

Verification and Validation of Comments in Personal Learning Network(s)

Puneet Joshi, Sheetal Sharma, Meer Shizan Ali, Divya Gautam

Abstract— Internet plays a vital role in information sharing. One can type in any word and fetch the data instantly. PLN targets the intellectual geeks and facilitates them by providing all the associated information together on one platform. PLN is a platform where one can upload, share and download videos. One can also upload text documents, images, etc. The user maintains his own profile by logging-in. They can start their own discussions by asking questions and can comment on the resources can be shared and downloaded, also their comments and reviews will be verified and validated. All resources are available at one common place thus saving users time. It is always a matter of concern for developers to decide on what technology will prove the most efficient for implementing the project. To justify the concept, this project is made in two different technologies namely J2EE with third parties as Struts, Hibernate and Java Script and the other as the basic J2EE technology.

Index Terms— PLN, Java, Struts, Hibernate, JSP, Data structures, MySql, comments, bad comments.

1 INTRODUCTION

Internet plays a vital role in information sharing. One can type in any word and fetch the data instantly. PLN targets the intellectual geeks and facilitates them by providing all the associated information together on one platform. PLN is a platform where one can upload, share and download videos. One can also upload text documents, images, etc. The user maintains his own profile by logging-in. They can start their own discussions by asking questions and can comment on the resources can be shared and downloaded, also their comments and reviews will be verified and validated. All resources are available at one common place thus saving users time. It is always a matter of concern for developers to decide on what technology will prove the most efficient for implementing the project. To justify the concept, this project is made in two different technologies namely J2EE with third parties as Struts, Hibernate and Java Script and the other as the basic J2EE technology.

PLN aims to promote a greater count of elites to splurge into the field of Technology. It integrates the benefits of learning with the convenience of a 'no-physical-bar' virtual learning environment, minus the commuting hazards and expenses. It will usher in the immense flexibility and sophistication in the existing learning platform structures, with the perfect blend of synchronous and asynchronous interaction. It provides a means of collaborative learning for its users.

PLN is an online service or platform that focuses on building and reflecting of educational networks among people who

share same interests and activities.

2 FEATURES OF PLN'S

PLN aggregates videos in different categories and allow users to collaborate on them by sharing, commenting and recommending to other users. Video Sharing is a type of service that allows anyone to upload videos and share them with the world.

There are discussion forums where a user can start a discussion by posting queries. Users are free to answer and discuss. They can also suggest edits to existing answers provided by other users. One of the time consuming things for many to take care of on a PLN is moderating the bad language and comments from people attacking your POST. The moderation blacklist allows you to select words that will always be banned from your forum. Add as many as you like and they will never appear again. You don't even have to choose rude words because they will automatically be removed depending on how you set your profanity blacklist. A quiz is posted daily as "question of the day" which enables the participants to explore new technologies.

There is a section providing details on various competitive exams which a student can take after engineering.

The placement cell provides updates on recruitment processes of various companies going on in the premises. Also available are the questions asked in the interviews.

Users can maintain their personal accounts by registering and logging-in.

The implementation of the same concept in two different technologies will flourish the solution for the question of the better technology to work in this situation taking into account all aspects.

- Puneet Joshi is currently pursuing bachelor degree program in Information technology engineering in Malwa Institute of Technology | RGTU, India, PH-09630974205. E-mail: puneetjoshi@hotmail.com
- Sheetal Sharma is currently pursuing bachelor degree program in Information technology engineering in Malwa Institute of Technology | RGTU, India, PH-07566268306. E-mail: sheetal96@live.com
- Meer Shizan Ali, Assistant Professor MIT, Indore E-mail: mshizan@gmail.com.
- Divya Gautam is Head of Department I.T. in Malwa Institute of Technology, Indore. E-mail: divyagautam06@gmail.com

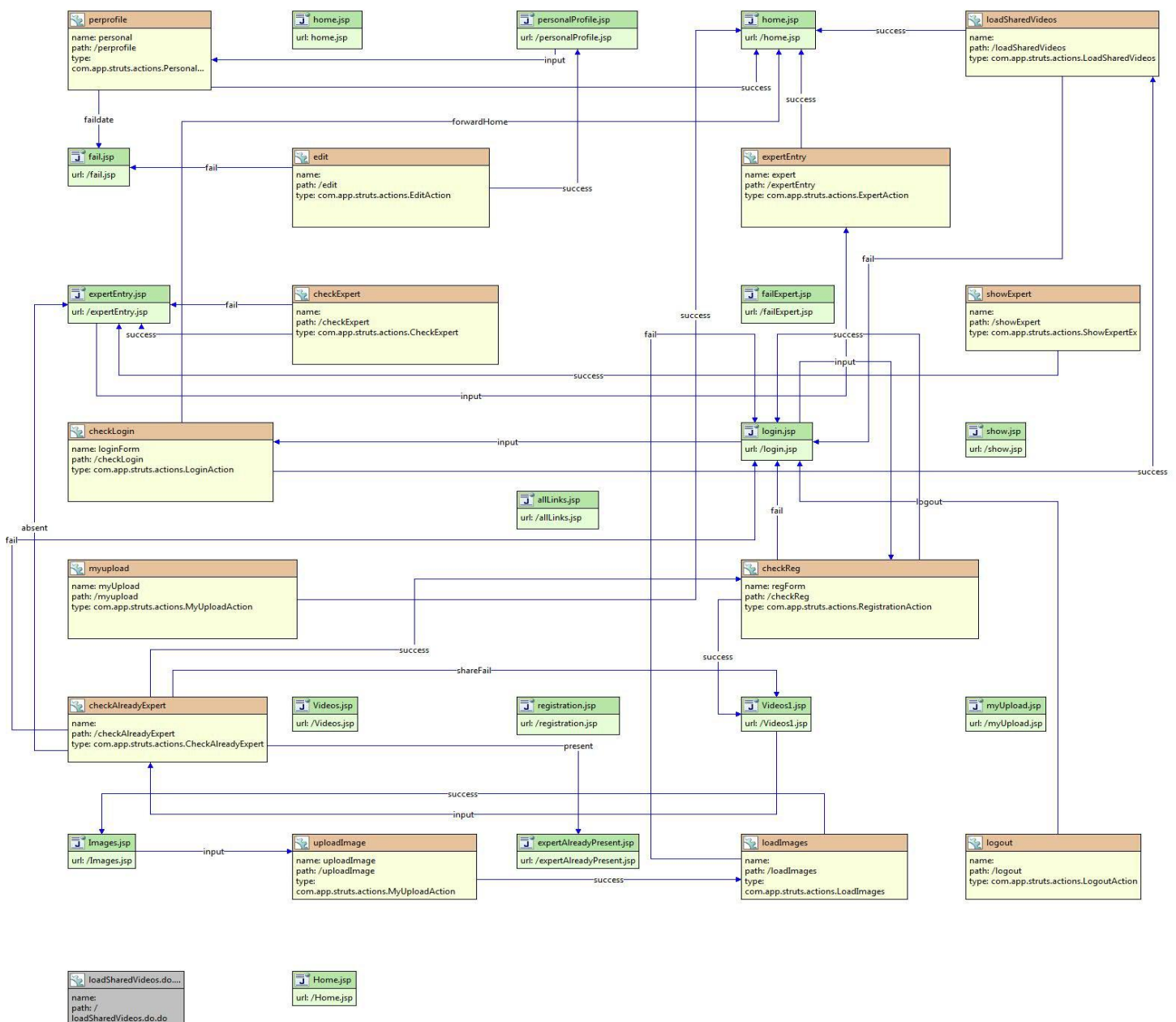
4 RELATED WORK

We have extensively studied the characteristics of comments in discussion forums and the technology used. In order to counter technology, bad commentors do change their mode of operation. These evaluations help us to enhance the existing technology and thereby help us to combat bad comments effectively. We observed that bad commentors use different combinations to post the comments.

5 ARCHITECTURE OF APPLICATION

Implementation of the code is done using the MVC architecture by struts and hibernates to optimize the entire work.

Struts-config.xml



The xml mapping of struts-config. Is shown, the bad comment validator is in the discussion forum part of the application, where the expert advice and doubt discussions are being done in the form of queries and replies.

6 OPERATIONAL METHODOLOGY OF COMMENT DETECTION

To prevent bad words from sites which allows commenting, blog filter can be used. This supports a reliable protection to the blog, forum & guestbooks owners. It works with detecting bad words in a comment and finding the probability whether the comment is bad or not. Incoming comment and earlier comments stored in database are used to derive this formula. To bypass comment filters, notorious users use punctuations, capital letters in between

small letters and special characters to modify the meaning to the filters. To avoid this, the words that are sent from users are counted and unnecessary words like punctuation and special characters eliminated. After this elimination process, the entire user comment should be stored in database in lower case format.

7. ALGORITHM

The following algorithm shows the structure of our unwanted comment blocking methodology.

Step 1: The system asks user to enter their name and email addresses. The entered names and email addresses are verified for its validity before connecting to the database and web page.

Step 2: The users IP address should be noted and stored in the database.

Step 3: Unwanted characters like punctuations and special characters should be removed.

Step 4: All words should be converted to lowercase letters.

Step 5: Unnecessary words should be removed & total number of words counted.

Step 6: If the user input is bad comment, the IP address of the sender will be stored in database.

Step 7: Go to step 01

Step 8: Again if there is any comment bad comment from the same IP address for a particular number of times, the IP address will be blocked for a particular period of time.

Step 9: To avoid flooding of data, the user should be barred to send comment without interval.

8. WORKING METHODOLOGY OF ALGORITHM

We selected a code to derive a formula to detect whether the given comment is bad comment or not. We derived this formula carefully to avoid false positive and false negative comments. A legitimate comment may contain a couple of bad words.

Also, a comment may contain some good words and this does not mean that the comment is legitimate. Therefore, a probability mechanism is designed to distinguish incoming comments. We derived this code to block incoming comments from the be used along with the existing data in the database end-users. To implement this code, the administrator should store sample bad comment words in the database. The incoming comment will.

The first 5 steps mentioned in this algorithm are basic programming techniques, so we are not going to explain them here. We are going to deal directly with the mechanism to identify the comment. Our first step is to establish a connection

between the database and our comment page.

In **Step 1**, the system asks users to enter their name & email address. The entered names and email addresses are verified for its validity before connecting to the database and web page. Without proper name & mail address, users are not allowed to enter a comment. To support dynamic user participation, the users are encouraged to post their comment without registering into the local host. After submitting, the page is directly connected to the database.

In **Step 2**, the users IP address should be noted and stored in the database. In the next step, unwanted characters like punctuations and special characters should be removed.

Users mix lower case, uppercase characters and punctuations to bypass the filters. To prevent this, all incoming user's comments will be free from punctuations and all words will be converted to lower case.

In **Step 3**, comments should be eliminated from unnecessary words. Those words are not counted as bad. However, they are counted from the bad word filter. To increase the performance of the bad comment filter, unnecessary words should be ignored. These words must be defined by the programmer. Selection of those words depends on the programmer.

In **Step 4** of the algorithm, unnecessary words should be removed & total number of words counted. The unnecessary words include pronouns-like 'am', 'is', 'are', 'he', 'she', 'it', 'you', 'we', 'they', 'i', 'have', 'has', 'had', 'and', 'us', 'do', 'does', 'did', 'was', 'were', 'a', 'an', 'in', 'on', 'the', 'to', 'but', 'of', 'from', 'them', 'also', 'their' will be removed. The administrator can define any such words by himself. This part of the program shows the indication of unnecessary words and elimination of those words from the comment text.

Basic methodology

// Elimination of unnecessary words from the comment

```
Extras=array
('am','is','are','he','she','it','you','we','they','i','have','has',
'had','and','us','do','does','did','was','were','a','an','in','on','the',
'to','but','of',
'from','them','also','their');
dizi=explode(" ",comment_db);
foreach(dizi as words){
counter=0;
foreach(extras as gextras){
if(gextra==words)
counter++;
}
if(counter==0)
gwords[words]++;
}
foreach(gwords as key=>value){
print(key." = ".value." ");
sor="insert into table2( form_id, words, adet) values ('1',
'key','value')";
register= mysql_query( sor );
```

// Defining bad words and their repetition time

```
bad_words=array( 'idiot' , 'stupid' , 'bad' , 'awful' , 'terrible' ,  
'disgusting' , 'silly' , 'freak' , 'fool' , 'rubbish' , 'ugly' );  
// more can be included in the array.  
count=str_word_count(comment_db);  
dizi=explode(" ",comment_db);  
    foreach(dizi: words){  
        foreach(bad_words : bad){  
            if(bad==words)  
                bad_count++;}  
    }  
print(" bad words in our comments:");  
print(bad_count);
```

9 CONCLUSION

In the scenario of PLN(s) and discussion forums, we have seen bad comments on some useful and important information. We implemented a module which is capable enough to verify and validate the bad abusive words from the comments of experts in our expert panel doubt discussion forum. This verification of comments will be real time. Which will stop the bad words in the important discussions and will enhance learning.

REFERENCES

No References.