Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 04, April -2018

PROPOSED WORK OF AUTOMATED QUESTION ANSWERING SYSTEM DEPEND UPON CROWDSOURCING

[1]Gulafshan Parveen, [2]Prof. Naziya Pathan, [3]Prof. Shyam Dubey

[1]*M.tech. Research scholar. NCEOT, Nagpur.*
[2]*Asst. professor, NCEOT, Nagpur*
[3]*Asst. Professor & HOD, NCEOT, Nagpur*

Abstract:-*Proposed system is used to solve the problem and deficiencies which come across in search engine, which produces a suitable answer for the received question and present to the questioner using crowd sourcing. The system will allows user or publisher to submit the question in the category of their choice. Crowd sourcing increases scalability due to assigning tasks to crowd. The purpose of this work is to combine strong human capability to understand computer science algorithm. Also, it provide the way by using which any user can ask his question without bothering about its answer. The main objective of this work is to make an application that will allow user to post the question in a specific category where he/she get a quick answer from crowd workers. Once the question is published in a particular category, the system sends the notification to its subscriber about the new question. This will be done after the admin approval. It uses internet facility for answer retrieval.*

KEYWORDS**:** *crowd sourcing, MongoDB, Servlet, JDK, Eclipse, Bootstrap Template, Tomcat Server, CQA(Community Question Answering)site, Subscriber(Crowd Worker), Publisher(Requester), RDBMS, API, JDBC, Mysql.*

1. Introduction

As the Internet is a junction of different types of users, each user usually uses different websites to search appropriate answers for their questions. But suppose, if a particular user is searching his answer and he will fail for getting the exact answer then he assumes that he will not find the answer of his question by searching directly in the search engine. This is the problem that real human being can understand better than the computer. In such a case, a user wouldn't refer to ask query to search engine rather than he would ask his question to some other community question answering site. The most common CQA is Yahoo. In this way user can get his answer with the help of other user. Also if the user wants to frequently ask the question in the search engine, he should use some special keys which is not use by other user.

Internet is a junction of user increasing per unit time, this leads to increasing number of questioners at a single time but the site contain limited number of accounts. Due to this maximum no. of questions are unanswered. The report shows that the result of search on Yahoo gives 15% of exact answer and 25% of answer are repetitious. Further, the percentage of other categories answer is very less. For example, if the question asked is about Persian or greek language then it will not answered since internet has very low content about Persian or greek language. Therefore, users unable to benefit from the advantages of feature which CQAs offer.

2. Limitations of Existing System

Question answering is one of the major components of such personal assistants. Existing techniques already allows users to get direct answers to their factoid questions. To cater to these informational needs, community question answering (CQA) sites emerged, such as Yahoo! Answers and Stack Exchange. These sites provide a popular way to connect information seekers with answerers.

Advantages:

- These sites provide a popular way to connect information seekers with answerers.
- It can take minutes or hours, and sometimes days, for the community to respond.
- Some questions are left unanswered altogether.
- Existing system is not limited to particular area of interest.

Disadvantages:

- However, there is still a large number of more complex questions, such as advice or accepted general opinions, for which users have to dig into the "10 blue links" and extract or synthesize answers from information buried within the retrieved documents.
- The existing system only allows users to ask question without providing.
- An additional data like documents in form of attachment.
- The existing system doesn't have any chat room functionality.

3. Problem formulation

Modern search engines have made dramatic progress in the answering of many user's questions about facts, such as those that might be retrieved or directly inferred from a knowledge base. However, many other questions that real users ask are more complex, such as asking for opinions or advice for a particular situation, and are still largely beyond the competence of the computer systems.
As conversational agents become more popular, QA systems are increasingly expected to handle such complex questions, and to do so in (nearly) real-time, as the searcher is unlikely to wait longer than a minute or two for an answer.
One way to overcome some of the challenges in complex question answering is crowd sourcing. We explore two ways crowd sourcing can assist a question answering system that operates in (near) real time: by providing answer validation, which could be used to filter or re-rank the candidate answers, and by creating the answer candidates directly. Specifically, we focus on understanding the effects of time restrictions in the near real-time QA setting.
The main objective is to make an application that will allow user to post the question in a specific category where he/she get a quick answer from crowd workers. Once the question is published in a particular category, the system sends the notification to its subscriber about the new question. This will done after the admin approval

4. Proposed Work

The main purpose of this research was proposing a QA system based on crowd sourcing platform using java technology. Our CRQA (Crowd-powered Real-time Question Answering) system represents a hybrid system, which will includes an automatic question answering and crowd sourcing modules.

In this work we will implement following modules

Modules:

Web Application:

- Admin
- Publisher (requester)
- Subscriber(crowd worker)
- Categories Service
- Questions Service
- Answers Service.

The database will be as follow

Database Design:

- Question table
- Answer table
- Admin table
- User table.

5. Design Description

This system will be developed using J2EE of java. In this, system will use bootstrap & jquery for developing the client ui. The client request and response will be handling by the service layer which will be implemented using Java servlet API. Servlet technology is robust and scalable because of java language. On the database side this system is going to use MySQL

database server which is an open-source relational database management system (RDBMS). For database connectivity this system will use JDBC which is a java API to connect and execute query with the database. JDBC API uses jdbc drivers to connect with the database. In another word Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database.

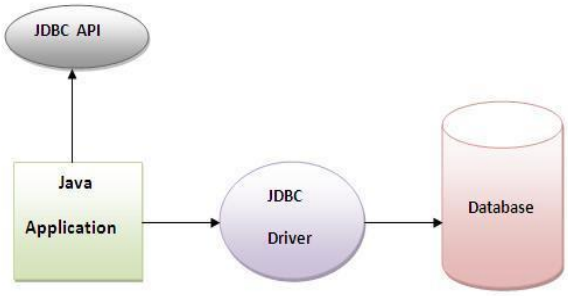

Fig. Interface Between API and DB

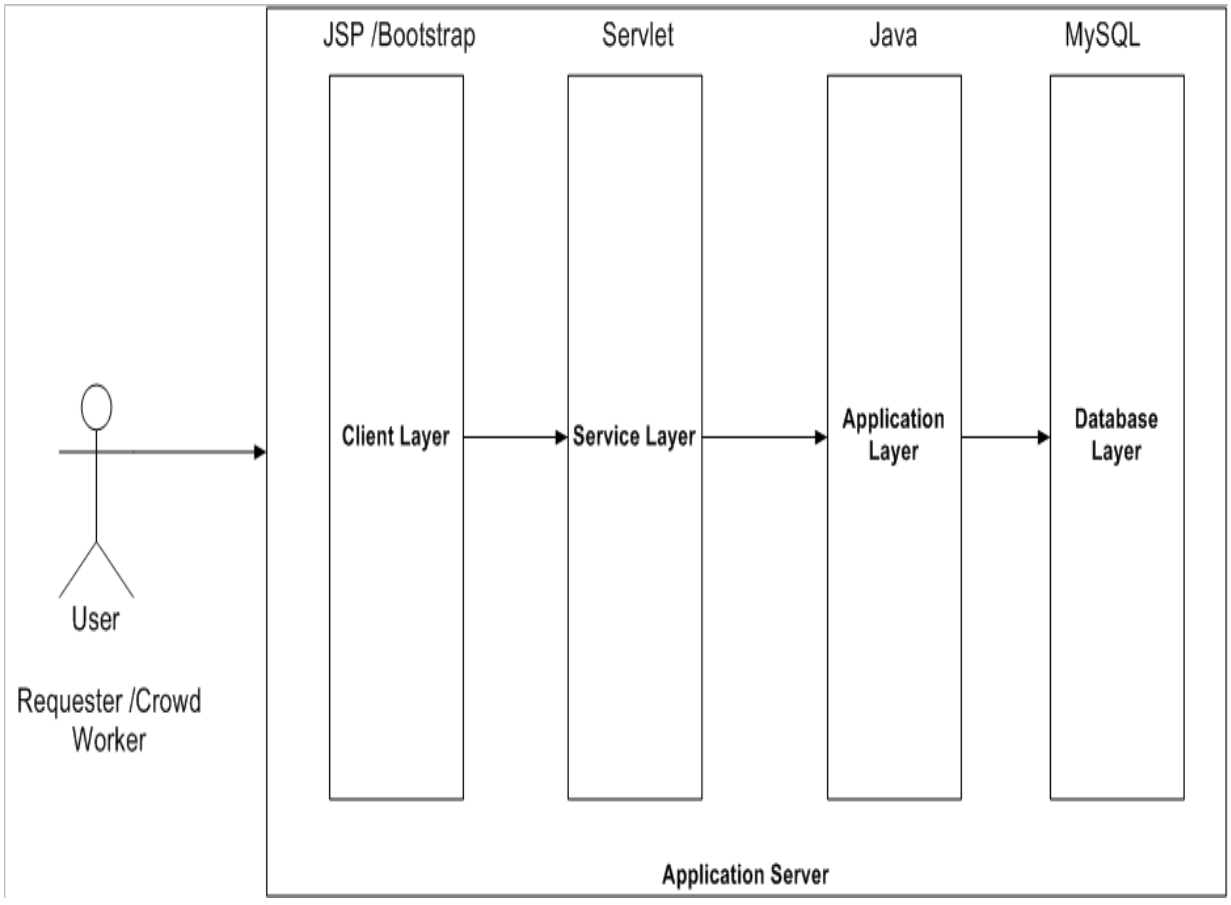

Fig. Interfacing of Each Layer

The jdbc is a low level data accessing api which allows application to connect with database. JDBC is part of the Java Standard Edition platform, from Oracle Corporation. In this application the term "requester" will be used for a user who is going to post a question using publisher modules functionality and "crowd worker" is one who is going to answer that questions using subscriber module.

The figure shown below describes the general procedure of QA system. In this system, requester sends his question and the application receives the question and again sends to the subscriber. After this, if any subscriber knows the answer of the question, requested by requester, gives the answer to the requester.

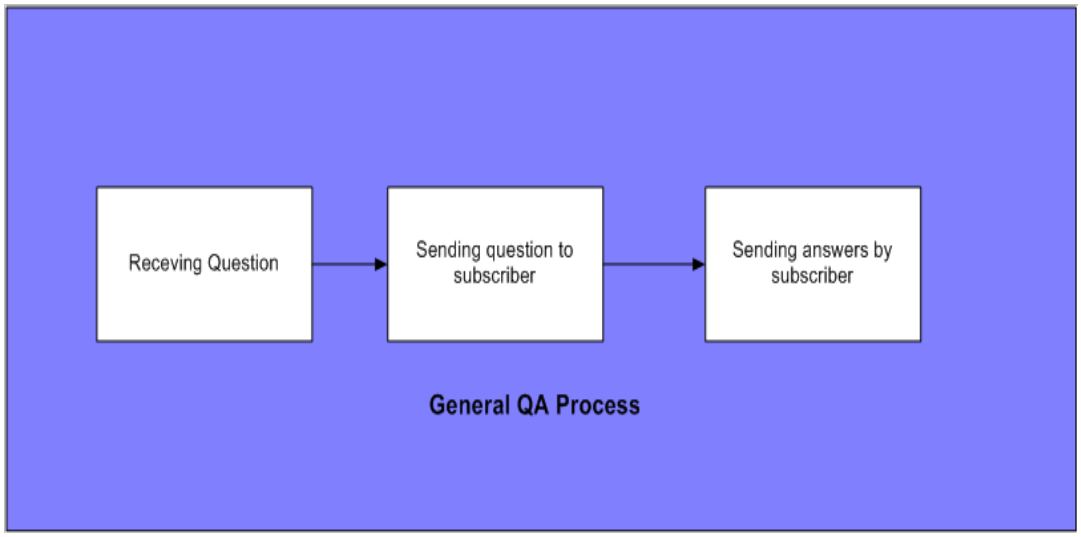

Fig . General QA Procedure

On the other hand, the proposed system receives the question from publisher or requester. After receiving the question, it is send to admin for approval. The given question is accepted by admin and it generates one notification which is send to any subscriber or crowd worker. If anybody knows the answer of the requested question then he or she can give it to the requester. The main advantage of this work is that, once the question is requested, anyone can answer to this question whether it is publisher or other subscriber. The overall working of this proposed system is shown by the diagram shown below.

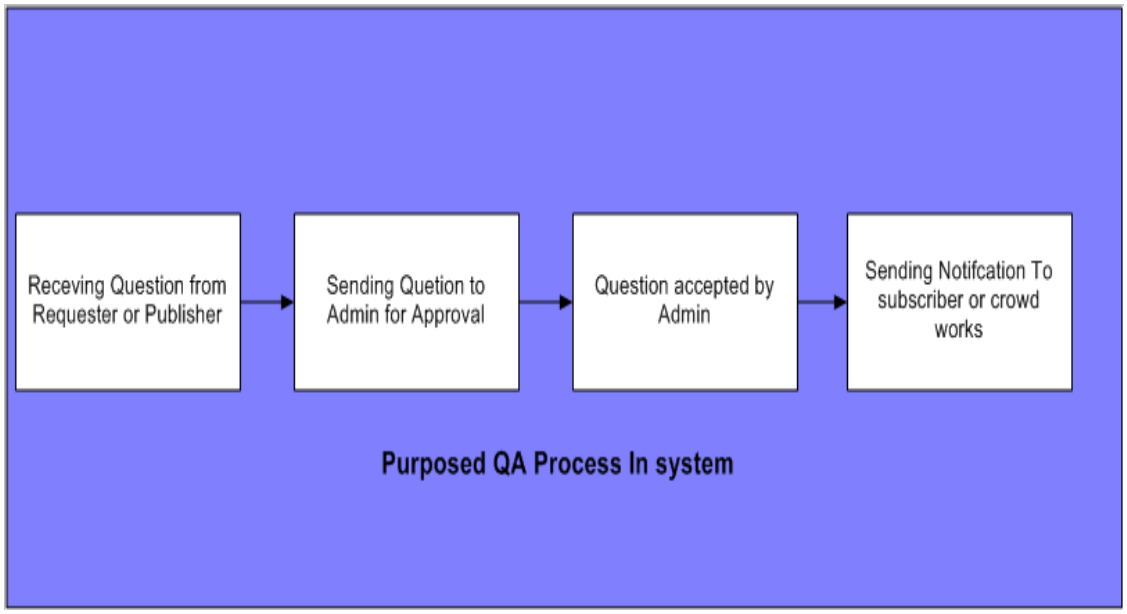

Fig. QA Process of Proposed System

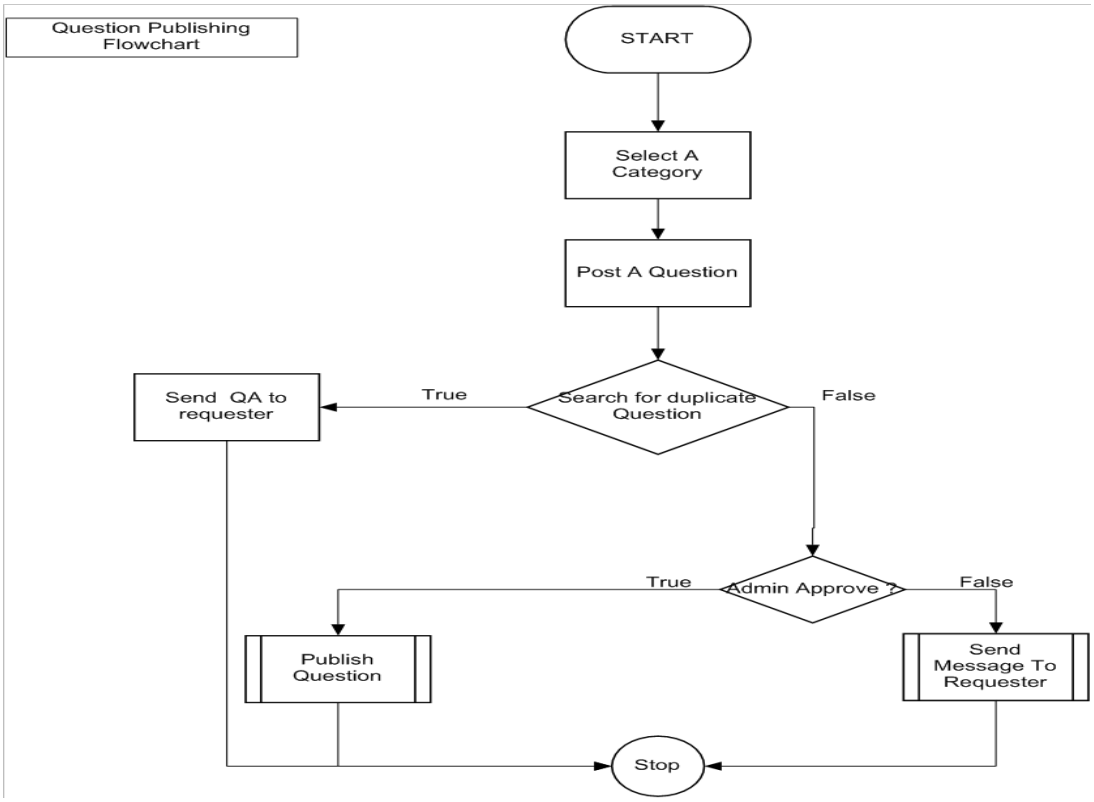

Fig. Flow Chart of Question Publishing

Once the question is published, the next step is posting of answer. The diagram shown below is used for the stated purpose.

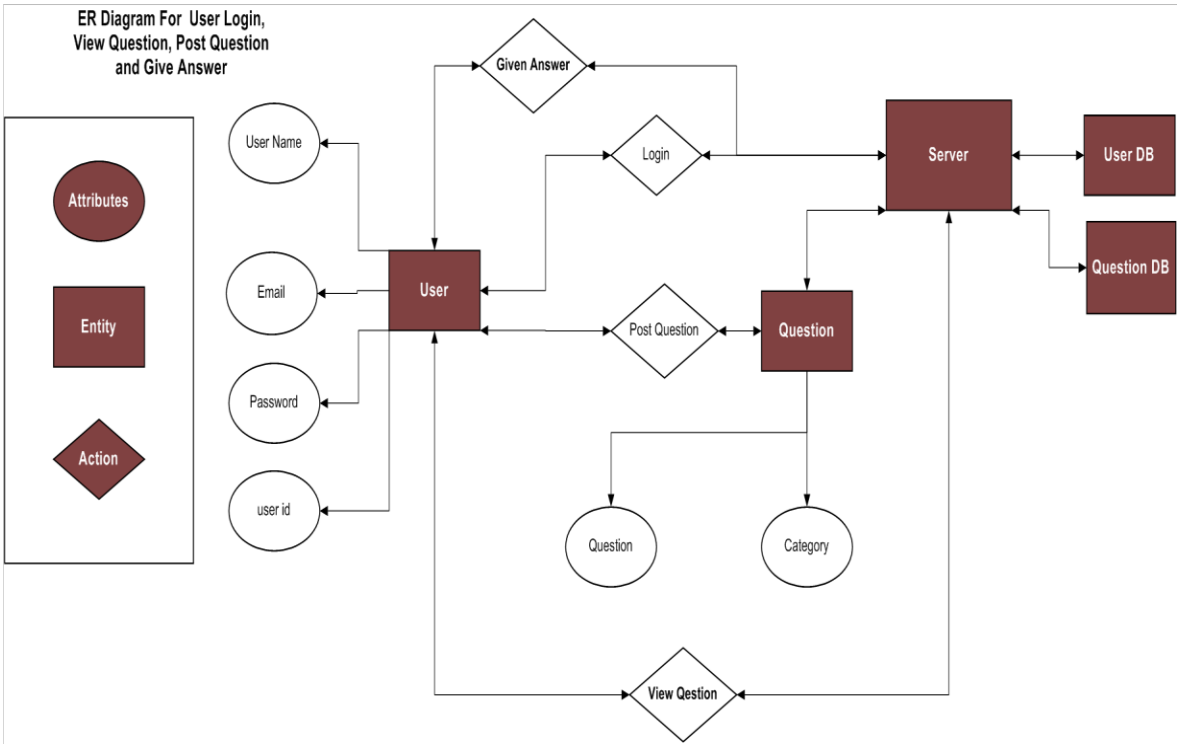

Fig. ER Diagram of View Question, Post Question and Give Answer

6. Conclusion

This system intend to provide a new QA system which will be designed using Java J2EE. This system will be crowd sourcing platform that will allow user to submit a question where crowd worker will answer that question as per there level of knowledge. The system will also be integrated with admin functionality which will act as a site administrator, who will monitor all the posted questions. Once a question is posted by the publisher, the admin will get a notification and after admin approval the question is available for subscriber to answer that question. In this system the publisher can also be a subscriber and a subscriber can also be publisher.

References

1. Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chandal K.Roy, and Kevin A.Schneider "Answering Questions about Unanswered Questions of Stack-overflow" in MSR'13 proceedings of 10th working conference on mining software repositories.
2. Shtok, G. Dror, Y. Maarek, and I. Szpektor, "Learning from the Past: Answering New Questions with Past Answers," in Proceedings of the 21st international conference on World Wide Web - WWW '12, 2012, p. 759
3. K. Kim, S. Lee, J. Son, and M. Cha, "Finding informative Q&As on twitter," in Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion, 2014, pp. 319–320
4. Ke Mao, Ye Yang and Qing Wang "Developer Recommendation for Crowd Sourced software Development Tasks " in 2015 Published in Service- Oriented System Engineering(SOSE), 2015 IEEE Symposium.
5. L. B. Chilton, G. Little, D. Edge, D. S. Weld, and J. A. Landay, "Cascade: Crowd sourcing Taxonomy Creation," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems -CHI '13, 2013, p. 1999.
6. G. Xintong, W. Hongzhi, Y. Song, and G. Hong, "Brief survey of crowd sourcing for data mining," Expert Syst. Appl., vol. 41, no. 17, pp. 7987–7994, Dec. 2014
7. J. Hentschel and J. Pal, "Sada Vehra: a framework for crowd sourcing Punjabi language content," in Proceedings of the Seventh International Conference on Information and Communication Technologies and Development - ICTD '15, 2015, pp. 1–4.
8. S.-K. Choi, J.-H. Shin, and Y.-G. Kim, "Semi-automatic Filtering of Translation Errors in Triangle Corpus."
9. L. Mitchell, S. O'Brien, and J. Roturier, "Quality evaluation in community post-editing," Mach. Transl., vol. 28, no. 3–4, pp. 237– 262, Dec. 2014.
10. O. Alonso, D. E. Rose, and B. Stewart, "Crowd sourcing for relevance evaluation," ACM SIGIR Forum, vol. 42, no. 2, p. 9, Nov. 4192
11. M.-C. Yuen, I. King, and K.-S. Leung, "A Survey of Crowd sourcing Systems," in 2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing, 2011, pp. 766–773

