Quick-Witted Reciprocity of Blood for Emergency Situations

Mr. C. Sabarinathan[1], Mr. K. Sai Prasanth[2], Miss Anishka Patwa[2], Mr. Karan Japee[2]

[1]Assistant Professor, Department of Computer Science and Technology, SRMIST, SRM Vadapalani, Chennai, India

[2]Student, Department of Computer Science and Technology, SRMIST, SRM Vadapalani, Chennai, India

Abstract

A blood bank is a complex where the blood or plasma is collected from donors, separated, stored and prepared for transfusion to recipients. It is sometimes referred to as a collection center. A Traditional blood bank system is all about working physically which is not capable all the time particularly, at emergency scenarios. The motive is to create an application and provide blood feasible whenever the requirement is given by an individual, even in emergency cases. We use Global Positioning System (GPS) to manage the blood bank using an android application. The application can be used in a wide range of smartphones which makes it more efficient and convenient. In this project, we develop two android based applications and one intermediate server. The first application is for blood requester, another is for a blood donor, and the server contains the previous recipient's database. Be it a requester or donor he/she has to register themselves before placing any request or accepting any requests. Blood Requester can send a query to the server for specific blood units via Google Cloud Messaging (GCM), the server processes the query and notifies the user whose data satisfied for the query. In the next scenario, the application sends the Gps coordinates and contact number to the requester, only if the blood donor accepts the request. We also use machine learning to classify donors. We are also concerned with the privacy of the donor. The data of the donor is not made visible to anyone until and unless he accepts the request.

***Index Terms* - Blood bank, Machine Learning, Gps, Cloud, Web Technology**

Introduction

Blood is the stealthiest and intriguing substances. It is a combination of plasma and cells that help in circulating oxygen, hormones, and sugars. It contributes 7% of total body weight. In remote areas, it is very hard to find blood banks because there is no proper system and information which fulfills the need. The main motive is, to provide everyone with a system which they can rely on, particularly in emergency situations. It provides a platform where the user can get information about the donors in their area depending on their geographical location using Gps technology. This paper emphasizes a brief knowledge of the functioning of the blood bank system and the various facilities provided by the system [1]. This is made by using various technologies like Android, machine learning, cloud and web mechanics [2]. The donor's privacy is protected by using the AES algorithm which is more efficient security algorithm [3]. The blood bank management system involves the following entities:

- **Requester:** Any individual who is in need of blood unit places the request. The need may arise because of Causality, Operations, and any other emergency situations.
- **Donor:** Any individual who donates blood which can be used in a transfusion. He/ She has to be healthy before the donation. There should not be any chronic diseases.
- **Blood Bank:** It is a complex where the blood or plasma is collected from donors, separated, stored and prepared for transfusion to recipients.

The user can access the application in two platforms i.e. web and Android [2]. It helps the user to switch between the applications with whatever he is comfortable. The application has an administrator is the trusted in-charge for gathering, sorting the information based on the location, blood category respectively. This information will be stored in the repository which is common for all and is stored in the database [4]. The requester can get the blood from the location he is requesting from [5] [6]. If the requester cannot get the blood from the location, then his/her request would be transferred to the nearby blood banks and hospitals.

Related Works

In Blood bank information system using Android application " byNeetu Mittal et.al[1] proposed an Information System using an android application. It consists only a database in which the people can register and search for the donors from the database. It is updated whenever the user registers himself in the application. They have methodologies like MY SQL, Android.

In "A Secure Cloud Computing Based Framework for the Blood bank" by Shreyas Anil Chaudhari et.al[2] proposed an Information System using an android application. It consists only a database in which the people can register and search for the donors from the database. It is updated whenever the user registers himself in the application. They have used the web application and Android application which is connected to the cloud.

In "Data security using authenticated encryption and decryption algorithm for Android phones" by Abhishek Vichare et.al[3] proposed an android application that will encrypt and decrypt data automatically. They attained this by using the AES algorithm, Android.

In “Automated online Blood bank database” by Muhammad Arif et.al[4] proposed an automated online database where the users can register themselves as donors. They attained this by using web page developed in HTML, CSS, JAVASCRIPT, and PHP.

In "Travel management system using GPS & geotagging on Android platform" by Amrah Maryam et.al [5] proposed a travel management system using GPS that access user’s current location and get information on their travel needs. There is an administrator to respond to the queries of the users.

In "Survey of Cloud Messaging Push Notification Service" by Na Li; Yanhui Du; Guangxuan Chen et. al [6] proposed a push notification service using google cloud messaging service. This is mainly used for communication between a web application and Android application. It can also be used in Apple push notification services.

In “A predictive modified round-robin scheduling algorithm for web server clusters ” by Xu Zongyu; Wang Xingxuan et. al [7] proposed a predictive system helps in managing the load on the servers and leads to the high performance of the server. The high performance is important in a client-server mechanism.

Technologies Used & Procedure

MySQL& Google Cloud Messaging (GCM):

This is a notification service used in sending push notifications from the server to the user. It mainly helps in the communication with the server and the end device. The connection will be between the web page and android application. The request is placed only one time and the server responds for the placed request. Whenever there is new data the server reverts back with the new data and there is no need to send requests often.

We used PHP as a server-side language and MYSQL as a server-side database language. The course is provided by the Android GCM.

1. Any device which is going to request the data must follow the pattern i.e.
2. The device must send its device Id to the server for registration.
3. The server acquires the device Id and reverts back with registration Id.
4. Again the device sends the registration Id to the server and the Id is saved to the DB.
5. Then, we prepare the functions using Php and connect it to the MYSQL database.
6. Before using the GCM, we need to acquire the API key so, as to integrate.
7. Then we create functions for push notifications that can send and receive requests and responses from the server.

Global Positioning System (GPS)

Global Positioning system plays a major role in the location services. It is a satellite-based communication service that provides time and location in any weather condition.

Steps to get the location of the user in android

1. Every Android application has a manifest file which is mainly used for giving permissions for the application services.
2. We need to give necessary permissions for the location update.
3. Location Manager instance created and the location services are enabled.
4. Location Listener then retrieves the location of the individual.

Proposed System

With this online management application, any individual/organization that is willing to donate blood can follow the same process of registration. Furthermore, the normal consumer can also request through this application. Administrator holds access to make any addition, deletion, and modification if required.

A distributed architecture having centralized database storage is being followed by our online blood donation management. The user interfaces have been built using Java. My-sql is used for the database necessities. SQL provides a methodology for database integration. The Security standards are maintained, and the data preservation is assured within the application.

Also, the communication will be between the client and the server. There is no chance for mediators. This type of technology is called Push Technology. It helps in the appropriate use of the data.

There are 337 million smartphone users in India as of 2018 which is a known fact. We proposed an android application, which can be used in any smartphone. Geographical Information System is an added advantage that an android provides. With the help of gis, the data can be transmitted between the web application and android application efficiently via a wireless network. Anyone can search easily for donors in a particular location. It mainly helps in emergency situations. In some situations, the requester might not be aware of the location, so by using this he/she can request the blood unit directly from his location.

The application is pretty adaptable, scalable and ready to use for any complex needs. Thus, this is an escalating method that aids not only people but also organizations, hospitals, etc.

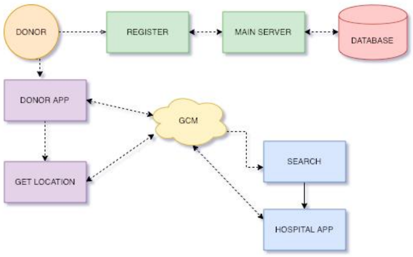


Fig. 1 System Architecture Diagram

Modules

There are basically 4 technical modules

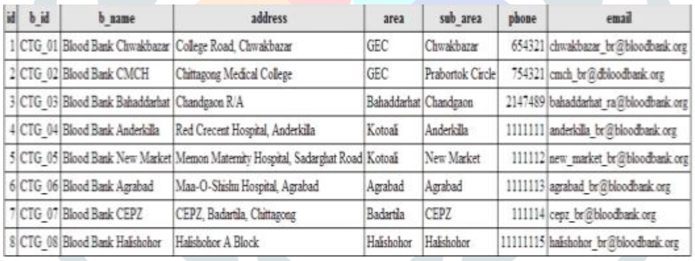
- User Registration
- Blood Request
- Alerting the donor
- Tracking the donor

User Registration

In this module, both the individuals (requesters/donors) register themselves by filling the basic details like Name, age, gender, blood group, address, and other information. The details are stored in the database by using the following query

INSERT INTO tables (Name, Age, Gender, Blood Group, Address)VALUES (‘John’, 21, ‘Male’,‘B+’, 'vadapalani’);

Blood group will be the primary key. Primary key makes the whole table unique and helps to locate the table easily during any searches. We will classify them according to the blood groups and the location. So, when a request has been placed it makes the system efficient during emergency situations.

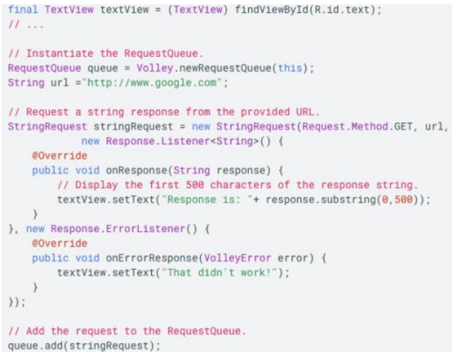


id	b_id	b_name	address	area	sub_area	phone	email
1	CTG_01	Blood Bank Chwakbazar	College Road, Chwakbazar	GEC	Chwakbazar	654321	chwakbazar_br@bloodbank.org
2	CTG_02	Blood Bank CMCH	Chittagong Medical College	GEC	Prabortok Circle	754321	cmch_br@dbloodbank.org
3	CTG_03	Blood Bank Bahaddarhat	Chandgaon R/A	Bahaddarhat	Chandgaon	2147489	bahaddarhat_ra@bloodbank.org
4	CTG_04	Blood Bank Anderkilla	Red Crecent Hospital, Anderkilla	Kotoali	Anderkilla	1111111	anderkilla_br@bloodbank.org
5	CTG_05	Blood Bank New Market	Memon Maternity Hospital, Sadarghat Road	Kotoali	New Market	111112	new_market_br@bloodbank.org
6	CTG_06	Blood Bank Agrabad	Maa-O-Shishu Hospital, Agrabad	Agrabad	Agrabad	1111113	agrabad_br@bloodbank.org
7	CTG_07	Blood Bank CEPZ	CEPZ, Badartila, Chittagong	Badartila	CEPZ	111114	cepz_br@bloodbank.org
8	CTG_08	Blood Bank Halishohor	Halishohor A Block	Halishohor	Halishohor	11111115	halishohor_br@bloodbank.org

Fig. 2 Sample Donor Database

Blood Request

When a particular individual tries to request blood of a particular group, the nearby donors are prioritized in terms of location. Creating a request we use volley method. Volley creates a request queue and passes it to the request objects. Before sending requests we need to give permission to the Internet access (android.permission.Internet). The user can also cancel the requests depending on his availability. By using onstop() method we can attain this.



```
final TextView textView = (TextView) findViewById(R.id.text);
// ...

// Instantiate the RequestQueue.
RequestQueue queue = Volley.newRequestQueue(this);
String url ="http://www.google.com";

// Request a string response from the provided URL.
StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
            new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        // Display the first 500 characters of the response string.
        textView.setText("Response is: "+ response.substring(0,500));
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        textView.setText("That didn't work!");
    }
});

// Add the request to the RequestQueue.
queue.add(stringRequest);
```

Fig. 3 Code Snippet for sending requests online

Alerting the donor

When the donor gets a notification from a receiver, it's up to the donor to accept the request. Only when the donor accepts it, the details are given to the receiver for further coordination. Firstly, we need to add the dependencies and give necessary permissions in the manifest file. Create an instance using subclass Instance service() and overriding it by using on handle() intent. Create InstanceIDListenerService class to create the request instance. It can be achieved by using the following code

```
java public class TokenRefreshListenerService extends InstanceIDListenerService{ @Override public void
onTokenRefresh() { Intent i = new Intent(this, RegistrationService.class); startService(i); } }
```

Then we need to start the registration service and display push notifications.

Tracking the donor

After the request gets accepted, the donor's location can be tracked by the receiver so that the further processes can be initiated. The location can be obtained by using an android's locationlistenerclass(android.location.LocationListener).Location_Manager.GPS_PROVIDERdetermines location using satellites. It returns the location of the user .onLocationChange() – updates the changed location of the user.

Methodologies

Following algorithms are used in shaping up the project

- Round Robin Algorithm
- Advanced Encryption Standard (AES) Algorithm
- Naïve Bayes Algorithm

Round Robin Algorithm

WHY? – The CPU is shared equally for all the process. All the process get the same priority as others.

This is a scheduling algorithm which is used by the CPU during the execution of the process. This is similar to first come first serve. It has a specific time which is called quantum time. Quantum time is the fixed time where the given process runs for that time. Basically, using the burst time and the number of the processes we can easily find out the average waiting time of each process. This will help us to maintain the queue without any barriers and serve equally in the given quantum time. In our project, we will place the requests in the queue. When the donors receive the requests, he will only see the first request, if he accepts then he will accept the request. Otherwise, he will be provided with other requests in the queue until he accepts. Accepting the requests also depends on the location of the donor. Location manager class helps us to get the latitude, longitude of the user.

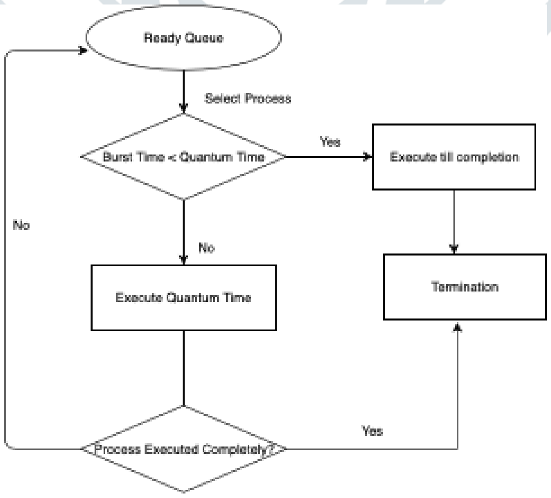


Fig. 4 Round Robin Algorithm

Advanced Encryption Standard Algorithm

WHY? – AES is more secure when compared to DES and 3DES. It enables faster encryption.

1. To protect the personal details of the users we used AES algorithm. It is a symmetric encryption algorithm.
2. We need to perform byte substitution which can be done with 16 input bytes referring the S-box. It results in a 4*4 matrix.
3. The matrix rows are shifted to the left. There are some basis shifting rules we need to follow.
4. We will mix the columns of the matrix using a special mathematical function. This results in a 16-byte matrix.
5. We will consider the 16 bytes as 128 bits and perform XOR to get the 128-bit round key. The output will be the ciphertext. The reverse order of encryption process is the decryption process.

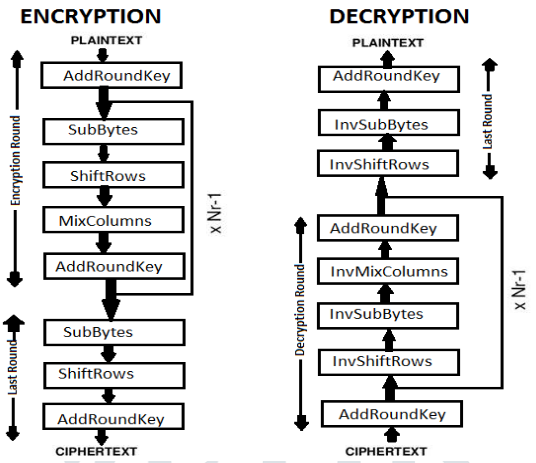
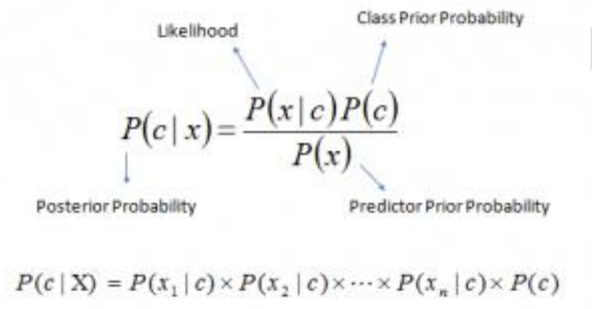


Fig. 5 – AES algorithm Process

Naive Bayes Classification

WHY? – It accepts continuous and discrete data. This requires less training data and is very efficient in the given data.

This is algorithm is supported by Bayes theorem. It assumes the individuality among the predictors. It searches for a specific feature of a category that is dissimilar to the opposite feature. It can be built easily and convenient for huge data sets. It provides some way of computing posterior likelihood P(c|x) from P(c), P(x) and P(x|c).



$$P(c\mid x)=\frac{P(x\mid c)P(c)}{P(x)}$$

$$P(c\mid X)=P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

Fig. 6 - Bayes Formula

Where,

- P(c|x) = posterior likelihood of the category for a given predictor.
- P(c) = class’s previous likelihood.
- P(x|c) = likelihood of the predictor for a five category.
- P(x) = predictor’s previous likelihood.

Result

By using this platform blood donation can happen more quickly by eliminating the manual work and saves a lot of time in emergency situations.

Future Work

Currently, the application works with a viable internet connection. In future, the application could be made offline and the communication must be through SMS which can be beneficial for the rural parts of the country, where network coverage is limited.

Conclusion

Thus, in this project, we find the nearest blood donor and track his details through the application and collect the blood during emergency situations. It helps in saving many lives.

Acknowledgment

The authors gratefully acknowledge Prof. C. Sabarinathan from SRM Institute of science and technology for their great contributions in writing help, language facilitate and coming up with the design and providing correct steerage in making this work.

References

[1]NeetuMittal ; Karan Snotra, “Blood bank information system using Android application”, (RDCAPE), 2017 International Conference.

[2]Shreyas Anil Chaudhari ;ShrutikaSubhashWalekar ; Khushboo Ashok Ruparel ; VrushaliMilindPandagale, “ A Secure Cloud Computing Based Framework for the Blood bank”, (ICSCET), 2018 International Conference.

[3] Abhishek Vichare ; Tania Jose ; Jagruti Tiwari ; Uma Yadav, “Data security using authenticated encryption and decryption algorithm for Android phones”, (ICCCA), 2017 International Conference.

[4] Muhammad Arif ; S. Sreevas ; K. Nafseer ; R. Rahul, “Automated online Blood bank database” (INDICON), 2012 Annual Indian Conference.

[5] N. Adarsh m, J. Arpitha , Md. Danish Ali ,N. Mahesh Charan , Pramodini G Mahendrakar,” Effective blood bank management based on RFID in real time systems”, Embedded Systems (ICES), 2014 International Conference.

[5]AmrahMaryam ; Nadia Siddiqui ; Mohammed A. Qadee, “Travel management system using GPS & geo tagging on Android platform”(WOCN),2016 International Conference.

[6] Na Li ;Yanhui Du ; Guangxuan Chen, “Survey of Cloud Messaging Push Notification Service”, 2013 International Conference on Information Science and Cloud Computing Companion.

[7] XuZongyu ; Wang Xingxuan , “A predictive modified round robin scheduling algorithm for web server clusters”, (CCC) 34th Chinese International Conference.

[8] P. Mathiyalagan,”Use of fuzzy TOPSIS techniques for selection of best alternatives of blood bank Supply chain” ,Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference.

[9] BalaSenthilMurugan L, AnithaJulian,”Design and implementation of Automated Blood Bank using embedded systems”, Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference.

[10] L.BalaSenthilMurugan, AnithaJulian,”Design and implementation of automated blood bank using embedded systems”,Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference.

[11] W. Domfang , M. La Raja ,F. Bellato ,R. Musi,”Modeling medical equipment standards for blood banking at different levels of health care system in countries with limited resources: the case of Cameroon”,Appropriate Healthcare Technologies for Low Resource Settings (AHT 2014).

[12] Siva Shanmuga and N. Ch. S. N. Iyengar, "A Smart Application on Cloud-Based Blood Bank,” Journal of Computer and Mathematical Sciences, Vol.7 (11), 576-583, November 2016.