International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

Implementing Knowledge based Band Management Tool

Yugandhara A. Nimbalkar[1], Justus Selwyn*[2]

[1]School of Computing Sciences and Engineering, VIT University, Chennai, India
nimbalkaryugandhara.ajitsinh2015@vit.ac.in[1]
[*2] School of Computing Sciences and Engineering, VIT University, Chennai, India
justus.s@vit.ac.in[2]

ABSTRACT

Band management tool (BMT) is designed for music bands to organize its regular functionalities, and collect data for analytics using the knowledgebase. The tool is a multipurpose organizer that is used for coordinating the activities of a band, by collecting the musicians in each of the bands, concert details, feedbacks, planning the concert & practice sessions, and finally collecting all these details for further improvement. Musicians can also share the related information and audio clips using this tool for further enhancing their knowledge and keeping them up-to-date. This tool tries to provide a knowledge based feedback for the band owner who will use the knowledge collected to improve their band's operational performances. The class design and implementation details of the BMT is presented in this paper.

Keywords: Web based tool, AngularJS, NodeJS, MongoDB, ExpressJS, and Client-Server Model

I. INTRODUCTION

Today, management and planning several activities are taking importance in common man's life. Everything should be planned and kept organized so that people associated to one's life need not waste their time and money. Many ideas are coming up for Management Systems in different fields. If management of data is vital for an individual, it has become more vital for organized groups like music/dance bands, football/cricket teams, or other similar groups that work towards a common goal. Corporates and multi-national organizations have a structured knowledge management systems (KMS) built for their own needs; such tools are not available for smaller teams or bands. This Band Management Tool (BMT) is a miniature of a bigger KMS, which serves the purpose of coordinating the bands and help them improve in their function over a period of time.

The motivation for developing such a software tool for small functioning teams has risen out of the need for managinglarge realtime data and their process improvement. To enable this, a knowledge based system or a tool is required to perform analytics and forecast future events and scope for improvement.

In this paper, the class design for each components of the BMT is presented along with its implementation details. A few parts of the tool are tested for its performance and throughput. Before designing such a knowledge based tool, relevant literature survey is carried out.

II. LITERATURE STUDY

F. J. Muñoz-Rodríguez and G. Jiménez-Castillo have designed, developed and implemented an app which manages to control an electronic lab bench [2]. The implementation has been done using Java. This can prove a good project in E-learning. Amjad Farooq and Imran Mir have designed and developed a web tool for University [1]. The proposed tool covers the important aspects of students and teacher and has got some new features which are different than the traditional web applications.

Yunhua Gu, Shu Shen, Jin Wang and Jeong-Uk Kim give the application of MongoDB in Information

Collection System [3]. They have proved that the relational database system having multiple tables and each of them having foreign key have low query performance. Also, they have high cost of vertical scalability. There is inconsistency in Object-Relation mapping.

Abdur Rahman.A and Chitra Devi S. have developed a web based content management system which carries the advantages of accessibility, consistency and security [4]. This software is developed using AngularJS and Joomla.

Ravi Sharma and Shipra Ravi Kumar have developed the strategies for Web Application development process [7]. They have discussed these methodologies based on agile development approach. One of the approaches for web application development discussed was UML based. It is easy and effective approach to manage and control development process by the development of UML diagrams.

III. THE PROPOSED TOOL DESIGN

In the proposed work, the web based tool is created for the Music Bands across the country. The users who will be using this tool are administrator, band leader, musicians, Event Organizers and band manager. Each of them have been assigned with the different access rights. Accordingly, their views will change. At first, the admin is created and given all the access rights. He can view the details of all the users. He can also see the events created by users and the transactions taking place between two users.

Event organizers and Band leader have the rights to view their own as well as the users' profile under him as shown in figure 3 and 4 respectively. While the band manager and musicians can view only their profile. If the band manager is the band leader himself then he still can view the other musician profile. The following sub-sections present detailed description of access rights of the actors.

A. Administrator

The administrator is the overall controller of the tool's operations. The access rights provided for the administrator is that he can view all users profile and details of the users' transactions and events. He can approve or disapprove the accounts created by any other users. He can broadcast some important messages to the different users group or all of them. If an account is inactive since long time, then he has the right to delete that account permanently from the database. The class diagram for administrator is given in fig. 1.

B. Event Organisers

The event organizer is the one who organizes events for music bands. He can invite the music bands as per the style of the music bands and the occasion. The creation of event has the fields like name of event, venue, start time, end time and so on.

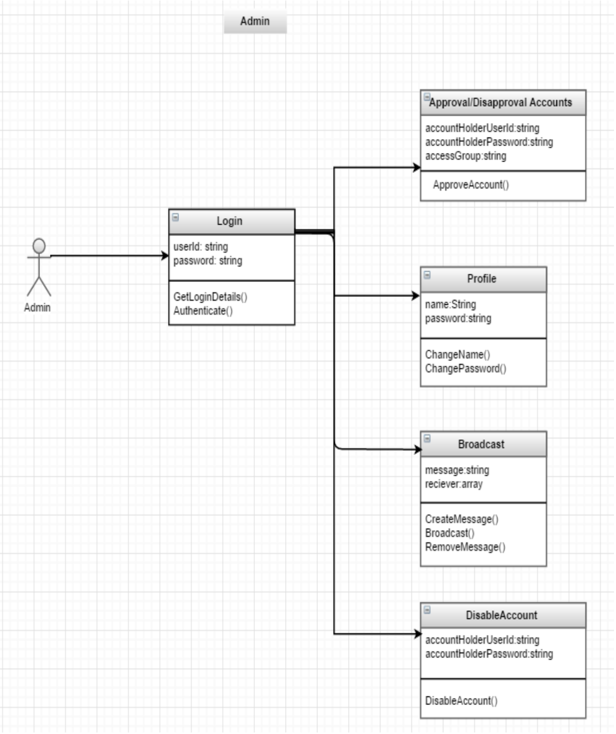


Figure 1. Administrator's class diagram

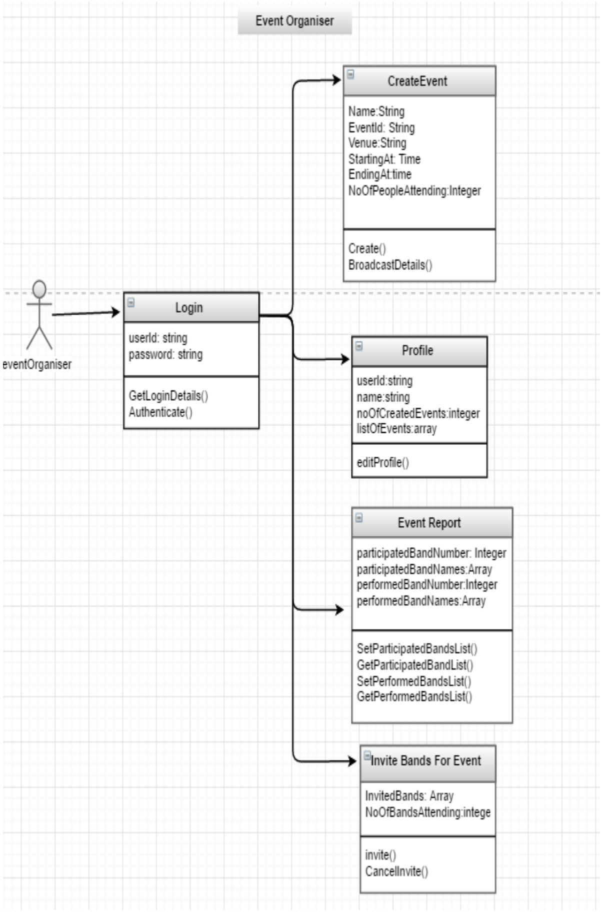


Figure 2. Event Organizer's class diagram

The methods of this functionality include create and broadcast the event. After the event is finished, he needs to create the report of the event and gives reviews about the participated bands. The fields included in this functionality include the number of bands participated, the number of bands performed, the names of bands participated and the names of bands performed. This will help other event organizers to select and invite those Bands. The class diagram for Event-Organizer is given in fig. 2.

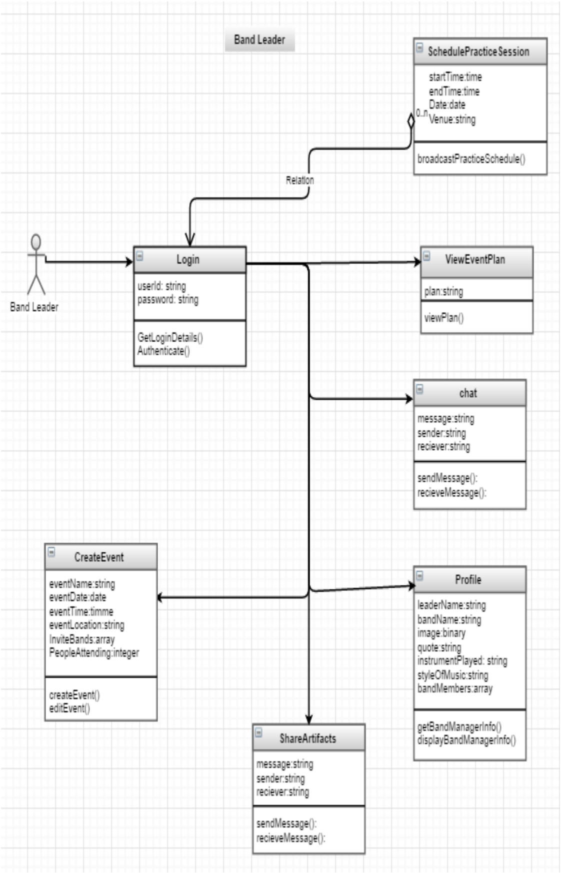


Figure 3. Band Leader's class diagram

C. Band Leader

He can choose one of the musicians as the band manager or he can choose himself or somebody else who is not a musician.

He also has the right to create events and invite other bands for the event. He has the rights to share the necessary artefacts (in the pdf format) with the band members. He can chat with his band members. He also has the rights to schedule the practice session for the upcoming events. He can view the detailed plan of the event created by the band manager for the band. The class diagram for band-leader is given in fig. 3.

D. Band Manager

Band manager is the one who works on the technical aspect of the band and helps the band leader in his work. It is band leader's wish to select a different person as band manager or to work on technical aspects all by himself. Depending on this, the band manager's access rights are separated from the band leader's ones. The class diagram for band-manager is given in fig. 4.

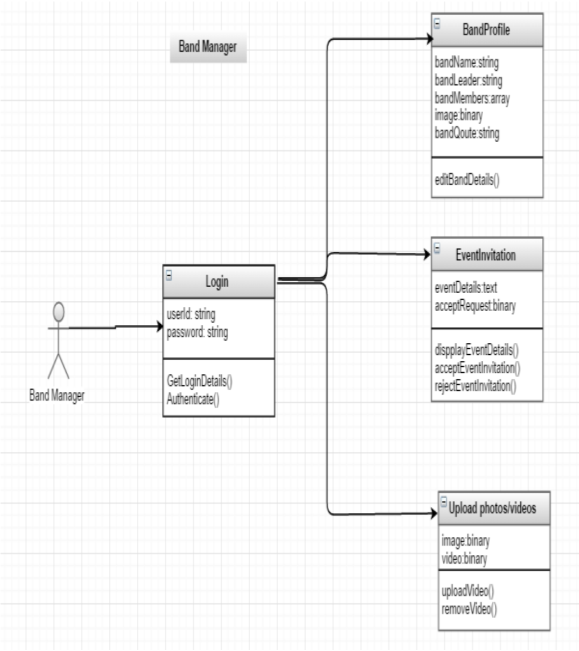


Figure 4. Band manager's class diagram

Band manager has the rights to upload the videos and images on behalf of the band regarding the event which has occurred. The fields included in this functionality are image or video file. The methods included are add video/image and remove video/image.

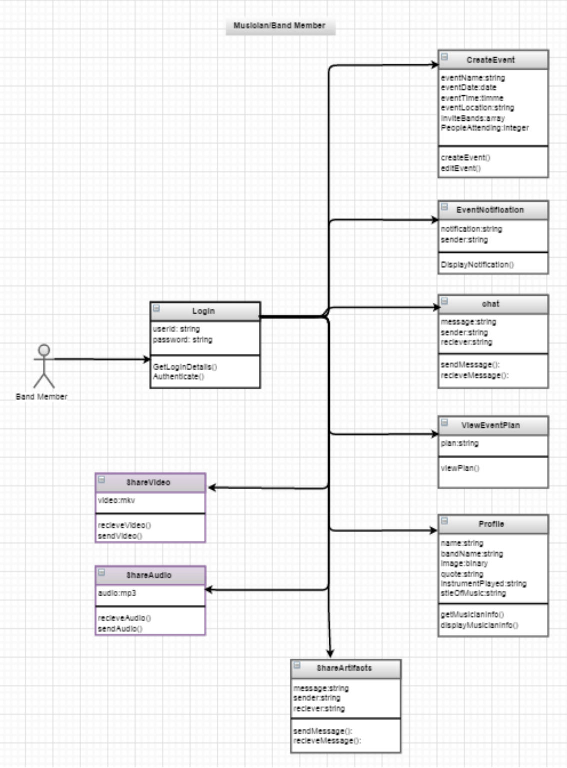


Figure 5. Band member's class diagram

E. Musician/ Band member

Band members and musicians together with their band member perform at the events. They play different instruments and form a team. They can view the detailed plan of the event sent by their band manager. The fields included in this functionality are plan and method is view the plan. They can chat with the other group members and share artefacts in the form of pdf. They can share audio and video necessary for their rehearsals. The fields for this functionality sender, receiver, and the audio or video file. The class diagram for band-member is given in fig. 5

F. Knowledge based Modules

While the above discussed modules form the main part of the functioning of BMT, the underlying framework is based on knowledge management principles. The Knowledge representation models in [8] [9], have been implemented in this tool. The actors – admin, band leader, band manager, member and event organizers – are represented as concept structured model (CSM) object, while the contents that are shared between these actors are represented as object structured model (OSM) objects [8]. These object models are stored in the object-relational framework and later converted to a MangoDB construct.

The NodeJs and the AngularJs constructs does the communication between the front-end application view container and the backend object-relational data constructs. The model, view and controller part of the BMT talks in terms of knowledge unit objects, which made the tool to function efficiently, the results are posted in the subsequent sections.

IV. TOOL IMPLEMENTATION

Some of the implementation specification for BMT is given. MEAN Stack is used. MEAN stands for M-MongoDB, E-Express, A- AngularJS and N-Node.js [5]

1. MongoDB- It is document oriented database system. It stores the data in the form of collections of documents [5]. The documents contain fields which are the elements of data.
2. Express- Express also called as ExpressJS is used at the backend and is the framework of Node.js. It is written in Javascript.
3. AngularJS- AngularJS along with HTML is used at the frontend. This is an open source software and is also written in Javascript. AngularJS can create light-weight, user-friendly and dynamic applications.

4. Node.js- It is an open source software and is used at the backend along with MongoDB. This can also be used to develop APIs.

At first, the server is created on the backend using NodeJs. In that web server is created using ExpressJs. The other necessary modules for authentication, mongoose, and development of log file, images and the other modules developed for routing of the routes are created. The mongoose module is necessary to connect NodeJs with MongoDB. The Schema is created using mongoose and this Schema is converted to models to put them into database. Each document in the database can be managed using CRUD (create, read, update and delete) commands.

The database of MongoDB contains collections of each of the users like band manager, band leader, band member, event organiser and admin. Also it has the collections of instruments played, genre, events created and invitations.

The client-side developed in AngularJs has model-view-controller. In AngularJs, the app is created that controls all the functionalities in the front-end. The $scope variable is used to create the model on the client-side. The view is created using HTML and controller is created using AngularJs. Each view is connected to the controller. The controller is the processing unit of the MVC model.
On the client-side, the template is set by the viewer container which is the basic view of the tool. The others views are injected in this view because this is a single page application.

V. IMPLEMENTATION RESULTS

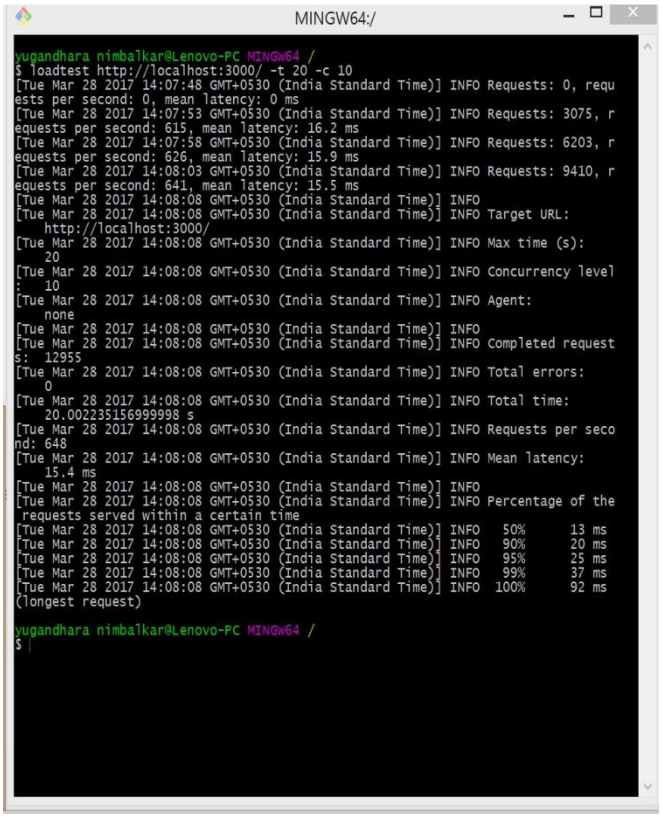
The Band Management Tool (BMT) is analyzed for number of requests sent in per second. The evaluation is also be carried over a period of time with number of different requests.

To analyse this, a module is installed in NodeJS called loadtest. This is used to run the load test on particular HTTP requests or URLs. For this, loadtest is needed to be installed. Following command is used for installing the loadtest:

```
# npm install –g loadtest  --save
```

Open 3 windows of command prompt. Run our application in one window. In second window, run the command #testserver-loadtest loadtest and in third window run the command, #http://localhost:3000/ -t 20 –c 10

This will generate the analysis as shown in fig. 6.



```
yugandhara nimbalkar@Lenovo-PC MINGW64 /
$ loadtest http://localhost:3000/ -t 20 -c 10
[Tue Mar 28 2017 14:07:48 GMT+0530 (India Standard Time)] INFO Requests: 0, requ
ests per second: 0, mean latency: 0 ms
[Tue Mar 28 2017 14:07:53 GMT+0530 (India Standard Time)] INFO Requests: 3075, r
equests per second: 615, mean latency: 16.2 ms
[Tue Mar 28 2017 14:07:58 GMT+0530 (India Standard Time)] INFO Requests: 6203, r
equests per second: 626, mean latency: 15.9 ms
[Tue Mar 28 2017 14:08:03 GMT+0530 (India Standard Time)] INFO Requests: 9410, r
equests per second: 641, mean latency: 15.5 ms
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Target URL:
    http://localhost:3000/
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Max time (s):
    20
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Concurrency level
:   10
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Agent:
    none
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Completed request
s:  12955
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Total errors:
    0
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Total time:
    20.002235156999998 s
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Requests per seco
nd: 648
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Mean latency:
    15.4 ms
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO Percentage of the
 requests served within a certain time
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO   50%      13 ms
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO   90%      20 ms
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO   95%      25 ms
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO   99%      37 ms
[Tue Mar 28 2017 14:08:08 GMT+0530 (India Standard Time)] INFO  100%      92 ms
(longest request)

yugandhara nimbalkar@Lenovo-PC MINGW64 /
$
```

Figure 6. Analysis of load screen shot

The Tool is executed for testing some of the performance parameters. The test results of the tool upon executing the tool with 100 virtual users generating number of requests is given in Table 1. The tool is executed with three different music band details, with each instance executed in the duration of 500 seconds to 1000 secs approximately. The request to the server per second is also given. The mean latency between two requests is on average 15.86ms. This is very significant and encouraging. In addition, the number of hits to the server per second based on the requests per second is also captured. It seems to be 0.6753 hits per request on an average has been encountered with the server.

Table 2 gives the analysis about some additional attributes in performance testing. The submodules in the BMT are taken separately and assessed for the following test metric values. The maximum execution of the test cycle is given in seconds, which is presented in the descending order of its execution time.

Apparently all the other parameters are also proportionately decreasing. This in one way shows the complexity of execution of these modules, where each module differs in access permissions, rules imposed etc. The completed requests in a given execution time are also found decreasing, proportionately with the execution time. The mean latency in handling a request depends on the number of requests on the test execution queue. Hence, as the test queue decreases, the latency is also decreased.

TABLE 1. REQUESTS AND MEAN LATENCY

Requests	Requests per second	Mean Latency	No. of Hits per second	Response Time
3075	615	16.2 ms	427	7.4ms
6203	626	15.9 ms	361	6.8ms
9410	641	15.5 ms	483	6.2ms
Average	627.33	15.83	423.66	6.8ms

TABLE 2. THE RESULT OBTAINED IN LOAD TEST

Modules	Max Time (m)	Concurrency Level	Completed Requests	Total Errors	Total Time	Mean Latency
Admin	20	10	12955	0	20.00	15.4
Event	17	9	10762	0	17.65	14.6
Band Leader	15	9	10211	0	14.27	13.2
Band Manager	13	7	8955	1	10.27	10.6
Member	11	6	7054	1	9.27	9.4

TABLE 3. BAND'S FEEDBACK AND IMPROVEMENT

Bands	Feedback Knowledge per event			Improvement Parameters			
	No. of members responded	No. of audience turned out per event	Provided Feedback	Members turning in	No of Events organized	Audience positive feedback	Audience Negative Feedback
Band-1	8/25	300+	275	18/25	11	72%	28%
Band-2	4/12	210+	134	10/25	9	77%	23%
Band-3	7/18	130+	67	13/18	8	65%	35%

The functionality of the BMT collects enormous data in terms of text, which are both structured and unstructured, and audio/video files that are unstructured. The data forms that are communicated within the tool give a greater credibility for the bands to improve on their functioning style. The knowledge based storage and retrieval approach that is adopted using CSM and OSM objects has made the storage and retrieval of these data constructs easier and efficient. The results are given in Table 3.

VI. CONCLUSION

A Web tool is designed and implemented which can help the Music Bands and various Event Organizers to connect with each other using a client-server model. The application is a single-page application, thus making it faster for its response. The software used in Web Tool is MEAN stack which helps create the light-weight application. Nodejs is used for developing the APIs on the server-side. While AngularJs is used on the client-side. The database used is MongoDB which is the document-oriented database. The language used CSS or Cascading stylesheets and the framework of Bootstrap makes the application user-friendly and responsive.

Various access groups are created in the application. They include Admin, Band Members, Band Managers and Event Organizers. Each of the user has different activities and privileges according to these access groups. Each user has a profile. This user has be given privilege to manage his profile.

If the user belongs to Event-Organizer access group, he has the privilege to create new events and send an invitation to bands to perform in his event. The band manager has got the privilege to accept the invitation and give the details of that event to other band members.

The band members can send artefacts and audio or video files to the other band members for preparation of the upcoming events.

The Admin can monitor all the activities which are going on using this application. The Admin has the privilege to remove the user from the database for being inactive for a long time or for misbehaviour.

VII. REFERENCES

[1] Amjad Farooq and Imran Mir, “Design and Development of a University Web Tools”, International Journal of Multidisciplinary Sciences and Engineering, vol. 1, no. 2, November 2010.

[2] F. J. Muñoz-Rodríguez and G. Jiménez-Castillo, “Web app for a Remote Electronics Instrumentation Lab”, IEEE Conference on Technologies applied to Electronics Teaching, Spain, 2014.

[3] Yunhua Gu, Shu Shen, Jin Wang and Jeong-Uk Kim, “Application of NoSQL Database MongoDB”, International Conference on Consumer Electronics-Taiwan (ICCE-TW), 2015.

[4] Abdur Rahman.A and Chitra Devi S., “A Framework for Ultra-Responsive Light Weight Web Application Using Angularjs”, Online International Conference on Green Engineering and Technologies (IC-GET 2015), 2015.

[5] MongoDB database Documentation docs.mongodb.com

[6] MEAN.io. (2015) MEAN Full-Stack JavaScript Using MongoDB, Express, AngularJS, and Node.js. [Online]. Available: http://mean.io/

[7] Ravi Sharma and Shipra Ravi Kumar, “Strategies for Web Application Development Methodologies”, International Conference on Computing, Communication and Automation (ICCCA2016), 2016.

[8] Hepsiba Mabel, Justus S., Performance of the three-layered knowledgebase and its representation models: an empirical evaluation, International Journal of Knowledge Engineering and Data Mining, vol.3, no.3/4, pp.239 - 254, 2015.

[9] Hepsiba Mabel V, Justus Selwyn, A Review on the Knowledge Representation Models and its Implications, International Journal of Information Technology and Computer Science, vol. 8, no. 10, pp.72-81, October 2016.