



# **iJRASET**

International Journal For Research in  
Applied Science and Engineering Technology



---

# **INTERNATIONAL JOURNAL FOR RESEARCH**

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume: 9      Issue: V      Month of publication: May 2021**

**DOI: <https://doi.org/10.22214/ijraset.2021.34189>**

**[www.ijraset.com](http://www.ijraset.com)**

**Call:  08813907089**

**E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)**

# Efficient Platform for Emergency Healthcare Services

Prof. Shubhangi Suryawanshi<sup>1</sup>, Aditya Mhatre<sup>2</sup>, Radheya Sarode<sup>3</sup>, Joanna Jadhav<sup>4</sup>, Omkar Fase<sup>5</sup>

<sup>1</sup>Assistant Professor, <sup>2, 3, 4, 5</sup>Student, Dept. of Computer Engineering, Dr. D. Y. Patil institute of technology, Pimpri, Pune 411018

**Abstract:** *Emergency healthcare services are the most time crucial services in which people operate, any error while executing these services decreases the chances of the victim's survival. There are many road accidents where the life of a human is lost due to mismanagement in the emergency services. The weakest link of the service comes out to be the initiation of the emergency services. Our goal is to strengthen the weakest link and enable the initiator to initiate the service within a single tap. This paper discusses how efficiently emergency healthcare systems can be initiated with the help of cloud computing and android application.*

**Keywords:** *Real Time System Architecture, Emergency Healthcare Services, Location Based Services, Database Management, Client-Server Communication.*

## I. INTRODUCTION

The term accident as frail it may seem, the amount of physical, economical damage it does every year is devastating. Road accidents have been one of the major reasons for deaths in India and the number of such accidents is increasing at a very rapid rate every year. One fact that stands out of all the facts is that middle-income countries contribute about 90% of the global accidents, despite having only 50% of the global vehicle share. 16% of all the annual death, and lead to a high rate of social and economic losses.

Among these, a significant amount of deaths occurs due to mismanagement of emergency services. Many things can go wrong while these services are operating, solutions to some problems are beyond the scope of any software implementation. However, the weakest link in this service is initiating the request of emergency services. Things that could go wrong during requesting the emergency service are

- A. Miscommunication between the two sides.
- B. The person who calls for an ambulance is unable to convey the exact location.
- C. Many times ambulance is not available and time is wasted calling another one.
- D. In the times of stress and shock, even the basic skill of communication and decision making is hindered.
- E. The person is not known to the nearest hospital and ends up calling the hospital which is far from the accident site.

These errors are highly likely to occur in most cases, and luckily enough this miscommunication can be solved with the help of software services and that too with very high efficiency and almost 100% success rate. To tackle all the mismanagements specified above is what this project is based on. The project consists of two android applications which will be the primary interface for common users and healthcare hospitals respectively. These applications are backed up with a fast server that sends the victim's precise location to the hospitals and also notifies a two-way acknowledgement to both the healthcare centre and the user who requests for the emergency service. This project aims to reduce the requesting time to its minimum (Just by one click) and providing the healthcare centre with all the necessary intelligence to reach the respective site of the accident in minimum time.

## II. LITERATURE REVIEW

Maintaining temporal consistency imposes time constraints on the database operations. The environment or an application has a specific requirement for the database reaction time, which can be seen as a time constraint. The performance of a database depends on how early the transaction is performed before the specified time constraint.[2]

In this paper, the author discusses developing a correct methodology for how the tasks on the cloud backend should be classified, how to identify the dimensions of workload and the determination of breakpoints in a cloud backend service.[5]

An approach is made to bundle multiple HTTP requests into a single request, assuming the client has a limited source of power, this method can save up to 50% power as compared to the HTTP requests sent individually.[7]

A detailed paper on how wireless technologies can be introduced in the field of emergency healthcare services. The paper also mentions what to expect from an emergency healthcare system like location tracking, load reducing techniques and many more. It also touches on the topic of sustainability of the system as per future requirements.[8]

This paper briefly revisits the client/server communication and computing, Thoroughly going through the different client/server architectures. A clear view of client/server architecture benefits in long-lasting business profits and a minimal migration.[3]

### III. METHODOLOGY

#### A. Concurrency in python SQLite

SQLite is a light database for local use and this database does not need a standalone SQL server. Most of the times SQLite database will be stored in the local memory of the system and operations will occur locally without any involvement of the server. The sqlite3 module was written by Gerhard Häring. SQLite is best for any small application which barely uses any concurrent read and write operations. SQLite is fast and reliable and it requires no configuration or maintenance. It keeps things simple. However, SQLite has the following disadvantages.

- 1) SQLite does not perform well in a network filesystem as it's latency issues of the filesystems.
- 2) SQLite does not perform too good in a heavy and write-intensive application.
- 3) An SQLite database is limited in size to 281 terabytes.
- 4) SQLite allows unlimited read operations however, it does have a cooldown time for write operations(write operations gets locked after a certain number of operations).

People still choose SQLite because many practical applications require storage and processing which is far below the actual limitations of SQLite. Although we need to perform concurrent read and write operations which is a major drawback of SQLite, the solution came up with the latest version of SQLite3. This problem can be solved by the new "Write-Ahead Log" option (referred to as "WAL") is available. WAL provides more concurrency as readers do not block writers and a writer does not block readers. Reading and writing can proceed concurrently. To enable WAL "PRAGMA journal\_mode=WAL" query needs to be executed after connecting the database. This query enables WAL and allows for concurrent read/write operations in the database.

#### B. Location Extracting from Google Maps

For acquiring user location, we use GPS of android device. Android app needs user permission to access GPS. It provides location attributes such as Latitude, Longitude and altitude but here we require only latitude and longitude attribute for determining user location. These will be sent to the server for further processing.

#### C. Determining Emergency centres in Vicinity

All the information and locations of Emergency centres are already recorded and stored in the database while the process of registration. When a request for help arrives with coordinates of the accident site, they are used to find nearby centre using the following formula:

$$D = \sqrt{(D1 - L1)^2 + (D2 - L2)^2}$$

Where,  $L1, L2$  are latitude and longitude of the affected person and,

$D1, D2$  are latitude and longitude of the healthcare centre.

Note that, distance calculated here is in Km. After that, it is verified that if the distance is less than a certain threshold to mark the particular centre as the nearby centre from the site.

#### D. Socket Based Communication

Sockets are commonly used for client and server interaction. The socket API is often used with the TCP protocol to provide a connection-oriented service called a reliable byte stream, which is very reliable as there are acknowledgements passed from both sides. To achieve a TCP/UDP socket connection the server must establish an address to which the client can connect. When an address is established then the server listen to any request which is addressed to it. The client-server communication occurs when the client requests a specific service from a specific port. The server performs the client's request and sends the reply back to the client. This is the most reliable and easy way to communicate with the server. In our case of communication, the client connects to a total of 8 different services via 8 specified ports. The client successively sends a request to the next specified port until the whole process is completed.

#### IV. ARCHITECTURE

A client-server architecture based on a connection-oriented TCP service is implemented for a reliable connection. The architecture consists of a python server and an android application.

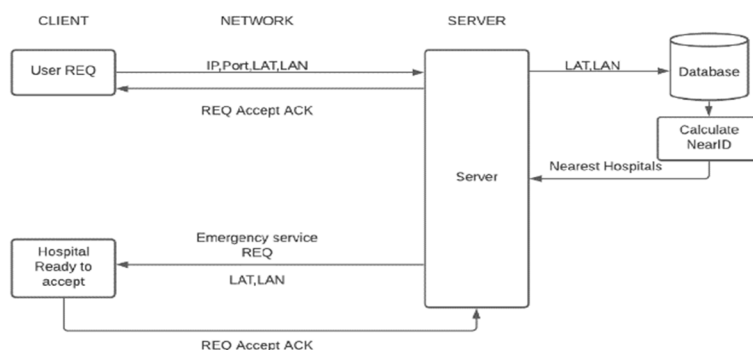


Fig. 1.1: Architecture

The services which are provided by the server are

- A. Login
- B. Signup
- C. Receiving Current Location
- D. Calculating nearest healthcare centres.
- E. Acknowledgement Services.
- F. The Next topic describes the communication methods used.

This is the more optimized choice of the architecture as compared to our previous architecture design. As the server receives the latitude and longitude from the user, the nearest healthcare centre/'s in the vicinity of 20 Kilometre is listed in a database and then Latitude and longitude are stored in the database because storing the location information beforehand would produce a delay in enlisting nearest healthcare centres to the database.

Further, As the database is updated with the list of healthcare centres another service picks up the information and notifies respective authorities of the centres. The first centre to respond acknowledge the server on the specified port number and the server responds by giving the GPS location of the user along with the suggested route on google maps.

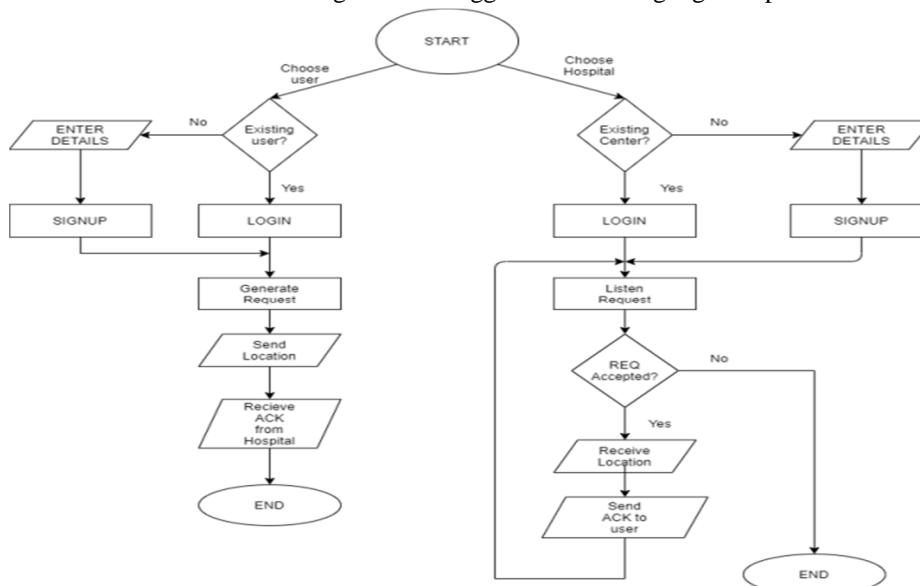


Fig. 1.2: Input Processing and Control flow Via Flow chart



The above diagram depicts the typical flow of the control.

- 1) *Case 1:* When a user launches the application.
  - a) If the user is already registered then head to the sign-in option.
  - b) If a user is not registered then enter details and register/sign-up.
  - c) User can initiate the service anytime when logged in.
- 2) *Case 2:* When a Healthcare centre launches the application.
  - a) If the centre is already registered then head to the sign-in option.
  - b) If a centre is not registered then enter details and register/sign-up.
  - c) Center now can listen to any incoming request and accept or reject the request.
  - d) If the healthcare centre rejects then no more future requests will be sent to the centre. The centre may start listening again when vacant or ready.
  - e) If the centre accepts the request then the location of the user is sent to the ambulance operator. The operator can then just use the preloaded google maps as the server will provide a direct hyperlink to the google maps application

## V. LIMITATIONS

- A. When the internet is off/No service.
- B. When any physical damage of device affecting malfunctioning of device.

## VI. CONCLUSION

The overall system when tested in real-time takes 5 seconds to notify the healthcare centre and if the centre responds immediately, the whole procedure will execute successfully in approximately 10 seconds. Again this time is the best-case scenario, human error may occur extending the overall time. One issue that will be solved in the next version is dynamically increasing the. Hence, the project is on the way to completion which when finalized and deployed will be able to reduce the requesting time to mere seconds and will be successful in giving all the required intel to the respective healthcare centres.

## VII. ACKNOWLEDGEMENT

We like to share our sincere gratitude to all those who are helping us in the completion of this project. During the work, we faced many challenges due to our lack of knowledge and experience but these people have always helped us to get over all the difficulties and will help us with their valuable guidance in the completion of this project.

## REFERENCES

- [1] Evans L. The new traffic safety vision for the United States American Journal of Public Health, 93 (9) (2003), pp. 1384-1386
- [2] Krithivasan Ramamritham, Real-time databases [https://www.researchgate.net/publication/225994091\\_Real-time\\_databases](https://www.researchgate.net/publication/225994091_Real-time_databases)
- [3] Carl S. Guynes, John C Windsor, Revisiting Client/Server Computing. [https://www.researchgate.net/publication/279647157\\_Revisiting\\_ClientServer\\_Computing](https://www.researchgate.net/publication/279647157_Revisiting_ClientServer_Computing)
- [4] Dimitri Yatsenko, Jacob, DataJoint: managing big scientific data using MATLAB or Python. <https://www.biorxiv.org/content/biorxiv/early/2015/11/14/031658.full.pdf>
- [5] Asit K. Mishra, Joseph L. Hellerstein, Towards Characterizing Cloud Backend Workloads: Insights from Google Compute Clusters. <https://www2.cs.duke.edu/courses/fall13/cps296.4/838-CloudPapers/Appworkload.pdf>
- [6] Y. S. Yilmaz, B. I. Aydin and M. Demirbas, "Google cloud messaging (GCM): An evaluation," 2014 IEEE Global Communications Conference, Austin, TX, 2014, pp. 2807-2812, doi: 10.1109/GLOCOM.2014.7037233.
- [7] Ding Li and William G. J. Halfond, Optimizing Energy of HTTP Requests in Android Applications. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.718.648&rep=rep1&type=pdf>
- [8] U. Varshney, "Pervasive healthcare," in Computer, vol. 36, no. 12, pp. 138-140, Dec. 2003, doi: 10.1109/MC.2003.1250897.
- [9] Arindam Banerjee, Prateek Agrawal, R. Rajkumar, Design of a Cloud Based Emergency Healthcare Service Model. [https://www.researchgate.net/profile/Rajasekaran\\_Rajkumar/publication/289395860\\_Design\\_of\\_a\\_cloud\\_based\\_emergency\\_healthcare\\_service\\_model/links/59665e3d458515e9af990fd3/Design-of-a-cloud-based-emergency-healthcare-service-model.pdf](https://www.researchgate.net/profile/Rajasekaran_Rajkumar/publication/289395860_Design_of_a_cloud_based_emergency_healthcare_service_model/links/59665e3d458515e9af990fd3/Design-of-a-cloud-based-emergency-healthcare-service-model.pdf)
- [10] Mareike Glöss, Moira McGregor, Designing for labour: Uber and the on demand mobile workforce. <https://doi.org/10.1145/2858036.2858476>



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)