International Conference on Computer Science and Computational Intelligence (ICCSCI 2015)

Architecture and Implementation of Instant Messaging in Educational Institution

Budi Yulianto, Eileen Heriyanni, Lusiana Citra Dewi, and Timothy Yudi Adinugroho*

Bina Nusantara University, Jl. K.H. Syahdan No. 9, Jakarta 11480, Indonesia

Abstract

Development of communication technology on smartphones can be used to meet the needs of actual mass communication (real time chat), concise, and efficient in educational institutions, especially for students and lecturers. This research aims to produce architectural design and implementation of web and Android based instant messaging (IM) which can be applied to educational institutions as a fast, provident, and effective communication means. This study was conducted by collecting data through questionnaires distributed among the students and the lecturers. The IM application is developed based on Rational Unified Process (RUP) software development method. The architecture and IM application are XMPP protocol based that can be implemented on web and Android platform with personal chat, group chat, broadcast messages, scheduled messages, and file attachments features. The application can facilitate the communication from an educational institution to or between students and lecturers, and gather students and lecturers’ phone number and email.

Keywords: instant messaging, real-time chat, xmpp, personal chat, broadcast message, scheduled message

1. Introduction

Instant messaging (IM) becomes important in the world of communication for supporting practicality and speed in real time chat which is not facilitated by e-mail1,2. Conversation via IM lasted 20 seconds in average, thus making this communication technology has become an efficient means to get a quick respond from a chat3. With the

* Corresponding author. Tel.: +62-21-5345830; fax: +62-21-5300244.
E-mail address: budi.yulianto@binus.edu;lcdewi@binus.edu; eheriyanni@binus.edu; tadinugroho@binus.edu
presence-awareness technology that can monitor whether a user is online or not, IM can improve the ability to respond faster than e-mail. In addition, when compared with a phone call, using IM is cheaper and considered less intrusive. Actualization and file sharing needs will increase in line with the number of IM users which already more than the number of e-mail users. Online collaboration through instant messaging is also easier because the interaction takes place within one working window (chat history can be seen), and low-bandwidth because the previous message is not attached as email reply as usual. According to the research conducted in Korea, the use of IM can also improve working relationship in a company.

The pattern of short and efficient mass communication such as IM is also needed for an educational institution. The role of communication is when the educational institution wants to deliver important information to students, lecturers, and staffs. In addition, the institution also want to deliver information to the student's parents and outside community. Information such as registration, payment, lectures, catastrophe, exam schedules, schedule change, and so on often hampered when sent through e-mail because of not immediately legible and limited quota of e-mail recipients. Moreover, the information via SMS is also having problems because they often change their phone number then difficult to contact. IM applications are expected to maintain contact information with the single account (user only has one account) even though the users change their phone number and e-mail.

State of the art of this research is to design and develop the architecture and IM application which can be used as a communication tool by educational institutions. Communication via IM has been used as an interaction tool between students and lecturers outside the class in some universities across the world. Communication like this can establish a close relationship between students and lecturers, create a sense of interconnected between the students with the class and university, reduce students' anxiety in the class, save time and students' effort, as well as add a pleasant atmosphere inside the class. Therefore, the IM application can be used as a solution to establish communication inside and outside the class and be able to help lectures activities. Previous researches had done developing instant messaging architecture (some for educational institution) to support only personal chat feature, without group chat feature to support communication from a lecturer to students in a class, scheduled message for reminder (assignment, exam, etc), file attachment for sharing, and collect students contact (phone number or e-mail) for urgency needs.

2. Proposed Application System

The developed IM application will implement push notification technology that runs on web platform and Android based smartphones. IDC research shows that the smartphone operating system market share in the second quarter of 2013 was dominated by Android at 69.1%. The distributed questionnaires to 122 respondents (students and lecturers) by researcher showed that 55% of respondents are using Android 4.x, and 33% of respondents are using BlackBerry 5-7. Not only developed on Android platform, the application will be also developed on Web platform that can be accessed while using the PC / laptop because of 72% of respondents wanted to. Furthermore, 78% of respondents are using LINE, then BBM (70%), and WhatsApp (66%). 87% of respondents are often using emoticon, then contact group (76%), file attachment (74%), group-chat (72%), and broadcast message (66%). These features will become a reference for IM application development.

The results of this study is an IM application with personal chat, group chat, broadcast messages, scheduled messages, integrated file attachment with GoogleDrive and OneDrive features. The expected benefit of this application is facilitating fast, cheap, and effective communication in educational institutions, among institutions, students, and lecturers.

3. Application Development Methodology

The application development methodology used in this research is Rational Unified Process (RUP) with 4 phase: inception, elaboration, construction, and transition. In the inception phase, scope definition and collection requirements were done by distributing questionnaire to students and faculties, and doing interview. In the second
phase, remaining requirements were collected and application system was modelled by using Unified Modelling Language (UML), such as Use Case Diagram (Fig. 1). The third phase carried out the development of non-critical features such as group chat, scheduled messages, and broadcast messages. In this phase also made implementation planning, testing and bug fixing, and user manual documents updating. In the fourth phase, implementation of the system was done then released the application to the user for early beta testing.

This IM application consists of two parts, front-end and back-end as described in Use Case Diagram (Fig. 1). Front-end side is used by the user (consists of students, lecturers, staffs, parents, and institution's associated communities) and developed on Android platform (Fig. 2) and web (Fig. 3).
Back-end side is used by the admin and super admin representing educational institutions for sending a broadcast message to the user and communities for administrative purposes and only developed on web platform.

Some obstacle during development is restricted OneDrive API to be implemented in Android platform. Limited articles of OneDrive API has hampered development process. Trial and error method is used during development to solve the problem. Other obstacles are bandwidth management for contact presence status, and loss of push messages when instable internet network happened.

4. Proposed Architecture Model

The Android IM application will be connected via XMPP protocol to a chat server that has been implemented with Openfire (Fig. 4). Web Server is used for implementation of web application and can communicate via XMPP protocol with chat server and MySQL database. Web server can also communicate with Android application through Web Service27.
Openfire is a Java based open-source application for chatting that implements Extensible Messaging and Presence Protocol (XMPP). Openfire can be implemented in web server on port 9090 (HTTP) or 9091 (HTTPS) with Windows, Linux, and Macintosh operating systems. XMPP is a network protocol that provides capability of real-time communication, such as for instant messaging, voice and video conference, real-time game, data synchronization, and XML searching. XMPP server and client communicate each other by sending a XML called stanza. Stanza XML structure is defined in 3 semantic elements: `<message/>`, `<presence/>`, and `<iq/>` (Fig. 5).

```
<stc://>

<presence>
  <affiliation type="colleague"/>
</presence>

<message to="zoo">
  <body/>
</message>

<iq to="bar">
  <query/>
</iq>

...

</stc://>
```

5. System Specification

The architecture design and back-end application for chat and web server are implemented on a server with the following specifications:

- Intel Core i7 3770 3.4 GHz Processor
- 2 TB hard drives
- 8 GB RAM
- Internet network connected
- Windows Server 2012 R2 operating system
- IIS version 6.2 web server
- 3.8.2 Openfire XMPP server version with Broadcast, Search, OpenArchive Plugin

The client application is implemented on S6310 Samsung Galaxy Young smartphone device with the following
specifications:
• 4.0 (Ice Cream Sandwich) Android operating system version
• MSM7227A Snapdragon 1 GHz Qualcomm processor
• 320 x 480 pixels touch-screen resolution
• 512 MB RAM
• 802.11 b/g/n Wi-Fi or mobile networks (EDGE, with speeds around 384 Kb/s)

Meanwhile the client computer used to run web platform has the following specifications:
• 1.9 GHz Intel Pentium P600 Processor
• 2GB RAM
• Last updated browser (Google Chrome 32 version / Firefox 20 version / Internet Explorer 11)
• Adobe Flash 11 version browser plugin

6. Experimental Results

The trial of application through performance test on Android is done by using DDMS debugging tool or Dalvik Debug Monitor Server. In the diagram shown that the application uses 7% of CPU resources (Fig. 6). The use of memory is 21 MB with one XMPPService that runs automatically (Fig. 7).

Web application performance analysis using the YSlow extension on Google Chrome 32.0.17 version browser.
The trial of application is carried out on 6 aspects of web application performance in general, like content, cookies, CSS & JavaScript, images, and server. The application assessment criteria using A (good) - F (poor) scale grade based on established standardized best practices (Table 1-5).

Table 1. Trial of Content

<table>
<thead>
<tr>
<th>No</th>
<th>Performance Criteria</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of HTTP request</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>Number of DOM element</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>Number of URL redirect</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>AJAX cache</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>Avoiding HTTP error 404</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>Number of DNS lookup</td>
<td>A</td>
</tr>
</tbody>
</table>

Table 2. Trial of Cookie

<table>
<thead>
<tr>
<th>No</th>
<th>Performance Criteria</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Size of cookie</td>
<td>A</td>
</tr>
</tbody>
</table>

Table 3. Trial of CSS & JavaScript

<table>
<thead>
<tr>
<th>No</th>
<th>Performance Criteria</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CSS expression</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>Minify CSS &amp; JavaScript</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>Laying CSS</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>Laying JavaScript</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>CSS &amp; JavaScript redundancy</td>
<td>A</td>
</tr>
</tbody>
</table>

Table 4. Trial of Image

<table>
<thead>
<tr>
<th>No</th>
<th>Performance Criteria</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Image Scaling in HTML</td>
<td>A</td>
</tr>
</tbody>
</table>

Table 5. Trial of Server

<table>
<thead>
<tr>
<th>No</th>
<th>Performance Criteria</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use of entity tags (E-tags)</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>Use of GET for AJAX request</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>Expires headers</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>Empty src or href</td>
<td>A</td>
</tr>
</tbody>
</table>

Based on the trial that have been conducted on several performance aspects in general, the web application gets rank B with overall performance score 85 (Fig. 8).

Figure 8. Overall Trial Result of Web Application

7. Conclusion

IM has been implemented in Bina Nusantara University, Indonesia. Institution can utilize the broadcast message feature on the application to send messages in real-time. The proposed architecture model is running well by combining chat server (OpenFire) to accommodate real-time chat for Android platform through XMPP, and Web Server to accommodate web platform through HTTP. Web Service is needed for data exchange between Android platform and Web Server.

80% of respondents are wanting information delivered via IM, and SMS (50%). 93% of respondents are willing to install IM if campus information disseminated via IM. IM can also collect user contact information with registration system on the smartphone using the phone number and or e-mail address. Phone numbers and e-mail
addresses can be used by institution to contact a user directly if there is an urgent need. The web application also supports accessibility for users who do not take a smartphone device.

Some features for further development are adding chat feature with multimedia-based content such as voice (VoIP) and video\textsuperscript{31}, and also developing the application on other operating systems. The application also should be able to classify user categories as per department/faculty or class in order to institution can send broadcast messages specifically, and complete the application with system updates of user's contact information such as phone numbers and e-mail periodically. The application development should consider secure connection factors to avoid data piracy\textsuperscript{32}.

8. Acknowledgements

This research was funded by grant from Bina Nusantara University, Indonesia for the year 2014.

9. References