

Design of Cloud based Instant Messaging System on Android Smartphone using Internet

Shubham Pandey
PG Scholar
Department of IT
SRM University

K. Navin
Assistant Professor
Department of IT
SRM University

G. Vadivu, Ph. D
Professor
Department of IT
SRM University

ABSTRACT

Cloud based instant messaging system for android smartphone with help of internet is specially design for the people to communicate with their families , friends, colleagues.in any organization employees can send and receive messages virtually without face to face contacting; meanwhile the employees can share data and information through instant messaging application. What's more, the staff can talk to customers or vendors online as well, in other word, now people can do business through the instant messenger direct rather than use the traditional method like phone calls. The use of instant messaging nowadays is more than the calling function itself. This paper puts forward the design method of instant messaging system based on SOAP and cloud computing technology for android Smartphone. SOAP interface is an application programming interface of TCP/IP network. Cloud platform is a platform where Cloud Application runs, it is an online application, which works over internet . Cloud Platform has been constructed of three layers: "SaaS", "PaaS", and "IaaS" . In this research work, cloud platform is used for deploying database of instant messaging application.

Keywords

Android ; cloud platform ; Eclipse Juno ; Cloud computing; java system

1. INTRODUCTION

Instant messaging is a set of communication technologies used for text-based communication between two or more participants over the Internet. IM allows effective and efficient communication, allowing immediate receipt of acknowledgment or reply. In the company, colleagues can send and reply instant message in real time without face to face, meanwhile the work report can be shared during the instant chat session; the IM can make a virtual conference without get all the related people together in a physical meeting room. Using instant messages for interoffice communication is quicker than phone calls or emails. More than one person can chat at the same time. This is a huge benefit of using an instant messenger. Instead of relying on a conference call or copying others on an email message, everybody can join and have a discussion in real time. Better than email, if you truly want to communicate instantly you need to consider all your options.Email was the first killer application for the Internet but now instant messaging is coming to cellphones.In this paper development and deployment of the Instant messaging application for android device over cloud platform has been described.[1]

2. CLOUD COMPUTING TECHNOLOGY

2.1.Cloud Computing Principle

Cloud computing is distributed processing, parallel processing and the development of grid computing, or is it the computer science concept of commercial realized.

Cloud calculative fundamental is, through the analysis of the distribution of the distributed computer, rather than the local computer or the remote server, the operation of the enterprise data centre will be more similar as Internet. This makes the enterprise can will resource switch on the application of need, according to demand access to the computer and storage system. This is a kind of revolutionary act, for example, this is just like is the old single generator model turned to the power plant to centralized power supply mode. It means computational ability to also can serve as a kind of goods circulation, as gas, water and electricity, take with convenient, fare is low. The biggest different depend on, it is through the Internet for transmission[2].Cloud computing has the blueprint of be vividly portrayed: in the future, only need a laptop or a mobile phone, you can realize our need through network service everything, even including supercomputing such task. From this perspective, the end user is the cloud calculative true owner. Cloud computing applications include such a new idea or new concept: the power of the world together, to give which every member of the using, as shown in Figure 1 shows.

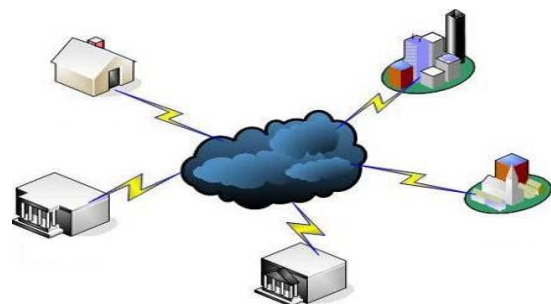


Figure 1. Cloud storage of cloud computing

2.2. "Cloud" Era

At present, the PC is still our daily life of the core tools-we use PC documents, storage material, through E-mail or U plate and share information. If PC hard disk is broken, we will because material loss and stranded.

And in "cloud computing" era (note: the following "cloud computing" are referred to as the "cloud"), "cloud" will be done for us storage and computing work, "cloud" is computer group, and each group includes the thousands of Taiwan, or even millions of computer. "Cloud" benefits are that one of the computers can update, guarantee "cloud" forever. Google is by several such "clouds" of composition, other IT giants such as Microsoft, Yahoo and Amazon also have or are building this "cloud".

At the time, we just need a computer can connect to the Internet, do not need to care about storage or computing happen where a "cloud", but once a need, we can be in any place with any equipment, such as computers, mobile phones, fast calculation and find these material. We will no longer have to worry about material lost.

2.3. The World Only Need 5 Computers

As shown in *Figure 1*, Cloud computing is considered as the science and technology industry of the next revolution, it will bring work methods and business model of fundamental change. First of all, to small and medium enterprises and entrepreneurs for, cloud computing means huge commercial opportunity, they can use cloud computing at a higher level and large enterprise competition. Since the 1989 years since Microsoft introduced version of Office software, the way we work has been great changes, and cloud computing would bring the clouds Office.

3. SOAP KEY TECHNOLOGY

SOAP and can be regarded as two program in one of the communication link endpoint, a program will write a message in the SOAP, the SOAP will this period of information sent to another in the SOAP, it makes this period of information can be transmitted to other applications, its communication

process as shown in *Figure 2* below:

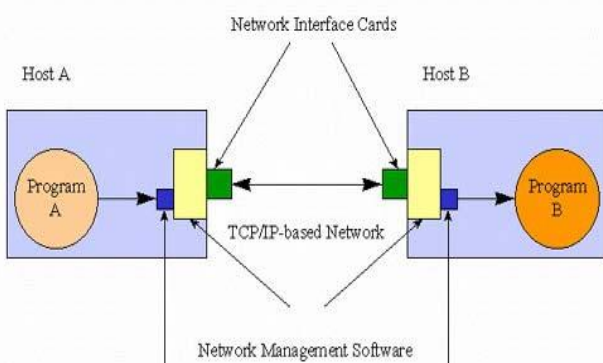


Figure 2. The communication schemes of SOAP

Java furnishes SOAP-type and Server SOAP-type realizing Socket communication, it handles user request and response, it is in bag of java.net Server Socket Used for the server, and it furnishes TCP connection services. SOAP used in establishing network connections, it furnish marriage partner of Communication both ends, connection when success, the application will create an instance of SOAP on both ends, operating this example, to complete the session.

4. ANALYSIS AND DESIGN OF SYSTEM

4.1. The System Definition

The system put completing interface design of system to use the component element AWT and Swing in Java, it build communication in both ends based on Socket port, introducing multithreading technology realize concurrent execution, using JDBC database connection technology to create a database connection[4].

4.2. Functional Requirement

4.2.1. Function of Users Login

Through the fill in the user name and password, choose the right to enter chat room on the interface.

4.2.2. Function of User Registration

Fill in and submit to a server in the personal information, server registered information will become successful after members. Submit personal information, the client to personal information of validation.

4.2.3. Function of Group Chat

user can choose to interface in group chat (chat room the interface) online of all users orspecific users send information, and receive other customer information. Private Chat: The user can chat room facing the world in a particular user sending private chat request, the other party has accepted, both sides will enter private chat interface to private chat[5].

4.2.4. Function of Warning and Kicking

The client and server administrator managers are warning and kicking through the rights to manage chat rooms.

4.3. Overall Design

System structure as shown in *Figure 3* shows. This system is divided into client and server two parts. The client is driven by login, registration, group chat, private chat module four function modules. The server is only one management module. If the server can use cloud computing technology, so the whole system is more efficiency and security.

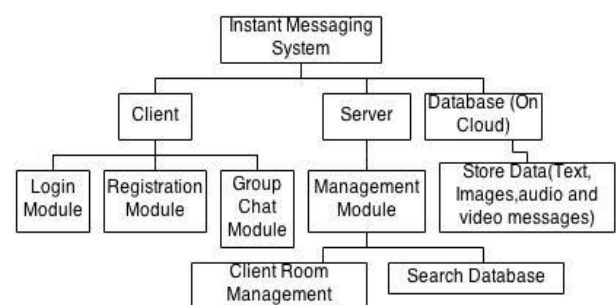


Figure 3. System structure diagram

4.4. Database deployment on Cloud

4.4.1. Database Base Table

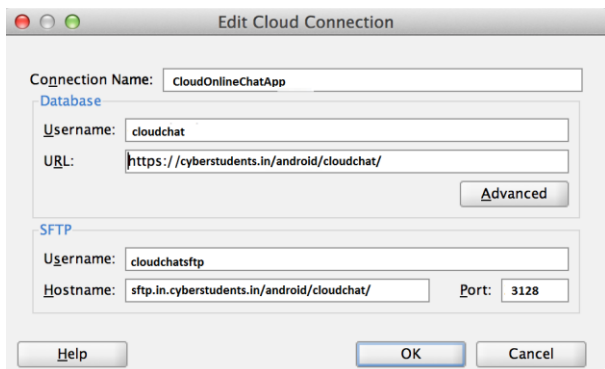
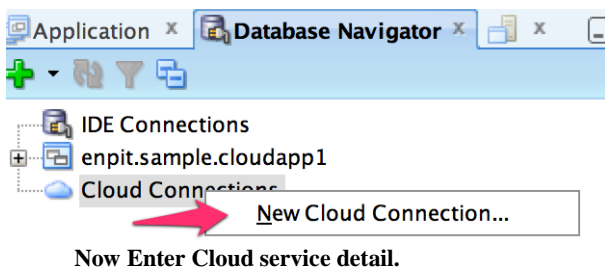
This system of data has two basic form, a user is the basic information table, a basic information table is administrator, field constitute as shown in TABLE I shows[6].

4.4.2. Database Connection to Cloud

To deploy the tables into cloud schema , we have to open Eclipse Juno and click on window button then click on preferences then select data management and then click on driver definition . in open new window of driver definition select Oracle XE database. Now just click on “+” and add a database connection by account.

Double click on created node connect to the database and explore it contents.

Next to connect to the oracle cloud database service.[6]



Now the cloud connection is established.

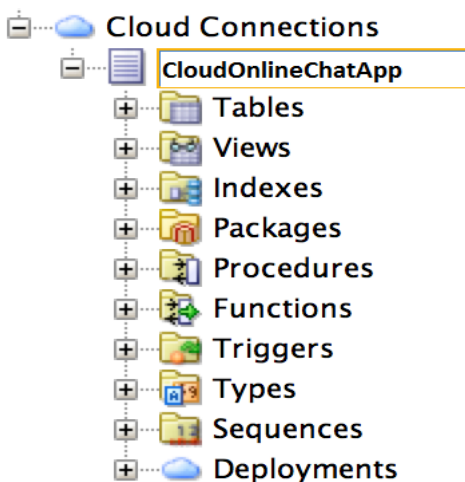


TABLE 1. USERINFOR

Field	TypeName	Description	For null
userID	Int	user number	not null
Username	varchar(20)	user name	not null
userPwd	varchar(20)	user password	not null
Email	varchar(20)	User Email	not null
Telephone	varchar(20)	contact way	not null

Use name for: shubham
Pass word for: 123456

4.5. Detailed Design

4.5.1. Login Module Design

This module will need a graphical user interface, for the user to enter your user name and password, and select login role.

4.5.2. Registered Module Design

This module will need a graphical user interface, for user input personal information, need special mark required fields, and to the input information for verification[7].

4.5.3. Chat Module Design

This module will need two graphical user interface, the world face and a private chat room chat interfaces, user chat of the main places. The user can accord to the personal like, selection of the group chat or secret private chat.

4.5.4. The Server Management Module Design

This module will need a graphical user interface, running on the server management of start-up and closed Socket, forward user chat messages, and the management of the chat rooms.

5. THE OVERALL DESIGN OF THE ANDROID INSTANT MESSAGING SYSTEM

5.1. Platform development for designing Process

For developing the Platform for the application designing, the following steps used to be followed out one by one:

For running Eclipse successfully installed JDK6.

Then downloaded and installed Eclipse Juno, version "eclipse-Juno-SR2-win32" in the system. As Eclipse can be downloaded free of cost from its website.

Then Android SDK 4.1 and ADT-20.0.0 had been installed in the Eclipse Juno for making it compatible with Android devices i.e. Phones, Emulator.

After the installation of all these software and plug-in packages, the platform for the designing of the cloud based application become ready.

5.2. Client Design

In order to perform chatting and data sharing between several user we have to develop all client modules as like login module, registration module, profile module and chatting module than only communication will be possible.

5.2.1. User registration

in user registration process firstly user has to register itself then only user can join the chat room and perform chatting and data sharing.[7]

The important API function used for coding user registration is as such:

```
package csp.ela.project.cloud.chat;
```

```
btn_signup.setOnClickListener(new  
OnClickListener()
```

```
{ @Override
```

```
public void onClick(View v)
```

```
{
```

```
if(checkInternetConnection())
```

```
{
```

```
if(!et_name.getText().toString().equals("")
```

```
&&!et_email.getText().toString().equals("")
```

```
&&!et_pass.getText().toString().equals("")
```

```
&&!et_cpass.getText().toString().equals("")
```

```
&&!et_age.getText().toString().equals("")
```

```
&&!et_phone.getText().toString().equals(""))
```

```
{
```

```
if(et_pass.getText().toString().equals(et_c  
pass.getText().toString()))
```

```
{ usrReg_thread();
```

```
}
```

```
Else
```

```
{
```

```
et_pass.setText("");
```

```
et_cpass.setText("");
```

```
Toast.makeText(getApplicationContext(),
```

```
"Password not match ...!",
```

```
Toast.LENGTH_SHORT).show();
```

```
}}
```

```
else
```

```
{
```

```
Toast.makeText(getApplicationContext(),
```

```
"Field Empty ...!",
```

```
Toast.LENGTH_SHORT).show();
```

```
}
```

```
}
```

```
else
```

```
{
```

```
Toast.makeText(getApplicationContext(),
```

```
"No Network Connection ",
```

```
Toast.LENGTH_LONG).show();
```

```
}
```

```
}
```

```
});
```

```
}
```

```
private boolean checkInternetConnection()
```

```
{
```

```
ConnectivityManager=ConnectivityManager)
```

```
getSystemService(Context.CONNECTIVITY_SERVI  
CE);
```

```
// test for connection
```

```
if (cm.getActiveNetworkInfo() != null
```

```
&& cm.getActiveNetworkInfo().isAvailable()
```

```
&& cm.getActiveNetworkInfo().isConnected())
```

```
{
```

```
return true;
```

```
}
```

```
Else {
```

```
//Toast.makeText(getApplicationContext(),
```

```
"No Internt Connection ",
```

```
Toast.LENGTH_LONG).show();
```

```
return false;
```

```
}
```

```
}
```

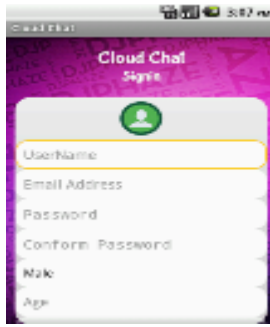


Figure 4: Registration Module

5.2.2. Login Activity

In login activity user should fill all relevant field by using same information that used at registration time. The important API function used for coding login activity is such as:

```
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.transport.HttpTransportSE;
btn_signin.setOnClickListener(new OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        if(!et_email.getText().toString().equals("")
        && !et_pass.getText().toString().equals(""))
        {
            AsyncCallWS task = new AsyncCallWS();
            task.execute();
        }
        else
        {
            Toast.makeText(getApplicationContext(),"Do not
            empty email and password",
            Toast.LENGTH_SHORT).show();
        }
    }
});

btn_signup.setOnClickListener(new
OnClickListener()
```

```
{
    @Override
        public void onClick(View v)
    {
        Intent intobj=new Intent(HomeActivity.this,
        RegisterActivity.class);

        startActivity(intobj); // Login Activity Start
        //finish();
    }
});

String res_info = result.getProperty(0).toString();

final String res_uid =
result.getProperty(1).toString();

if(res_info.equals("Login Success"))
{
        handler.post(new Runnable()
        {
            @Override
            public void run()
            { if (simpleWaitDialog != null)
                { simpleWaitDialog.dismiss();
                }

            Intent iobec=new Intent(HomeActivity.this,
            UserActivity.class);

            iobj.putExtra("id", res_uid);

            startActivity(iobj);

            Toast.makeText(getApplicationContext(),
            "Login Success ...!",
            Toast.LENGTH_SHORT).show();
        }
    });
    //finish();
}
```

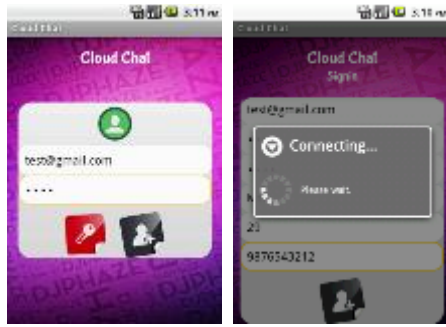


Figure 5: Login Module

5.2.3 Profile Management Activity

In profile management activity user can manage his profile information such as change his profile photo and editing other information which is related to his profile. The important API function used for coding login activity is such as:

```
package csp.ela.project.cloud.chat;

import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import
org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

//Showing temporary image as profile picture
private static final String TEMP_PHOTO_FILE =
"temporary_holder.jpg";
private static final int REQ_CODE_PICK_IMAGE =
2;

private File getTempFile()
{
    if
(Environment.getExternalStorageState().equals(Envir
onment.MEDIA_MOUNTED))
{
        File file = new
File(Environment.getExternalStorageDirectory(),
TEMP_PHOTO_FILE);
        try
        {
            file.createNewFile();
        } catch (IOException e)
        {
            e.printStackTrace();
        }
        return file;
    }
    else
    {
        return null;
    }
}

protected void onActivityResult(int
requestCode, int resultCode, Intent
imageReturnedIntent)
```

```
{
    super.onActivityResult(requestCode,
resultCode, imageReturnedIntent);
    switch (requestCode)
    {
        case REQ_CODE_PICK_IMAGE:
            if (resultCode ==
RESULT_OK)
            {
                if (imageReturnedIntent != null)
                {
                    File tempFile = getTempFile();
                    String filePath =
Environment.getExternalStorageDirectory() + "/" +
TEMP_PHOTO_FILE;
                    System.out.println("path " + filePath);
                    BitmapselectedImage =
BitmapFactory.decodeFile(filePath);
                    //_image =
(ImageView) findViewById(R.id.imageView1);

                    //_image.setImageBitmap(selectedImage);
                    /*
                    if (tempFile.exists())
                    tempFile.delete();
                    */
                }
            }
    }
}

private class AsyncCallWS extends
AsyncTask<Void, Void, Void> {
    @Override
    protected Void doInBackground(Void...
params) {
        Log.i(TAG, "doInBackground");
        call();
        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        Log.i(TAG, "onPostExecute");
        if (simpleWaitDialog != null)
        {
            simpleWaitDialog.dismiss();
        }
    }

    @Override
    protected void onPreExecute() {
        Log.i(TAG, "onPreExecute");
        simpleWaitDialog
ProgressDialog.show(ProfileActivity.this, "Wait",
"Connecting...");
    }

    @Override
    protected void onProgressUpdate(Void... values)
    {
        Log.i(TAG, "onProgressUpdate");
    }
}
```



Figure : Profile Management Module

5.3. Designing of Server

When you want to connect two android devices , one must as a server (by holding an open socket for communication) , which is designed to listen for incoming connection requests. When client get any socket request from serversocket can to been destructed , unless you want to listen for more connection requests. Establish the service socket and listening for connections, the basic coding is as follows:

```
import java.net.InetAddress;
import java.net.NetworkInterface;
public class ChatServer extends Activity
{
    private String var_port;
    String SERVERIP = "10.0.2.15";
    SERVERIP = getLocalIpAddress();
    Status.setText("Listening on IP: " + SERVERIP);

    public void onClick(View v)
    {
        // Perform action on click
        if (Port != null && Port.getText().length() != 0)
        {
            var_port = Port.getText().toString();
            intent intObj = new Intent();
            intObj.setClassName("cloudproject", "Cloudproject.Server");
            intObj.putExtra("port", var_port);
            intObj.putExtra("ip", SERVERIP);
            startActivity(intObj);
            finish();
        } else
        {
            Toast.makeText(getApplicationContext(),
            "Please Enter Port", Toast.LENGTH_LONG).show();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        // Inflate the menu; this adds items to the action bar if it is
        present.
        getMenuInflater().inflate(R.menu.activity_chat_server, menu);
        return true;
    }
}
```

```
// Gets the ip address of your phone's network
public String getLocalIpAddress() {
try {
for (Enumeration<NetworkInterface> en =
NetworkInterface.getNetworkInterfaces();

NetworkInterface intf = en.nextElement();
for (Enumeration<InetAddress> enumIpAddr =

intf.getInetAddresses(); enumIpAddr.hasMoreElements();
{
InetAddress inetAddress = enumIpAddr.nextElement();
if (!inetAddress.isLoopbackAddress()) {
return
inetAddress.getHostAddress().toString();
}
}
}
```

5.4. Messaging System

The messaging system is used to be coded like that the messaging has to be done between many android devices. The messaging system is coded using Java language:

5.4.1. Sending chat message

Once you have an active chat session , use the sendchatmessage method to send message to the users in that session, as shown in the following code snippet:[8]

```
chatsession.chat.sendchatmessage("hello word");
```

5.4.2. Recieving Chat Message

To listen incoming message we have to implement ChatListner interface.

```
ChatListener chatListener = new ChatListener.Stub()
{
public void newMessageReceived(String from, String body)
{
// TODO Handle incoming messages.
}
// Required group chat implementation stubs.
public void convertedToGroupChat(String oldJid,
String groupChatRoom,
long groupId) {}
public void participantJoined(String groupChatRoom, String
nickname) {}
public void participantLeft(String groupChatRoom, String
nickname) {}
public void chatClosed(String groupChatRoom) throws
RemoteException {}
public void chatRead(String arg0) throws RemoteException {}
}
```

```
}; // Add Chat Listener to the chat session.
chatSession.addRemoteChatListener(chatListener);

// Add Chat Listener to the instant messaging session
.imSession.addRemoteChatListener(chatListener);
```

5.4.3. Transmitting Data Message

For transmitting data message as like images audio and video we use broadcasting an Intent to a remote device, such as the one shown in the snippet below:

```
public static final String ACTION_OTA_ELIMINATE =
"com.paad.ota_eliminate_action";
```

The next code shows how to create a simple Intent that will be packaged within a data message to transmit the above action to a remote device:

```
Intent intent = new Intent(ACTION_OTA_ELIMINATE);
Send the message using the sendDataMessage method,
passing in the target username and the Intent to broadcast.
The sendDataMessage is available on IM Session or Chat
Session objects, as shown below:
```

```
String username = "test@gmail.com";
```

```
// Send to target user.
```

```
imSession.sendDataMessage(username, intent);
```

```
// Send to all chat room participants.
```

```
chatSession.sendDataMessage(intent);
```

6. CONCLUSION

This paper presents an idea to develop an android instant messaging application for the internet users to send and receive messages and data. Now the application is working properly on android devices with using cloud as data storage. But this application only works for over android devices, it has to be design for windows, still need to further improve the usability and functionality of the system.

7. REFERENCES

- [1] Entao WANG, Xiang LI., The communication research of mobile phones and database server based on Socket. Technology and Development of Computer, vo17. 2, pp.82-84, 2007 (In Chinese).
- [2] Doc88.com, Cloud Computing of New Techniques of Computer Network. pp1-4, September, 2010
- [3] Cloudcomputing-China, china.cn/, Cloud Computing Knowledge, March, 2012
- [4] Jiahui HUANG, Java Network Program Design. Beijing: Tsinghua University Press, pp.30-49, 2002
- [5] Hau WANG, Ailiang YE, Lixeu QI, Visual C++6.0 programming Examples and skills.
- [6] Oracle Database developers-Advanced database training. <http://multikoop.blogspot.in/2012/12/deploying-database-objects-to-oracle.html>.
- [7] Android Developers Advanced Training <http://developer.android.com/training/cloudsync.html>.
- [8] Reto Meier, "Professional Android Application Development" Wrox publication.