Moodle Notifications through Facebook Messages

Pemasiri H.T.A.S.
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka
akila.10@cse.mrt.ac.lk

Abstract - Moodle (Modular Object-Oriented Dynamic Learning Environment) is a learning management system that is used in many countries throughout the world. Facebook is an online social networking service that is used by many people throughout the world. Typically when considering a person with a Facebook account as well as a Moodle account, the time that is spent on the Facebook tends to be higher than the time spent on the Moodle. The login frequencies also have the same characteristic. This research paper suggests a method to integrate Moodle with Facebook to increase the effectiveness of the Moodle usage.

I. Introduction

Moodle is a learning management system that is used by 129 countries throughout the world with 73,751,148 users [1]. Facebook is a social networking service that is used by 1.4 Billion people in the world. The average amount of time a person uses Facebook per month is 15 hours and 33 minutes [2]. This is a very high compared to the time a person spends on the Moodle.

Typically people tend to access Facebook through the mobile phones even without a specific reason. It has been recorded that more than 250 million of people [2] access Facebook through their phones. Thus more than 2.5 million websites have been integrated with Facebook [2].

Though Moodle is an effective learning management system, there are inherent pitfalls of the system which have not been addressed yet. One of them is that a user has to navigate through every course page to see what has been added or posted anew. This reduces the user friendliness of the system and increases the probability of user not knowing of an update.

The method suggested in this research paper is to integrate Moodle to Facebook so that the user gets the updates of the Moodle as Facebook messages.

This method ensures that the users receives notifications of all the courses that they The four main sub systems of the whole system are as follow.

have registered on the Moodle and the users are able to stay updated more frequently.

II. Literature Review

According to the statistics it shows that the 50% of the active users log into Facebook at any given day. But when considering the Moodle this number will not be that much higher. Therefore now the attention has been paid on integrating Moodle to Facebook [4].

One approach suggested for the implementation is launching the courses through Facebook [5]. But that will make it compulsory to have a Facebook account and it violates the functionality provided by Moodle. Moreover handling quizzes like things would be harder in such an environment and the possibility of students tend for plagiarism is higher as well.

To eliminate the possibility of missing the updates, a suggested approach is to display all the new notifications in the Moodle home page. But this requires the user to login to Moodle to view the notifications.

Due to the drawbacks of the suggested approaches this paper suggests a new method to integrate Moodle with Facebook, while keeping the original functionality of Moodle as well as keeping the users updated frequently. The method is sending Moodle notifications to Facebook as messages.

III. Design

At a higher level of abstraction, separate systems were designed for each type of notifications. The reason behind the approach is that the administrators of the Moodle sites can add those systems (plugins) to their systems according to their requirements.

The four main sub systems of the whole system are as follow.

- Quiz notification plugin
- Deadline notification plugin

 Resource tracking plugin

Quiz notification plugin and the Deadline notification plugin have the same internal structure whereas the Resource tracking plugin has a different structure.

Figure 1 illustrates the data flow and the process that is used by the system at very high level.

Each subsystem requires the users to be enrolled to the each service separately as the services can be installed separately and they are executed separately.

When considering the internal structure of each subsystem, both Quiz notification plugin and the Deadline notification plugin the process followed and the data flow are very similar.

Figure 2 illustrates the basic process that is followed by Quiz notification plugin and Deadline notification plugin. Through checking the database relevant to the columns that are specified at the implementation level, system finds the tuples that satisfies the given condition. Then the system creates the message using the details available in the other columns. After that the details of the registered users are fetched using the newly created table from the plugin. Finally the system sends the message to the email server.

Both Quiz notification plugin and Deadline notification plugin use Moodle Cron [6] which runs as a background process in their execution.

Resource tracking plugin is designed using the triggers of the database. At each triggers the corresponding details are fetched and a message is created. This again will send the message to the registered users of the system.

IV. IMPLEMENTATION

Moodle core API (Application Programming Interface) is written in PHP. Therefore for the development of the system PHP is used.

Furthermore Moodle is an open source project which is contributed by many developers. Thus Moodle has specified standards for plugin development. For this project the developed plugins are Moodle Blocks [7].

Except the logic used for each of above described plugins, the implementations follow the same format.

Each of the plugins have a class with the plugin name and that class is extended by block_base super class of Moodle. Within the cron() method of this class the main algorithm is implemented and within the get_content()[8] function the user interface related things are implemented. Except these functions there can be many other functions such as createMessage(), sendMessage() to get the relevant tasks done.

Each plugin package has a directory named "db‟ where the database related things are stored. This is used especially when a new table is being created by the plugin. Within the folder "db‟, there is a PHP file named "access.php‟ which holds the capabilities of the newly created block. The "install.xml‟ file in the same directory has the table structure that should be adhered by the newly created table. This file can be easily generated using the XMLDB editor of Moodle [9]. "upgarade.php‟ file in the "db‟ directory contains the code that should be run to upgrade the Moodle database [10].

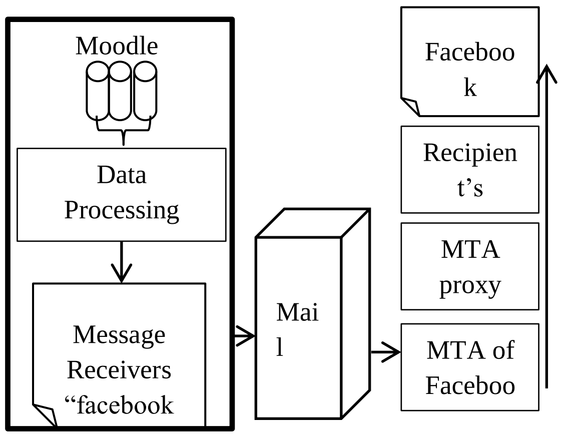


Fig. 1 High-level Design

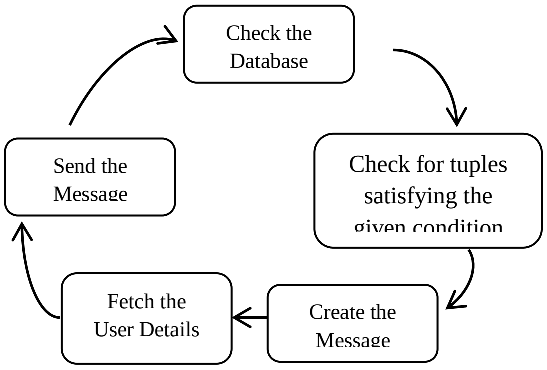


Fig. 2 Process in Quiz notification plugin and deadline notification

```
Quiz_notification
        db
                access.php
                Install.xm
                Upgrade.php

        lang
                en
                        block_quiz_notification.php
                Test
                block_quiz_notification_test.php
README.txt
Block_quiz_notification.php
Version.php
```

Fig. 3. Structure of a sample block

```
[mail function]
SMTP = mail.uom.lk
smtp_port = 25
sendmail_from = moodle@moodle.com
```

Fig. 4. The configuration change in php.ini

Each plugin contains a directory named "lang". Within that directory there is a directory corresponding to the language used in the block. For the implementation the inside directory is "en", as English is the language used for the Moodle that is used for this project. But this can be replaced with another appropriate language code. Language Oriented specific terms are included in the "en" directory that appear in the system.

The "version.php" file at each plugin is used to upgrade the plugin with its evolvements.

For the implementation to access and query the Moodle data base, Data Manipulation API [11] is used. This gives a high level of abstraction and guarantee that the database manipulation will work against different RDBMSes.

Other than achieving the functional requirements, to achieve the non-functional requirements writing to files was done in order to keep the logs.

The server should be configured to send the mails through a mail server. In this implementation php.ini files is edited by adding the mail server related data.

Figure 4 shows an example configuration that is done to php.ini file.

To improve the robustness of the system unit tests are carried out using phpUnit test frame work.

V. Results

The system is implemented in such a way that successfully provides the promised functionality. For the system testing the local host and the mail.uom.lk mail server is used. As the Facebook mail server does not accept the incoming mails from the UOM mail server, for the demonstration purposes mails are sent to the users email.

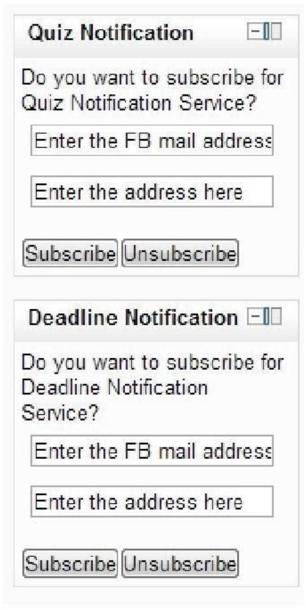
The user interface is shown by figure 5. This will register a User successfully if he/ she has not subscribed to the service.

Figure 6 is a screen shot of an email that has been sent by the system, informing the users about a quiz to be held in the course, which has the course-id 4.

VII. Conclusion and Further Work

The suggested system can be used to use Moodle in more effective and user friendly manner.

Other than what is been implemented up to date the system can be extended by adding features like generating the messages to all the updates of the Moodle, letting the user to select the required Facebook updates and letting the user to decide the frequency to receive reminders on the upcoming deadlines and upcoming quizzes.



Quiz Notification

Do you want to subscribe for Quiz Notification Service?

Enter the FB mail address

Enter the address here

Subscribe Unsubscribe

Deadline Notification

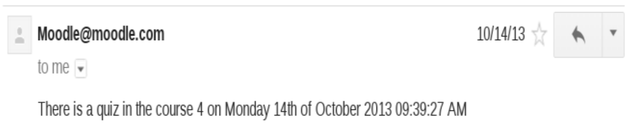
Do you want to subscribe for Deadline Notification Service?

Enter the FB mail address

Enter the address here

Subscribe Unsubscribe

Fig. 5. User subscription user interface


Moodle@moodle.com 10/14/13

to me

There is a quiz in the course 4 on Monday 14th of October 2013 09:39:27 AM

Fig. 6. An email sent by the system

ACKNOWLEDGEMENT

This work was supported by Moodle Community and the Department of Computer Science and Engineering, University of Moratuwa, to whom I am grateful.

BIBLIOGRAPHY

[1] Moodle Community, “Moodle Statistics”. Internet: https://moodle.org/stats/ [JUN 22,2013]

[2] Statistic Brain, “Social Networking Statistics”. Internet: http://www.statisticbrain.com/social- networking-statistics/ [JUN 22,2013]

[3] Jiakai Liu,“Inside Facebook Messages' Application Server”.Internet: http://www.facebook.com/notes /facebook-engineering/inside-facebook-messages-application-server/10150162742108920, Apr.29,2011 [JUL 25,2013]

[4] M Cirovic, , N. Milenkovic,N. Petrovic, V.Jeremic, Z. Radojicic. “FACEBOOK VS. MOODLE: WHAT DO STUDENTS REALLY THINK?” [Online]pp413-421. Available:http://www.icicte.org /Proceedings2013/Papers%202013/ 12-1-Petrovic.pdf [JUN 22,2013]

[5] Moodle Community, “Moodle Statistics”. Internet: https://moodle.org/stats/ [JUN 22,2013]

[6] Artistic Brain, “Social Networking Statistics”. Internet: http://www.statisticbrain.com/social- networking-statistics/ [JUN 22,2013]

[7] Jiakai Liu,“Inside Facebook Messages' Application Server”.Internet: http://www.facebook.com/notes /facebook-engineering/inside-facebook-messages-application-server/10150162742108920, Apr.29,2011 [JUL 25,2013]

[8] M.Cirovic, , N. Milenkovic,N. Petrovic, V.Jeremic, Z. Radojicic. “FACEBOOK VS. MOODLE: WHAT DO STUDENTS REALLY THINK?” [Online]pp413-421. Available:http://www.icicte.org /Proceedings2013/Papers%202013/ 12-1-Petrovic.pdf [JUN 22,2013]

[9] Zaid Alsagoff. “Facebook For Learning? Boleh!”. Internet : http://www.slideshare.net/zaid/facebook- for-learning-boleh, Apr.20, 2010 [JUL 5, 2013]

[10] Jon. Papaioannou. "A Step-by-step Guide To Creating Blocks” Internet: http://docs.Moodle.org/dev/Blocks[JUL 14,2013]

[11] Moodle Community, "Blocks/Appendix A." Internet: http://docs.moodle.org/dev/Blocks/Appendix_A. [JUL 14, 2013]

[12] Moodle Community, "XMLDB editor." Internet: http://docs.moodle.org/dev/XMLDB_editor. [AUG 16,2014]

[13] Moodle Community, "Data manipulation API." Internet: http://docs.moodle.org/dev/Data_manipulation_API. [AUG 30,20]