

XMPP based communication application (campus connect)

Venugopal Reddy M ¹, Srinidhi P Katte ², Yuvaraju B N ³

¹Department of CSE, The National Institute of Engineering, Mysuru.

²Department of CSE, The National Institute of Engineering, Mysuru.

³Department of CSE, The National Institute of Engineering, Mysuru.

Abstract - "Campus Connect" is an android application uses XMPP protocol for communication. XMPP stands for Extensible Messaging and Presence Protocol. It is a set of open-source technologies for instant messaging, presence, multi-party chat and generalized routing of XML data [1]. XMPP protocol allows client to exchange data with server in the form of xml. In this application we have used Xabber, which is a XMPP open-source server [2]. This android application is developed for the university use to share several kinds of files (pdf, text, docx). It also helps the college admin to inform students or lecturers about the news or events that the college is going to be organized by broadcasting news on the college timeline page. This application also enables students and faculty to directly communicate with one another. This application also allows lecturers to inform students about the course details, class schedule etc.

Key Words: XMPP, Android application, File sharing, Messaging application.

1. INTRODUCTION

This application provides a way for one-to-one communication between the students and lecturers. The users of this application are students, admin and the staffs of an academic institution. This application is useful when there is a need for lecturer to send files to group of students, files could be pdf, docx, mp4 etc. Instead of emailing each student, he can just put it in multichat room of XMPP [3]. Smartphone have become more affordable and portable nowadays. Users can manage their day to day tasks, email, communicate with others through smartphones. They are nearly as powerful as desktop computers. It would be great for academic institute to have an android application which is one place to manage, publish news, communicate and share materials to students irrespective of their role. The main objective of this paper to explain how we can use features of XMPP in our android application – "campus connect".

2. APPLICATION FEATURES

The proposed system consists of several features for students, lecturers and admin. This mobile application for your institution will make give a better experience for students. The faculty and student will benefit from a college application in many ways. Regular notification and

campus news on the timeline page is displayed. The user can share several attachments (files, images, video and audio) to others in one touch. This android application can be used by the following,

A. College admin-

- Can post important news about the college.
- Can post about college events like annual sports, exam schedule, faculty-parents meeting etc.

B. Department head-

- Can share the information with the faculty members.
- Can directly communicate with the students about various department activities.

C. Lectures-

- Can inform students directly about the classes and syllabus etc.
- Can share the course materials with the students easily.
- Can directly communicate with any student.

D. Students-

- Can get to know about the college events.
- Can receive course materials from the lectures.
- Can communicate with other students or faculty members directly.

Notifications will also be sent to the student whenever college posts anything on the timeline or when any faculty shares the course materials.

To summarize our application includes the following features

A. College timeline page, where one can see important news and events of college.

B. One to one messaging and group messaging.

C. File sharing.

D. Notifications about college events and whenever lecturer shares course materials.

3. SYSTEM DESIGN

3.1 XMPP OVERVIEW

XMPP uses Decentralized Client Server architecture. The idea behind the XMPP is using xml data to exchange between client and server. Most of the complex operations are handled by server like authentication, encryption, maintaining online presence status, maintaining contact list etc. When client requires information, it just sends query request in the form of xml stanzas. The server reply the response in xml form. The XMPP application architecture is as shown in the figure below.

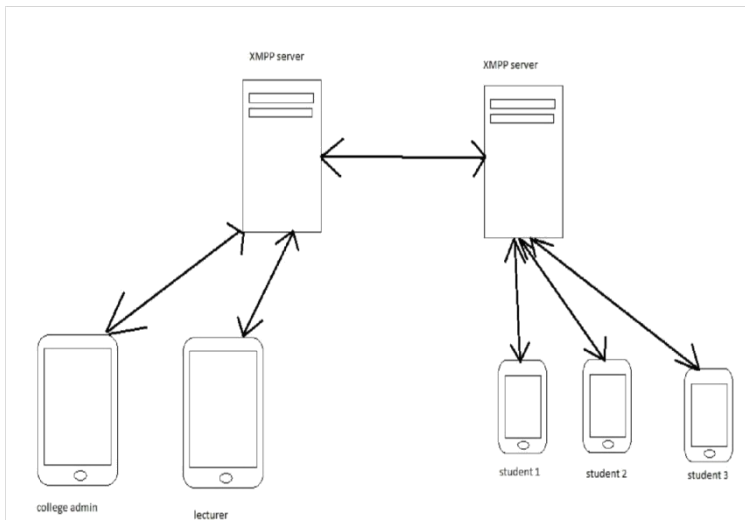


Fig-1:XMPP architecture

Since it uses decentralized architecture ,separation of concerns is easy-client developers concentrate on user experience ,and server developers can focus on reliability and scalability .XMPP servers are federated which means when you send an XMPP message to one of your contacts at a different domain ,your client connects to your home server which then connects directly to your contact’s server without intermediate hops [4] .Like email, XMPP clients have unique addresses .The format of XMPP client address is “user@server.com”, where “user” is username and “server.com” is the name of XMPP server name. A address in XMPP is refered as Jabber ID abbreviated JID. Basic unit of communication in XMPP is called as “stanza”. There are three types of stanzas- Message stanza , an IQ stanza and a Presence stanza. Each of these stanzas is handled differently by XMPP servers and clients.

3.2 COLLEGE TIMELINE PAGE AND NOTIFICATIONS

In our android application ,the college admin can post college news and events on the timeline page, all the users of application can view these news and events on the college timeline page as soon as they login to the application. As soon as the college admin posts the college news or events, the information will be pushed to the database and then it is retrieved from the database and displayed on the timeline page of the college which is accessible to all users once they login to the application. In our application ,users will get notifications whenever there is new college events and when user gets new messages and files. We can send notifications to users using cloud notification platform like firebase[5] .

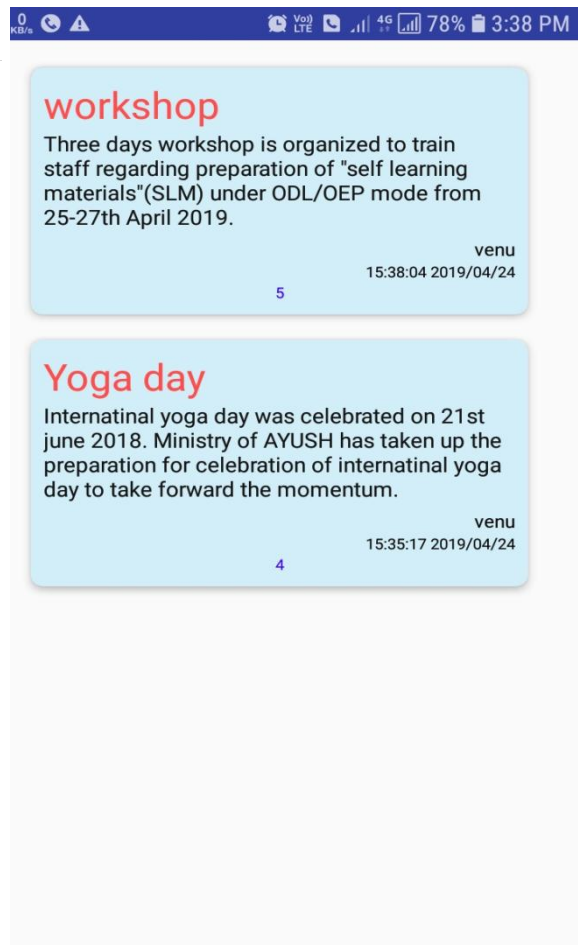


Fig-2:College timeline page

3.3 ONE TO ONE MESSAGING AND MULTICHAT MESSAGING

In XMPP, “roster” means contact list of the user. A user contact list will be stored on XMPP server. If your JabberID is “role1@xabber.org”, your contact list is stored on the XMPP server with the domain name “xabber.org”[6] . To add a new contact to the roster, user sends a roster set IQ stanza to the XMPP server, with the new roster entry inside. Even if the device is lost, the user can retrieve the contact list from XMPP server. For our android application we can automate this process, we just use a button click ,all the contacts of the institution will be fetched from college database and be updated to our contact list. A XMPP <message/> stanza uses push method for getting information, there is no acknowledgement received for messages sent from XMPP server. XMPP uses “fire-and forget” mechanism. For example lets say user “role1@xabber.org” sends a message to “role2@xabber.org” [7]. The xml data sent to the XMPP server is shown as below.

```
<message from="role1@xabber.org/android"
to="role2@xabber.org"
type="chat">
<body>Hi,good afternoon</body>
<subject>Query</subject>
</message>
```

Our android application uses multi-chat room features of XMPP to broadcast message to group of students [8]. Suppose when the lecturer needs to inform a group of students, he can just send a message to multi-chat room which will be broadcasted to all students. Same features can be used to share course materials to students. Below is the example xml data sent to multichat room in XMPP.

```
<message from="role1@xabber.org/android"
to="1styearstudents @xabber.org"
type="groupchat">
<body>Todays class is at 10 AM. </body>
<subject>Query</subject>
</message>
```

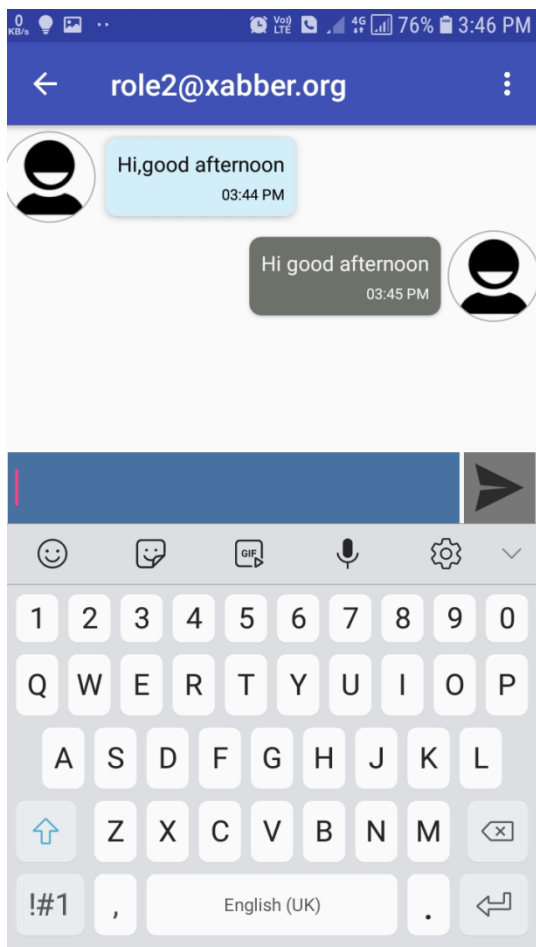


Fig-3:XMPP Messaging

3.4 FILE SHARING

In our android application lecturer can share the course materials with a selected group of students. Lecturer can select the file to be sent from his/her local file system and then he will be provided an option to choose semester and section of students to whom the course materials needs to be sent. The FIG-4 shows the various steps involved in XMPP file sharing.

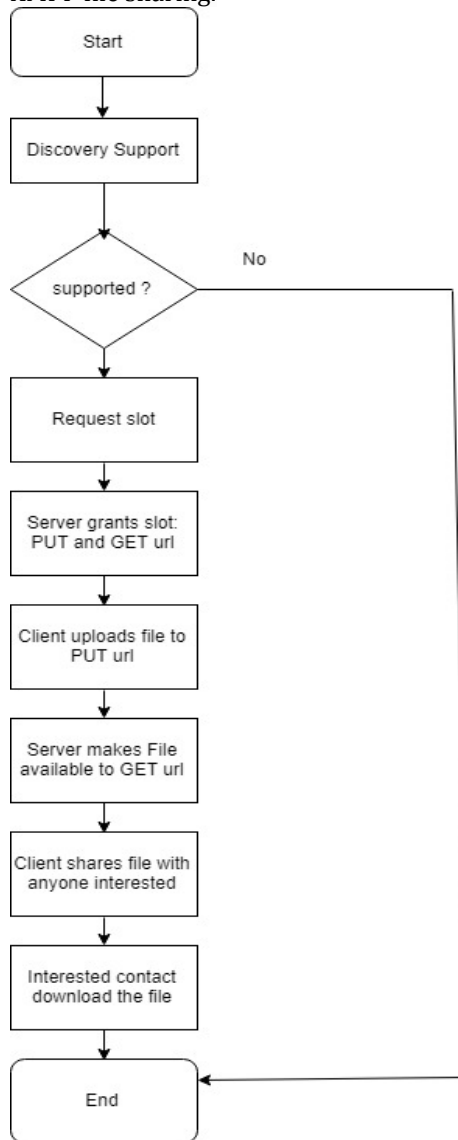


FIG-4: FLOW CHART FOR XMPP FILE SHARING

XMPP File transfer happens only if XMPP servers support HTTP protocol [9]. Let us understand the various steps involved in XMPP File Transfer mechanism through flowchart mentioned above. Initially the client ask the XMPP server if it supports HTTP protocol. It does this by sending IQ stanza mention below.

```
<iq from='role1@xabber.org/device1'
  id='s1'
  to='xabber.org'
  type='get'>
  <query
  xmlns='http://jabber.org/protocol/disco#items' />
</iq>
```

This stanza queries the features supported by XMPP. XMPP server replies with the list of all the features it supports with IQ stanza shown below.

```
<iq from='xabber.org'
  id='s1'
  to='role1@xabber.org/device1'
  type='result'>
  <query
  xmlns='http://jabber.org/protocol/disco#items'>
  <item jid='upload.xabber.org'
  name='HTTP File Upload' />
  <item jid='conference.xabber.org'
  name='Chatroom Service' />
  </query>
</iq>
```

If the reply contains 'HTTP File upload', then we know the server supports HTTP protocol. If not, file transfer is not possible. If XMPP server supports 'HTTP File Upload', then we get more information about this feature by sending IQ stanza as shown below.

```
<iq from='role1@xabber.org/device1'
  id='s2'
  to='upload.xabber.org'
  type='get'>
  <query
  xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

then XMPP server replies with IQ stanza shown below. This contains maximum size of file upload it supports in bytes. If we try to send file of size more than this, we get error message from XMPP server.

```
<iq from='upload.xabber.org'
  id='s2'
  to='role1@xabber.org/device1'
  type='result'>
  <query
  xmlns='http://jabber.org/protocol/disco#info'>
  <identity category='store'
  type='file'
  name='HTTP File Upload' />
  <feature var='urn:xmpp:http:upload:0' />
  <x type='result' xmlns='jabber:x:data'>
  <field var='FORM_TYPE' type='hidden'>
  <value>urn:xmpp:http:upload:0</value>
  </field>
  <field var='max-file-size'>
  <value>5242880</value>
  </field>
  </x>
```

```
</query>
</iq>
```

Then we request slot from xmpp server to obtain PUT and GET urls. Using PUT url we upload file using HTTP and using GET url we can download the file from server. The IQ stanza to get slot as shown below.

```
<iq from='role1@xabber.org/device1'
  id='s3'
  to='upload.xabber.org'
  type='get'>
  <request xmlns='urn:xmpp:http:upload:0'
  filename='picture1.jpg'
  size='43568'
  content-type='image/jpeg' />
</iq>
```

Here we specify file name and file size in bytes. After this the XMPP server replies with PUT and GET urls as shown below.

```
<iq from='upload.xabber.org'
  id='s3'
  to='role1@xabber.org/device1'
  type='result'>
  <slot xmlns='urn:xmpp:http:upload:0'>
  <put url='https://upload.xabber.org/4f581ac1-8gb2-4f4c-9700-f45a2451nvcqbb67/gr%C3%A9%20cool.jpg'>
  <header name='Authorization'>Basic
  Base64String==</header>
  <header name='Cookie'>foo=bar;
  user=role1</header>
  </put>
  <get url='https://upload.xabber.org/4D673ac3-f0b&-4k4z-9700-d2a58fa2bb67/tz%D3%R8s%20cool.jpg' />
  </slot>
</iq>
```

client upload a file to PUT url. Client shares the GET url to those who want to download the file. One security aspect of this is who can download the file. Anyone with the GET url, but the access to the file is in the hands of client who uploads the file. Without client sharing this GET url, it is almost impossible to guess the GET url as it is cryptic path. As we can in the above IQ stanza it is long url with random letters and digits. In our android application we can hide this url from client and show only the file from device which is uploaded.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The entire application is developed using Android studio IDE and is tested on various versions of Android. Java is the programming language used. We have used Xabber-XMPP server. The database required to store user information, messages and files can be deployed in Linux VM provided by Amazon Web Services. The FIG-5 shows the control flow diagram for our Android application.

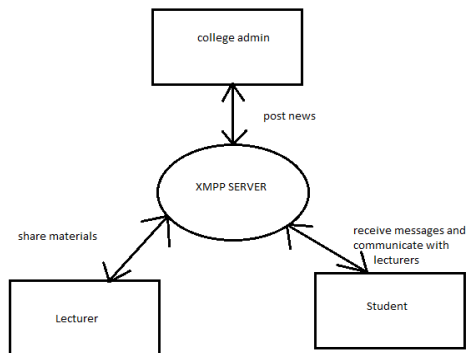


FIG -5: CONTROL FLOW DIAGRAM OF OUR “CAMPUS CONNECT” APPLICATION

CONCLUSION

The proposed system uses various features of xmpp like instant messaging ,multichat room ,online roster ,file upload. It makes it easier to manage activities of university in one place .The number of users of smartphone is increasing every day. The regular email services lacks some of the features explained above .By building custom application for the campus, it will be easier for the students to use a wide range of features of XMPP.

REFERENCES

- [1] <https://xmpp.org/about/technology-overview.html>
- [2] <https://www.xabber.com/>
- [3] <https://xmpp.org/extensions/xep-0045.html>
- [4] <https://www.cisco.com/c/en/us/support/docs/unified-communications/expressway/213865-configure-and-troubleshoot-xmpp-federati.html>
- [5] <https://firebase.google.com/>
- [6] <https://xmpp.org/rfcs/rfc6121.html>
- [7] <https://xmpp.org/uses/instant-messaging.html>
- [8] <https://xmpp.org/extensions/xep-0045.html>
- [9] <https://xmpp.org/extensions/xep-0363.html>