International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 8 Issue 8 ǁ 2020 ǁ PP. 31-38

A New Approach for Developing SMART DEPARTMENT

Ajish Antony[1], Amal Alby[2], **Gokul Babu**[3], Jill Antony[4], Paul P Mathai[5]
[1]*Federal Institute of Science & Technology, India*
[2]*Federal Institute of Science & Technology, India*
[3]*Federal Institute of Science & Technology, India*
[4]*Federal Institute of Science & Technology, India*
[5]*Federal Institute of Science & Technology, India*

ABSTRACT
Smart Department System is a powerful system that provides users with a rich and responsive user interface and can collect information about system resources within the department. The system provides an appropriate database system to store this information. In addition, it provide overall monitoring of systems connected to servers in the department network. The administrator is granted permission to add or delete systems. The system allows different types of users to collect information about client systems in the department. In addition, administrator users can also shut down the system by monitoring the system, usage of processors, memory, etc. To be precise, it can be said that the Smart Department system is an efficient system that can be used to monitor and control connected systems.

KEY WORDS: *Server, Client, Monitoring, Controling,LAN*

Date of Submission: 15-08-2020 Date of acceptance: 01-09-2020

I. INTRODUCTION

As technology advances, the web becomes more and more important in our daily lives. In fact, everything we do today involves the use of the web. The main architecture associated with the Web is a client-server model, so the communication between the client and the server is our first concern. The client-server system divides the function of sharing information between the client and the server, thereby making the application development time shorter and shorter. The client is the requester, and the server is the service provider. In most client-server environments, data processing is handled by the server and the results are returned to the user to speed up performance. The client server system is so popular because it is used for different applications almost every day.

The client-server system can be defined as a software architecture consisting of a client and a server. The client always sends the request, while the server responds to the request. The client-server provides inter-process communication because it involves the exchange of data from the client and server, so that they each perform different functions.The main difference between a client-server and a peer-to-peer system is the client-server architecture.[1]

The Smart Department System works as a client-server model. A Web app and Android app are provided as a user interface. This system can be used to gather information of all systems connected to a server within a department. The system provides two login privileges, the admin and the user. Administrators have all authorizations, while users have limited authorizations. There is a root administrator who provides all authorizations and cannot be deleted. The user requests information about the client system from the server, and the server responds according to the user's visibility. Administrators can collect more information about the client system and can shut it down when the usage limit is exceeded.

II. RELATED WORKS

2.1 Remote Desktop Monitoring and Controlling

In today's era, every organization, whether it's the company or university wants to see the event executed by employees/students computer. To solve this problem, Remote Desktop surveillance and control becomes a reality, it continuously monitor every action used by anyone on the desktop. Remote desktop monitoring and control is based on the client server architecture, in which a PC or LAN or WAN will act as a server. With this server, the administrator or teacher can continuously view the activity of each client on the desktop or PC during any online test.[2] By monitoring each client or student system through the server or teacher computer, the teacher can easily judge whether there are illegal activities or activities outside the scope of student rights on the system. The concept of remote desktop monitoring is helpful for monitoring screens of multiple computers remotely in this particular scenario. This system monitor client computers connected to the server computer.It also performs remote operations to control the customer's machine.[3]

2.2 A Client Group-Server DBMS Architecture and Inter-Client Communication

Researches on client-server DBMS focused on Clientside data caching scheme based on LAN Client-server architecture. Here the study is on the problem of client data cache in WAN environment.[4] In a WAN environment, there are usually many clients gathered in a local area, through LAN, and often request the same data from the same remote server. To take advantage of this feature, Client group-Server DBMS architecture is considered, where Clients connected by various LANs. The client group communicates with the server through the agent, special customers belonging to this group and maintained. A type called Inter-Client Suitable for the communication cache (ICCC) of the client group server DBMS architecture.[5]

In ICCC, cached data can be shared. Through the communication between customers under the intervention of the broker, some simulation experiments are done to evaluate performance ICCC. The results show that the performance of ICCC is significantly better than Standard callback locking scheme used.[6]

In a WAN environment, investigation on WAN-based client-server DBMS environment was done. Instead of traditional LAN-based client server DBMS architecture, importance was given to client-server. In DBMS architecture, clients connect through each local area network zones form client groups and communicate with servers through brokers, special customers belonging to this group. Interclient communication is also used for data sharing among customers belonging to the same customer group are being exploited.[7]

2.3 Directly and Indirectly Synchronous Communication Mechanisms for Client-Server Systems Using Event-Based Asynchronous Communication Framework

The proposed a synchronous communication mechanism that can be used in the event-based communication framework (CM). Synchronous communication mechanism supports direct synchronous communication between the two clients through the server and indirect one-to-many simultaneous communication with multiple clients.[8]

To support synchronization Communication, CM uses a synchronization mechanism between the main thread and processing threads while maintaining a multi-threaded structure. The application can use both asynchronous and combine the CM's synchronous communication service according to its requirements. For performance analysis compared CM synchronous and asynchronous communication methods with qualitative and quantitative experiments. Through qualitative analysis, synchronous services can design application logic more intuitively than asynchronous services. From Quantitative experiments and also verified that the response time of the synchronization method is short. Although the difference in response delay is not significant, its advantages are not obvious compared to asynchronous methods.[9]

The synchronous communication Service mechanism in CM, which is asynchronous based on events Communication framework. CM also provides Synchronous and asynchronous communication services between a client and a server or multiple clients. CM has applied synchronization internal thread mechanism to support one to one, direct, one-to-one indirect and one-to-many indirect Synchronous communication scheme.For investigation synchronous and asynchronous features,conducted a qualitative analysis and quantitative experiments. Due to performance analysis, calling synchronous API is more effective than calling synchronous. Customers need to use a set of efficiency and intuitiveness[10]. Response delay synchronization API time is less than Asynchronous API. In future work,conduct practical case studies, can be applied to various types of synchronous communication. Method for verifying the effectiveness of synchronous communication. In addition,planning to extended requestreply mode. This is a communication Supports synchronous and asynchronous CM methods used to publish-subscribe model communication.[11], [12]

III. METHODOLOGY AND SYSTEM ARCHITECTURE

3.1 Methodology

The methodology of the system mainly includes three parts. Server, client and user interface. These components should be in a single network to work properly. The main purpose of the system is to collect the memory usage rate, CPU usage rate, processor usage rate and other information of the client system in the relevant department. Our method is mainly based on the client-server architecture and can be accessed using an appropriate user interface. For this, a server with a database is needed to store the information, and the client can connect to this. In order to collect information, you need to set up the client, which needs to be done using PHP scripts with the support of the Nginx server. The clientshould be able to provide general and detailed information to the server. Based on this, a client system will be established.

In order for users to obtain all information from the client system, a user interface will be developed. The user interface to be developed is a web application program for accessing the system through the web and an android application program for accessing the system from an android device. These interfaces should be connected to the server, and information should be requested from the server. The server should execute the request, forward it to the client system, and should return the client information to the user. The system should be developed in such a way that different types of users can access the system, such as admin and other users.

3.2 Architecture

System architecture mainly includes server, client and user interface.

A) Server: The server uses the Nginx server and MySQL database to support scripting in PHP. The Nginx server provides all the support for PHP code, while the MySQL database is used to store detailed system information. The database consists of two tables-user (to store all user information) and system (to store client system information).

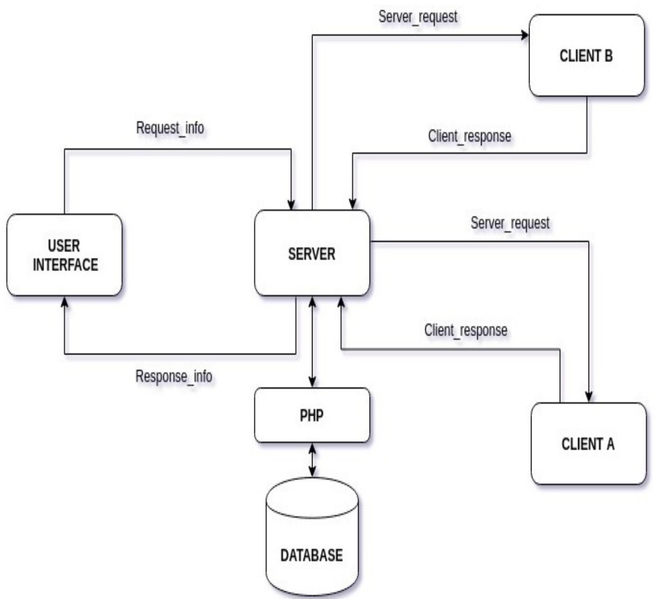


Figure 1: System Architecture

B) Client: Client side also uses PHP to write scripts, with the support of Nginx Server. These clients will connect to the server. The client system returns two types of information to the server according to the request. These are general and detailed information about the client system. To this end, two programs were coded on the PHP client.

C) User Interface: The Smart Department System has two user interfaces. The only thing is that we need to establish a proper connection between the server and this user interface, by setting the server URL.

Web App: Users can use this user interface to collect information from the client system through the server. This also provides the administrator users with the preparation of collecting detailed information from the client system, adding new users, and viewing a list of all users. The user interface also provides a graphical view of memory and processor usage.

Android App: Similar to web app, Android app also provide users with a better interface, to collect information from the client system. It also provides administrator users with all other regulations similar to web applications and can be installed in any Android device.

IV. SYSTEM DESIGN

Smart Department System is a system for collecting information from different systems. This way we can monitor all systems on one server. It allows the server to monitor each independently running system so that the server can be shut down when data usage is high by verifying it The system provides a web application and an android application, by which the system can be accessed from either a system or an android device. The basic architectural model we used in the building of our system is a client-server architecture.

The client-server architecture is the architecture of a computer network, where many clients request and receiving services from a centralized server. The users are provided with an interface that allows them to interact with server by providing requests. The server forwards this message to client systems and gather information[13] based on the request and response back to the user. Ideally, the server can provide standardized transparent client interface, and user need not be aware of the specifics of the system that is providing the service. This calculation model is particularly effective when the client and server are used separately.

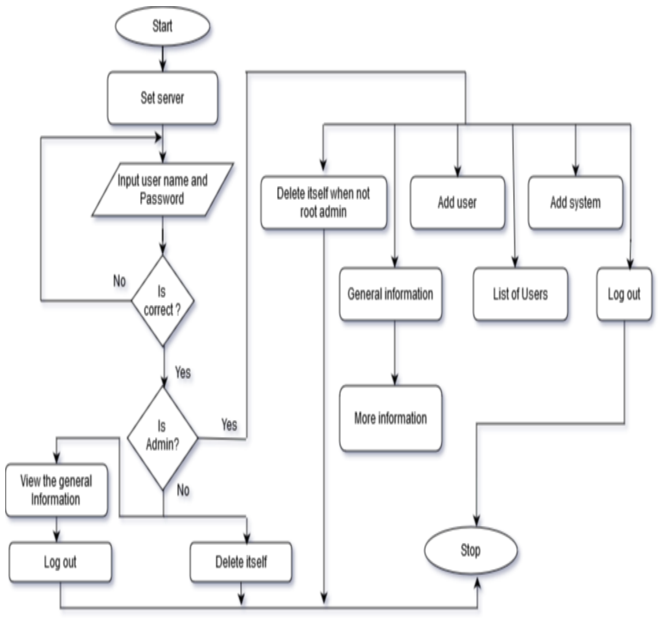


Figure 2: System Flowchart

The initial stage is to set up the server by providing the URL. Then, the user needs to provide the user name and password, and then check. The server checks whether it is an administrator or another user who provides its own authorization. If the logged-in user is not an administrator, the user has two privileges. That is, view the general information of all client systems and delete themselves, otherwise, if the user is an administrator user, the administrator user has the right to add new users, add new client systems, and obtain the detailed information of all users and can delete itself if it is not a root admin.

V. SYSTEM IMPLEMENTATION

At the implementation part, the main purpose was to collect information from clientr systems. For this, the Nginx server and PHP will be installed on the client. The program parts related to the client are written on PHP (ie, to get general information[14] and detailed information). Also a program for shutting down the system was written on the client. In addition to retrieving information, we also need to store this information. To this end, we build a server system with the support MySQL database. PHP is installed on the server side with the support of Nginx server. Server is given the whole responsibility to take care of every system and user. It also stores these details in the database. To handle the users and system, separate program are written on the client side. The users interact with server to get information and server calls the requested client. In both client side as well as in server side php is used with Nginx server for coding.

Basically, three PHP files are used on the client side: setup.php, overview.php, info.php. The necessary configuration settings are handled by setup.php. To get an overview of the system, we use overview.php and get more information about the system, such as average load, disk usage, and memory usage, we use info.php. In order to collect information for users, a web application and an android application are developed. Web App is one of the user interfaces developed with the support of JavaScript, HTML and CSS.

For each user, username and password will be provided. They can use them to log-in to the system and system provides them with client system information based on their privileges. The android application is another user interface of the smart department system[15]. It was developed in react native. Both interfaces provide information to users based on their privileges. The system can be accessed by two types of users (admin and user). The administrator is one who has all privileges and can collect all client system information.

For more information, the server will find and verify the request in the database. In case the request is for a valid client system, and the server sends the request to the system to get detailed information. MySQL is used to process the database in the server. There are mainly two tables, User tables and system tables. Create users, collect user information, delete users, log in and logout is handled by the user table. Add system, get information about system, delete system are handled by system tables.

5.1 Client-Server Model

1) Client: : The client is a program that runs on the local computer and requires the server to provide services. The client program is a limited program, which means that the service is started by the user and terminated when the service is completed. As mentioned before, we basically uses three php files, setup.php, overview.php, info.php. We also use different classes on the client side. The console class is used to get a better view of the output in the terminal. To obtain information about the CPU, ie, overview information and detailed information, we use the CPU class. To obtain an authentication token, we use a header class. To get information from the file ".env", we use key.php. In order to obtain memory information and arrange ordinary html files into json format by inserting http_status_code and required headers, we use memory and response classes. These classes maintain the overall work of the client. We also use certain functions in the overall coding of the customer section. In order to provide information about average load, average memory, CPU, we use sys_get_loadavg(). For more detailed information, such as using the publisher ID, release version, and codename lsb_release. To obtain information about disk usage, use the df-xtmpfs function.

2) Server: : A server is a program that runs on a remote computer and provides services for clients. When a client requests a service, the server opens the door to the request, but it never starts the service. The server program is an unlimited program, meaning that it will run indefinitely at startup unless there is a problem. The server waits for incoming requests from the client. When the request reaches the server, it responds to the request. MySQL is used to process the database in the server. In user.php, three methods are used: GET, POST and DELETE.

POST is used to create a new user. It also checks the conditions for creating new users. The condition we provide here is that the method should be post, the user must be logged in and the type must be equal to admin. Also create a new user and add it to the database user.php using a function register(). GET is used to forget the information about the user by passing the user name. DELETE is used to delete the user. The root administrator can delete all other users, but can’t delete itself. Administrators can delete users other than root admin, and can delete themselves. Ordinary users can only delete themselves. The other function of user.php is that it is used to collect information about all users for login and logout purposes. In system.php, we also use GET, POST and DELETE methods. Here POST is used to create a server or add a new system to the server. GET is used to obtain system information. DELETE is used to delete the added system and can be executed by the administrator.

Other functions of system.php are used to obtain basic information of all systems. Like the client, we also use certain classes on the server side. The console class is used to get a better view of the output in the terminal. The database class is used to establish a connection with the database and perform operations such as create and delete by executing queries. The key class is used to collect information from the ".env" file. It either checks the .env file, or checks the system environment[16], or finds abnormalities or terminations. The A remote class is used to process requests from the server to the client and collect information. The request class parses the request in the form required by each request. In order to organize ordinary html files into json format, we use a response class. Here we also use system classes to collect information from each system’s database. And use the remote class to display information to the user. Most importantly, a user class used to log in to the database, create and delete users.

5.3 Web App

This is a user interface for logging into the server. It provides information about the currently logged in user. That is, the type of user, its state, and memory. It also provides logout, user list, add user and system functions. It provides general information for all clients, and provides the function of obtaining more information and deleting the system. It also provides administrator users with more information, including graphical representations. Using this information, the server may shut down the system due to excessive use. Web application modules are developed with the support of HTML, JavaScript and CSS.

5.3 Android App

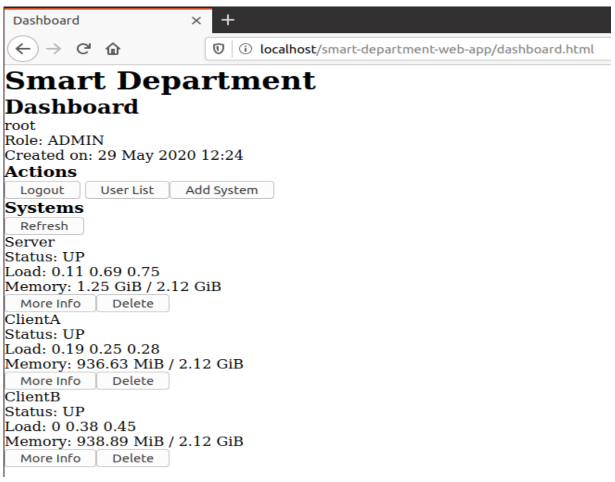
Android is a mobile operating system based on a modified version of Linux kernel and other open source software. It is mainly used for touch screen mobile devices such as smartphones and tablets. It is free and open source software. Its source code is called Android Open Source Project (AOSP), and it is mainly licensed under the Apache license.Android devices come with other proprietary software pre-installed, the most famous is Google Mobile Services (GMS), which includes core applications such as Google Chrome, digital distribution platform

Google Play and related Google Play services development platform. Android applications are developed using React Native, and the code is actually JavaScript code. Later we use react native application builder to build android application.

It is a user interface that can be used to log in to the server. It mainly provides two tabs, system and configuration files. The system provides general information about the client system and provides the client user with detailed information when needed. There is also a button for adding a new system. The configuration file provides information about the currently logged in user, user list, and conditions for adding users.

VI. EXPERIMENTAL RESULTS

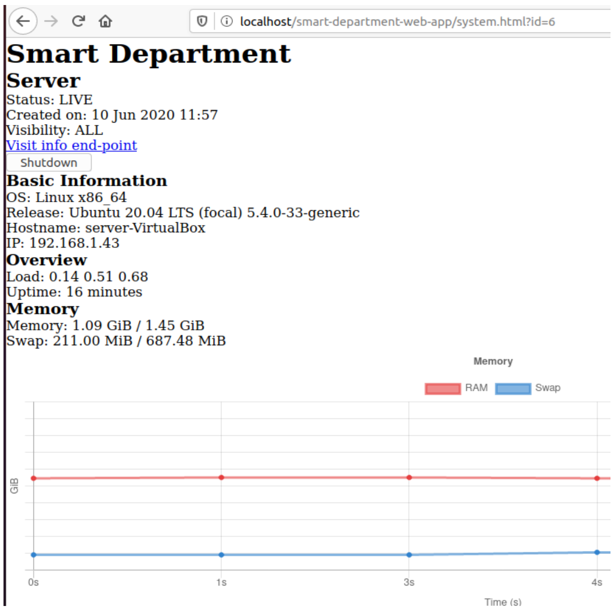
6.1 Web App Results



Smart Department
Dashboard
root
Role: ADMIN
Created on: 29 May 2020 12:24
Actions
Systems
Server
Status: UP
Load: 0.11 0.69 0.75
Memory: 1.25 GiB / 2.12 GiB
ClientA
Status: UP
Load: 0.19 0.25 0.28
Memory: 936.63 MiB / 2.12 GiB
ClientB
Status: UP
Load: 0 0.38 0.45
Memory: 938.89 MiB / 2.12 GiB

Figure 3: Web App Dashboard

Fig.3. shows dashboard of Web App. This gives user information, provision to logout, add user and list users, and general information of all systems.

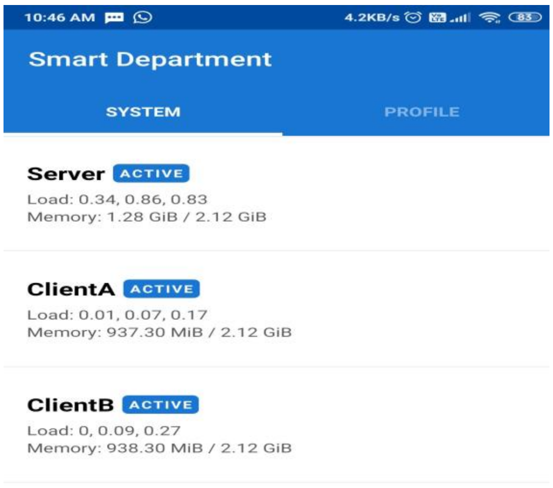


Smart Department
Server
Status: LIVE
Created on: 10 Jun 2020 11:57
Visibility: ALL
Visit info end-point
Basic Information
OS: Linux x86_64
Release: Ubuntu 20.04 LTS (focal) 5.4.0-33-generic
Hostname: server-VirtualBox
IP: 192.168.1.43
Overview
Load: 0.14 0.51 0.68
Uptime: 16 minutes
Memory
Memory: 1.09 GiB / 1.45 GiB
Swap: 211.00 MiB / 687.48 MiB

Figure 4: More Information on Web App

Fig.4. shows the Web App more information page. Information regarding memory, cpu, processor and all the detailed information of a system is provided with the graphical representation of memory and processor usage. Also a provision to shutdown system is provided.

6.2 Android App Results


10:46 AM 4.2KB/s 83

Smart Department

SYSTEM PROFILE

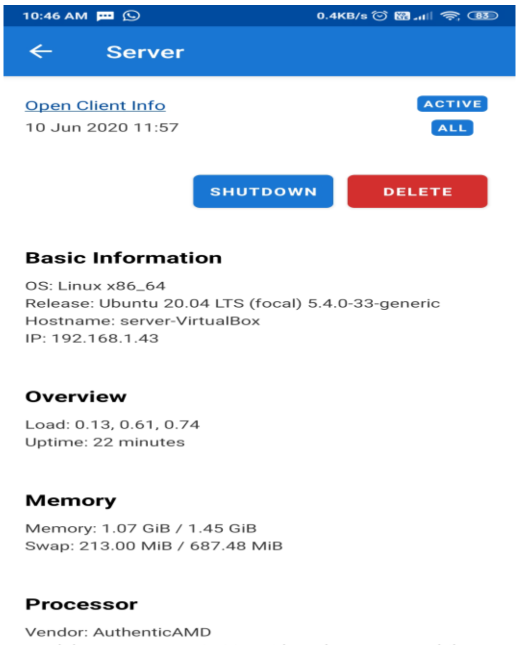
Server ACTIVE
Load: 0.34, 0.86, 0.83
Memory: 1.28 GiB / 2.12 GiB

ClientA ACTIVE
Load: 0.01, 0.07, 0.17
Memory: 937.30 MiB / 2.12 GiB

ClientB ACTIVE
Load: 0, 0.09, 0.27
Memory: 938.30 MiB / 2.12 GiB

Figure 5: System Details in Android App

Fig.5. shows the overview details of all the systems connected to the Smart Department system. This gives general system details such as system name, whether it is active or down, load and memory details.


10:46 AM 0.4KB/s 83

← Server

Open Client Info ACTIVE
10 Jun 2020 11:57 ALL

SHUTDOWN DELETE

Basic Information
OS: Linux x86_64
Release: Ubuntu 20.04 LTS (focal) 5.4.0-33-generic
Hostname: server-VirtualBox
IP: 192.168.1.43

Overview
Load: 0.13, 0.61, 0.74
Uptime: 22 minutes

Memory
Memory: 1.07 GiB / 1.45 GiB
Swap: 213.00 MiB / 687.48 MiB

Processor
Vendor: AuthenticAMD

Figure 6: More Information on Android App

Fig.6. shows the detailed information of a system. Only admins can access this and it includes an option to shutdown the system, delete the system, basic information, memory and processor information, etc.

VII. CONCLUSION

In the new era of computers and technology, analyzing performance and efficiency is very important. Smart Department is an interface which provides the user to evaluate the performance and efficiency of a system. It also allows users to collect and store information about all systems connected to a server in the department. Allows the administrator to monitor the client and if the data usage is high, you can shut down the system. The client-server architecture is regarded as the basic model. We have successfully developed a Web application and an Android application as the user interface.

The smart department is the idea generated by our research on generating efficient interface, used to collect information from multiple systems. We have studied several articles about Client server architecture and communication. Detailed study on asynchronous and synchronous Communication has been completed.

Simplicity, efficiency, convenience are the main features that we expected and we are proud to say that we met these standards without any shortcomings. Whenever one demand ensues, and it can be improved with better technologies that existed at the time, so that you don’t feel obsolete its existence has its meaning.

REFERENCES

[1]. T.Auer, M. Alan, M. C. M., S. Pezanowski, M. Stryker,HerbariaViz: **A web-based client–server interface for mapping and exploring flora observation data**,Ecological Informatics, In Press, Corrected Proof,Vol.18,pp.123-128,September 2010.

[2]. Harsh Mittal, Manoj Jain and Latha Banda, Harsh Mittal, Manoj Jain and Latha Banda, **“monitoring local area network using remote method invocation”**, IJCSMC, Vol. 2, Issue. 5, May 2013.

[3]. Wang ping, wany Zheng, **“IEEE, Design and Implementation of Open Computer Lab Monitoring and Management system”**. Computer and modernization, IEEE.11.pp.125-128,2007.

[4]. M. J. Carey, M. J. Franklin, M. Livny, and E. J. Shekita. **“Data Caching Tradeoffs in Client-Server DBMS Architectures”.** In Proceedings of the ACM SIGMOD Conference on Management of Data, pages 357–366, Denver, CO, May 1991.

[5]. Y. Wang and L. A. Rowe. **“Cache Consistency and Concurrency Control in a Client/Server DBMS Architecture”.** In Proceedings of the ACM SIGMOD Conference on Management of Data, pages 367–376, Denver, CO, May 1991.

[6]. P. Vaeková, I. Vaeková, I. erná, **Automated Computing of the Maximal Number of Handled Clients for Client-Server Systems**, Electronic Notes in Theoretical Computer Science, Vol. 260,pp. 243-259, January 2010 Pearson Education, 2007.

[7]. A. S. Tanenbaum and M. van Steen, **“Types of communication,”** in Distributed Systems: Principles and Paradigms, 2nd ed. New York, NJ, USA: Pearson Education, 2007.

[8]. M. Lim, **“Supporting synchronous and asynchronous communications in event-based communication framework for client-server applications,”** Int. J. Adv. Comp. Res., vol. 9, no. 40, pp. 1119, Jan. 2019. doi: 10.19101/ IJACR.COM16004.

[9]. M. Lim, B. Kevelham, N. Nijdam, and N. Magnenat-Thalmann, **“Rapid development of distributed applications using high-level communication support,”** J. Netw. Comput. Appl., vol. 34, no. 1, pp. 172182, Jan. 2011. doi: 10.1016/j.jnca.2010.08.003.

[10]. S. Tai and I. Rouvellou, **“Strategies for integrating messaging and distributed object transactions,”** in Proc. IFIP/ACM Int. Conf. Distrib. Syst. Platforms Open Distrib. Process., pp. 308330, 2000. doi: 10.1007/3-540- 45559-016.

[11]. D. Pakkala, P. Pakkonen, and M. Sihvonen, **“A generic communication middleware architecture for distributed application and service messaging,”** in Proc. Int. Conf. Autonomic Auton. Syst. Int. Conf. Netw. Services, Papeete, Tahiti, French Polynesia, Oct. 2005.

[12]. M. Tortonesi, N. Suri, C. Stefanelli, M. Arguedas, and M. Breedy, **“MOCKETS: A novel message-oriented communications middleware for the wireless Internet,”** in Proc. WINSYS, Setubal, Portugal, Aug. 2006, pp. 258267.

[13]. Mathai, Shelmy, Mathai, Paul P. and K. A. Divya. **"Automatic 2D to 3D video and image conversion based on global depth map."** *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. IEEE, 2015.

[14]. Abidha, T. E., and Mathai, Paul P. **"Reducing false alarms in vision based fire detection with nb classifier in eadf framework."** *International Journal of Scientific and Research Publications* 3.8 (2013): 50.

[15]. Mathai, Paul P., RV Siva Balan, and Ierin Babu. **"An efficient approach for item set mining using both utility and frequency based methods."** *International Journal of Applied Engineering Research* 12.12 (2017): 3470-3473.

[16]. Babu, Ierin, RV Siva Balan, and Paul P. Mathai. **"Envisaging the Academic Performance of Students."** *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE, 2019.