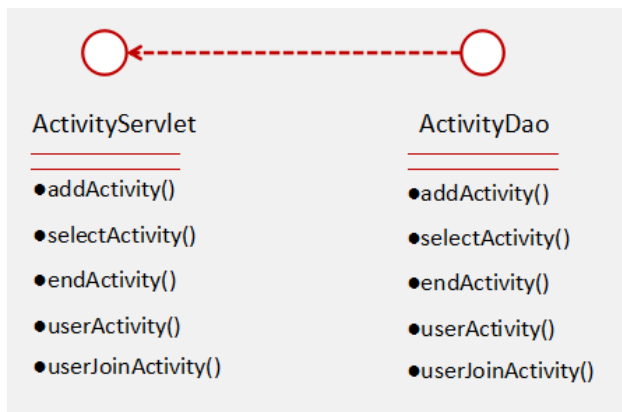


Figure 12 Interface of agreement of balls.

4.6. Design of Part V

Part V is Me. Its main functions are the display and modification of personal information; the use of functions such as collecting, posting, initiating, participating and setting. Figure 13 shows the Process of Me. Figure 14 shows the Interface of Me.

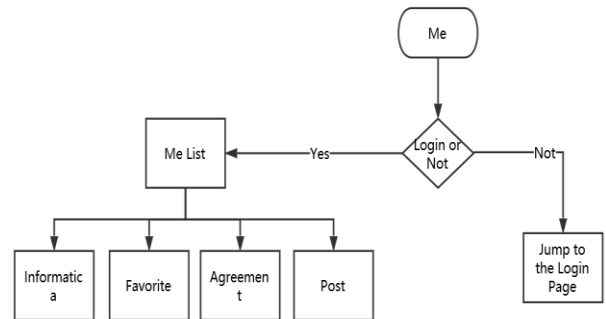


Figure 13 Process of Me.

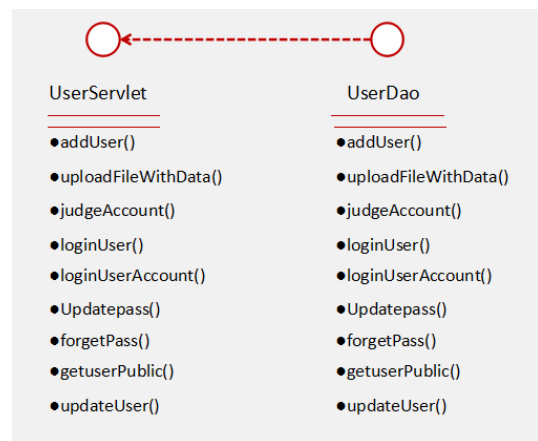


Figure 14 Interface of Me.

5. Implement of System Software

The Android App is developed by eclipse with related plug-ins and Java language. The web server uses Tomcat 7.0 as the Servlet container, and MyEclipse and Java language are used to build the Servlet class. In this App, the initialization phase is mainly to realize the deployment of web server and listen to Android requests.

5.1. Chat Module Based on MQTT

Before use, the following parameters are required to configure MQTT on Android side:

Topic: subscribed events. URI: address of the MQTT server. Username & Password: account and password. ClientID: Android ID, which can be customized and must be unique. Otherwise, the server will be disconnected when connecting to the server. In order to facilitate the confirmation of users and better communication between users and users, the Android ID is used as the Android ID in this App.

First, add dependency; second, configure Android related parameters; third, connect to MQTT server; fourth, manage MQTT connection and message sending and receiving through service background service; fifth, realize chat interface.

Code Example:

```

/** * Register Broadcast */
class ServiceBroad extends BroadcastReceiver {
public void onReceive(Context arg0, Intent arg1) {
startMQTTService();
}
}
/** Register service of MQTT */
public void startMQTTService() {
Log.e("MQTT Service", "Start Service");
MyApplication app = (MyApplication)
getApplication();
int id = app.getUser().getId();
Editor editor =
getSharedPreferences(PushService.TAG,
MODE_PRIVATE).edit();
editor.putString(PushService.PREF_DEVICE_ID,
String.valueOf(id));
editor.commit();
PushService.actionStart(getApplicationContext());
}

```

5.2. Implement of Android

In Android, HttpPost object is used to send and receive information. In this way, Publisher's data encapsulation and Subscriber's data analysis can be automatically completed. When sending, you only need to encapsulate the data name and attribute value and submit the data to the server. Finally, you can receive the information through the execute method of HttpResponse object. On the Web Server side, the information is encapsulated as JSON objects and sent back to the Android side. Therefore, it is necessary to write a parser class on Android side to parse the data and display it on the activity.

Code Example:

```

public class ActivityParser extends BeanParser {
public BeanData parser(String result) {
ActivityData aData = new ActivityData();
try {
JSONObject json = new JSONObject(result);
aData.setFlag(json.getInt("flag"));
aData.setCode(json.getInt("code"));
if
(aData.getFlag() == StatusCode.Dao.SELECT_SUCCESS)
{
ArrayList<ActivityEntity> list = new
ArrayList<ActivityEntity>();
JSONArray jsonR = json.getJSONArray("list");
for(int i=0;i<jsonR.length();i++){
JSONObject activity = jsonR.getJSONObject(i);
ActivityEntity aEntity = new ActivityEntity();
aEntity.setActivityId(activity.getInt("activityId"));
aEntity.setActivityTitle(activity.getString("activityTitl
e"));
aEntity.setActivityDate(activity.getString("activityDat
e"));
aEntity.setActivityMess(activity.getString("activityMe
ss"));
list.add(aEntity);
}
}
}
}

```

```

}
aData.setList(list);
}
} catch (JSONException e) {
e.printStackTrace();
}
return aData;
}
}

```

5.3. Implement of Web Server

On the Web Server side, the getWriter method of HttpServletResponse object is used to send information, and the Java object is transformed into JSON format for sending. HttpServletRequest object is used to receive information.

Code Example:

```

public class BaseServlet extends HttpServlet {
Public Gson gson=new
GsonBuilder().setDateFormat("yyyy-MM-dd
hh:mm:ss").create();
public void requestProcess(Callback call,
HttpServletRequest request, HttpServletResponse
response) {
Map<String, Object> map = new HashMap<>();
String result = null;
int status = StatusCode.Common.SUCCESS;
try {result = call.callback(map);} catch (Exception e) {
status = StatusCode.Common.FAIL;
e.printStackTrace();}
map.put("code", status);
if (result == null) {
response.setContentType(CONTENTTYPE_JSON);
String json = gson.toJson(map);
try {response.getWriter().write(json);} catch
(IOException e) {
e.printStackTrace();}
}
if (result != null) {
response.setContentType(CONTENTTYPE_HTML);
if (result.equals("")) {
throw new PageException("Page return value cannot
be null !");}
request.setAttribute("map", map);
try
{request.getRequestDispatcher(result).forward(request,
response);
} catch (ServletException e) {
e.printStackTrace();} catch (IOException e)
{e.printStackTrace();}
}
}
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {}
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {}
}
}

```

5.4. Implement of Database Interaction

In the process of interaction with DataBase, PreparedStatement object is used in this system. Its advantage is that it can greatly improve efficiency for batch processing. PreparedStatement object not only contains SQL statements, but also has been precompiled in most cases. Therefore, when it is executed, only DBMS is required to run SQL statements without compiling first. Secondly, the PreparedStatement object passes parameters through ‘?’, which can prevent SQL injection and has high security.

Code Example:

```
public int addActivity(ActivityEntity aEntity){
    int row = 0;
    openConnection();
    try {
        ps=con.prepareStatement("insert into
sp_activity(activityId,activityTitle,userId,activityDate,act
ivityMess)
values(sq_spactivity.nextval,?,?,?)");
        ps.setString(1,aEntity.getActivityTitle());
        ps.setInt(2, aEntity.getUserId());
        SimpleDateFormat df = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        ps.setString(3, df.format(new Date()));
        ps.setString(4, aEntity.getActivityMess());
        row = ps.executeUpdate();
    } catch (SQLException e)
    {e.printStackTrace();}finally{
```

Table 1 Partial Test Summary

Test unit	Test page	Function description	Expected results
News information	News added	Fill in the news information and update the sports news status in time	Add success
	News revision	Modify the news information	Modification successful
	News viewing	Display all cache news information (including fuzzy queries)	Find the correct
	News deletion	Delete selected news	Delete successfully
Post information	Post addition	Add a post	Add success
	Post modification	Modify the added post	Modification successful
	Post View	Display all posts (including fuzzy queries)	Find the correct
	Post Delete	Delete Post	Delete successfully
Post comments	Comments added	Comment on the post, add information	Add success
	Comments and modifications	To modify a post comment	Modification successful
	Comment view	Display all comments on the post	Find the correct
	Comment deletion	Delete post comments	Delete successfully

```
closeConnection();}
return row;
}
```

6. Test

6.1. Test Conditions and Environment

There are many options for Test Conditions: if the DataBase is located in the intranet computer, the Android terminal cannot use the mobile network for test, and it must be connected to the same network as the DataBase before test. If the DataBase is located in an external network, the computer where the DataBase is located needs to have an IP and port number of the public network before the next test can be carried out. At this time, Android can directly use the mobile network for testing.

Test Environment:

Operating system: the operating system is Windows 7 or above equipment;

Support environment and version: Tomcat 7.0.24;

Hardware platform: 8G memory, 2G + hard disk, i5 + processor, 1600x900 resolution;

Development environment: myeclips10 / myeclips2015;

The software can run on Android 2.2 or above, and the device needs to have networking function, memory of 100Mb and installation space of 40Mb.

6.2. Test Summary

Table 1 Shows a Part of the Test Summary, Which Mainly Extracts Some Test Cases When Android side and Webserver Interact with Servlet and Jdbc, and Expects the Test Cases.

Table 2 Partial Test Results

Test operation	Expected results	Actual results	Test results
Add news name, number, title, author, date to add information.	The news message was added successfully.	Add success	pass
Click add when filling in the empty information	Pop-up prompt message prompt information must be filled in	The pop-up prompt message prompt must be filled in	pass
Click to view the page (including fuzzy queries)	Display view information	Display view information	pass
Click on the news information to modify	Go to the Modify Page	Go to the Modify Page	pass
Fill in empty information when modifying	The pop-up prompt message prompt must be filled in	The pop-up prompt message prompt must be filled in	pass
Check the information for deletion	Delete successfully	Delete successfully	pass

6.4 Analysis of Test Results

After repeated tests on different devices for many times, the App has a fast response speed and realizes the function of FragmentTabHost + Fragment bottom navigation bar. The News page uses ViewPager to realize the rolling and cycling broadcast of news pictures and content. The viewed news will be broadcast to all users to display the number of real-time views. The Schedule page realizes the function of regularly refreshing the game live and can be clicked the live video of specific matches (football or basketball) has been achieved, the chat content has been transmitted in real time (including pictures, voices and words), the Agreement of balls has realized the real-time refresh of the ball engagement project and can click in the address bar to jump to the third-party map app navigation, and the Me page has realized the modification of personal information as well as the viewing of registration and initiated ball engagement activities. During this period, app has not broke down, basically realizing the functions designed in function design stage.

7. Conclusions and Future Work

This paper first introduces MQTT technology, Servlet Technology and JDBC technology, and uses the modular idea, C/S communication model, Servlet Technology and JDBC database-access technology to design and implement a Social Sports App with social functions based on MQTT(Message Queuing Telemetry Transport) protocol and efficient interaction between Android and DataBase. The disadvantage of this App is that it takes up

6.3. Test Results

Table 2 Shows Part Of the Test Results, Mainly Extracts the Test of Android side Interaction and Servlet Interaction with Webserver side, and Shows the Test Results.

too much memory. In the process of using, Android, which has a relatively small running memory, is prone to run slowly at the same time as other Apps. In the future work, strengthen the learning of Memory management, Allocating and Reclaiming App Memory and Sharing Memory, and do a good job of APP memory management.

References

- [1] A.P., Haripriya , and K. K. .”Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things,”EURASIP Journal on Wireless Communications and Networking, No.1,2019:90.
- [2] Peng Fengling, Tuo Xianguo, Wang Honghui, Zhang Guiyu, Pu Jianhua, “Design and realization of interactive systems between Android phone and database,” Computer engineering and design, vol.34 No.11 2013,pp.3907-3911.
- [3] Yi D, Binwen F, Xiaoming K, et al..”Design and implementation of mobile health monitoring system based on MQTT protocol,”2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE, 2016.
- [4] Yan Bicheng, Li Feng.,”Communication mechanism between mobile terminal and remote computer,” Computer Engineering, vol.24 No.36 2010,pp.93-95.
- [5] Zhu Minghui,Zhao Xinguang.You Xingyi,”Marine monitoring network framework based on FreeRTOS and MQTT,”Application of Electronic Technique,vol.1 No.44 2018 ,pp.41-44.
- [6] Li GuangMing, Sun YingShuang, Dang XiaoJuan,”Design and implementation of remote monitoring system based on Android,” Computer engineering and design, vol.2 No.37 2016, pp. 556-561.