

An ERP Based Blood Donation Management System for Hospital and Donor

Ghulam Muhammad¹, Hamza Asif¹, Farrukh Abbas¹, Imran Memon¹, Hadiqua Fazal¹

Abstract:

The objective of this paper is to develop a mobile blood donation management system application. This paper an android based application development technique by using an ERP model database management system. The techniques involve using mobile development IDEs and adequate APIs to have desired functionalities. There are two main mobile developing platforms present in the world iOS and Android. We have developed our application on Android OS. Different applications were surveyed and used to explore the presently available features to the end-user. After trials and research, the outline was made to what extent should go and developed. As the system is developed for hospitals and donors, the hospital puts a request for blood groups on the application. The system then finds nearby users online with the requested blood groups, if anybody required for his / her relative he or she can use this application for find donors who are available or offline with full information of donor if donor or user is interested in donation then he has done to the requester. Users' locations are tracked in real-time. If the users proceed to the hospital, the tracking shows if they are coming or not.

Keywords: *Android GUI, ERP Model, Hospital model, Donor model*

1. Introduction:

Nowadays, health technologies are developing rapidly. Incurable diseases are cured with the latest technology and medical research. With the rapidly developing technologies, medical professionals must cope up with changing trends and technologies. These technologies require fast response from professionals [1,2,3,4].

Such technology is a blood donation management system. A system that is capable of catering multiple users at a time. Hospital operations which include surgeries, transplantations, and medical processes, require a blood supply to facilitate patients undergoing those processes [5,6,7,8]. A

blood donation management system can manage information of users, track their arrival to hospitals, informs about users online on the system with the registration of authentic user cell number and hospital related information based on google map location [9,10,11].

2. Problem Statement

Blood arrangements are uncertain. People who committed to donate often change their plan to donate. As a result of the survey done, we found that conventional apps just register users and donors [12,13,14]. The user puts requests and donors reply to those requests on the application. It is then not guaranteed

¹ Department of Computer Science, Bahria University Karachi, Pakistan
Corresponding Author: ghulammuhammad.bukc@bahria.edu.pk

Table 1. Comparison of Applications

Applications on playstore								
APPS	Sms code	User Profile	User profile related services	User contact details	Limitation Factor of 3 months	Availability Nation wide	Availability Internationally	Location Provision
Blood Community		✓	✓	✓				
Blood Bank Pakistan	✓		✓	✓	✓	✓		
Blood Donors		✓	✓	✓		✓		
Hayaat		✓	✓	✓		✓		
LIFE	✓	✓	✓	✓			✓	
Revive						✓		
Tahir Blood Bank	✓	✓	✓	✓				
Life Saver		✓		✓				✓

Our Blood Donation Management Application	
FEATURE	YES/NO
SMS code verification	YES
User Profile	YES
Profile related Services	YES
User Contact Details	YES
Limitation Factor of 3 months	YES
International Availability	NO
Location Provision	YES

whether the commitment done between the user and donor is fulfilled or not. On further analysis it was found that this posed some of the problems which were the following:

- 1) Either the user is authentic or not. Is he/she really having the need for blood?
- 2) Criminals can misuse the application. Invite people to donate and then loot them.
- 3) Donors show their will to donate on the application but do not proceed for donation hence wasting the time of users.

3. Related Work

We surveyed 8 mobile blood donation management applications that are currently being used. All the applications have most of the features in common mentioned below. We will discuss some of them in this section

These features include user registration through SMS verification, user profile, profile related services, real-time tracking, user personal information, limitation factor, availability countrywide, availability internationally, and location provision [15,16,17,18]. Table 1 shows the comparison

of present applications that we surveyed. This provides a bird's eye view on application features. Indicating what they are capable of and whatnot.

As we can see the comparison in the below table, there is no application with all features. Our aim and objective are to not only cover these deficiencies but to develop with more features.

After the survey and research, the architecture of two side applications was selected. The hospital will have its own application while donors will have their own application. This is discussed separately in the subsections below.

3.1. User Side App (Hospital)

Applications first module, which is the main module, is the user's (HOSPITALS) side application. Hospitals will use the app so that the donors have trust and they know they are in safe hands. Any person in need of blood will contact the hospital. The hospital will generate requests on the app [19]. The app will find online donors first. If there are no online users, then it will no offline donors [20]

3.2. Donor side application

The donor side application will allow registering donors. Put their personal information in their accounts. They will be tracked in real-time if they are proceeding to donate or not so that hospital can arrange alternatives.

4. Methodology

The design and methodology section cover system architecture use case diagram and UML diagram. The system architecture comprises of the main architecture in which we have a user application to which other entities are sending and receiving data. It features a firebase cloud messaging, a firebase database, and ASP.net web services [21,22].

Next, a use case diagram shows how actually the architecture is working between the donors and the hospital. It shows both online and offline case. Please refer to Fig. 1 for further understanding.

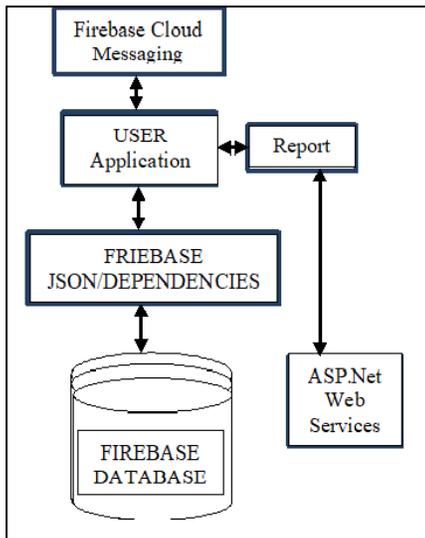


Fig. 1: System Design

The Fig. 2 Use Case Diagram explains the workflow of donors and hospitals. At the donor end, the request is received. The donor can accept or reject the request. On the

hospital end, searching for a donor is being done on two occasions. One is when the donor is online. If the donor is online, the request is generated. If the donor is offline, an SMS broadcast is done.

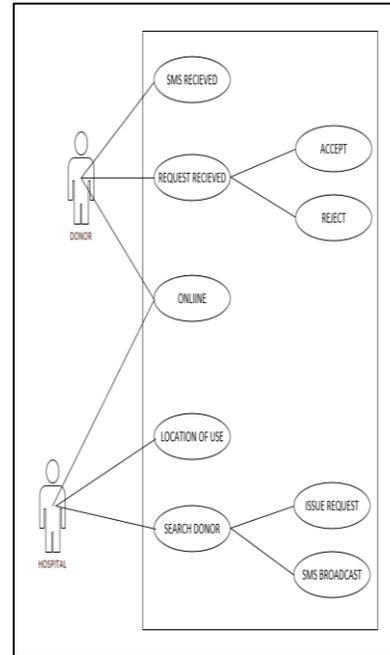


Fig 2: Use Case Diagram

Finally, we have the system UML diagram Fig. 3 which shows the data flow between the classes and entities of the application. Fig. 4 shows donors and their history data flow and Fig. 5 shows the requested functionality that how to request sent from hospital to donor, in this application hospitals register themselves to help patients by providing them donor successfully and save their lives. Many donors donate their blood without any required money they will arrive or reach at hospitals donate blood either match or unmatched group, then hospital blood bank provides blood to patients instead of an unmatched group. By using this application donor are free from any fraud or misuse of blood, the hospital is responsible to manage this type of activity to handle and control.

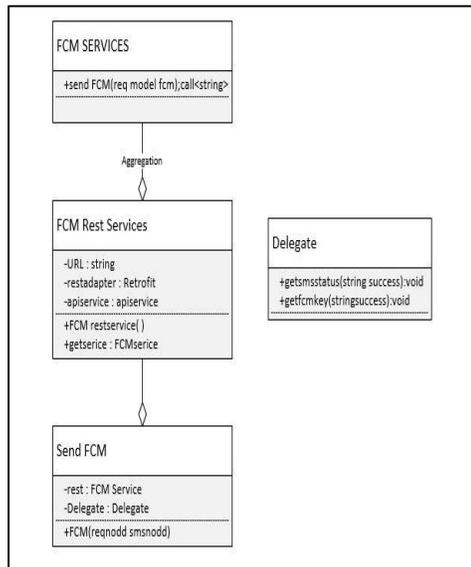


Fig 3: UML Diagram

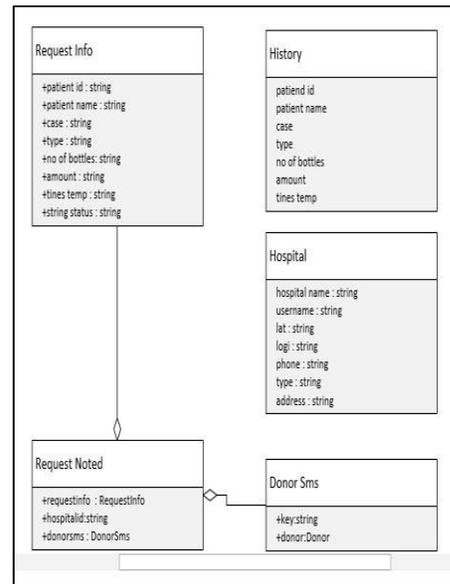


Fig 5: Request Functionality

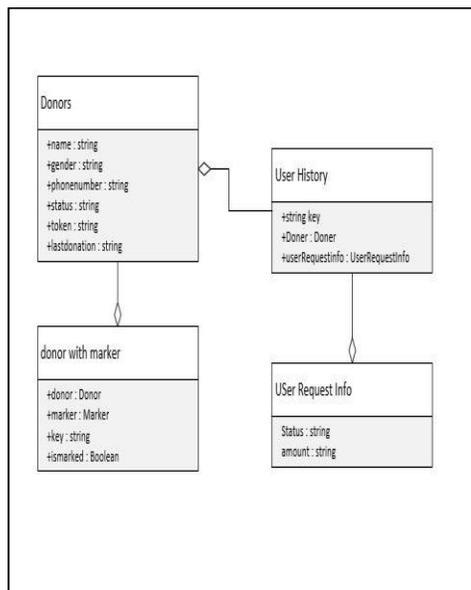


Fig 4: Donors Data

In Fig. 6 represent the graphical user interface of the hospital where hospital management or hospital blood bank control this application and this application could be registered by multiple hospitals because patients are everywhere and the donor could reach over there in case of emergency, in this regard employee of the hospital do connect patients and donor via this application where patients could get benefit from it.

Fig. 7 about the user interface of the patient where the patient himself or his/her relative or friend can register after registration a unique ID will be generated for the user which will be sent to the donor when request sent from patient/user, in this portal patient must link with the concerned hospital where he/she is admitted because in this application donor will reach a destination which is the hospital,

Fig. 8 define the blood group matching mechanism where donor always confusing about matching that his/her donation goes to

the right patient or submitted in blood bank instead of unmatched blood group, sometimes patient relative don't know that which blood group is required in case of O+ and O- blood group, because these types of blood group are very less in humans and very difficult to get it from friends or family, so in critical conditions, it will help in this blood group matching that who can provide either same or cross blood or replaced from the blood bank. When the donor agreed and accept request from the sender then it can be located that how much far from the hospital, in Fig. 9 focus on donors' availability that how many donors are available nearby or far from the hospital. If a donor is not available, then we had done another solution that it could be found through offline request where the user inputs his cell number but forget to ON mobile data or may not available online then send a message through his cell number when he will online then check his status on google map Fig. 10 that where is he either he will be reach at a hospital within time or not or decline him and find another donor.

Fig.11 shows the request portal which is sent by the patient or from hospital blood bank reception to the donor.

In Fig.12, the complete message sent to the donor where the amount shows that you have requested for the amount and complete data of a patient with hospital name is there, at the end we design a Blood Bank user interface portal which is in Fig. 13. If the patient relative or any friend is not available, the hospital is responsible for this patient, in this case the blood bank send request to donors for donate blood and pay for it. As surveyed in many hospitals we get that many blood banks done earlier if patient is in critical condition after successfully patient is recovered then ADD amount in Patient Bills.

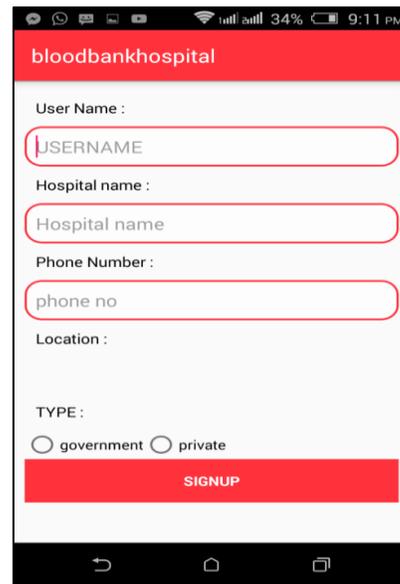


Fig. 6. The graphical user interface of Blood Bank of Hospital

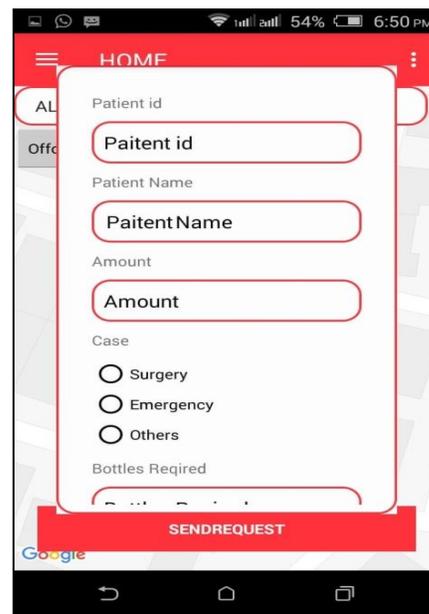


Fig 7. The graphical user interface of Patient

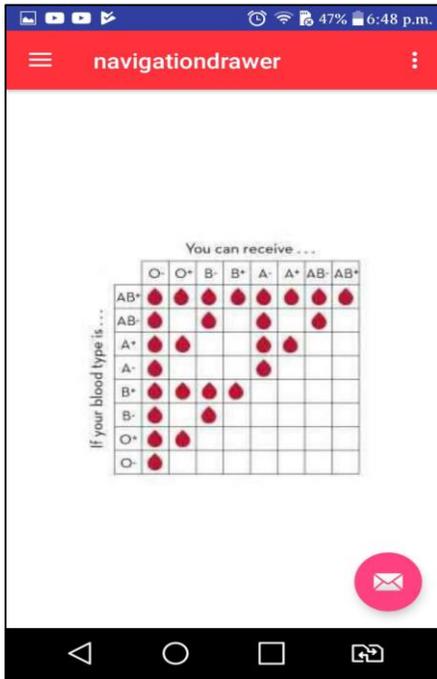


Fig 8. Blood Group Matching

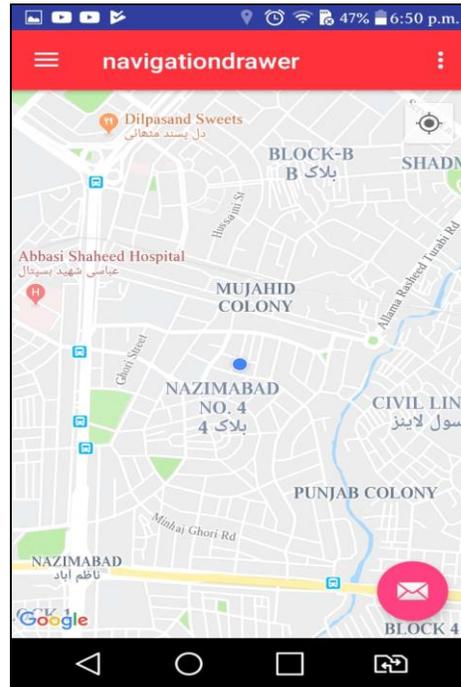


Fig 10. Location of Donor

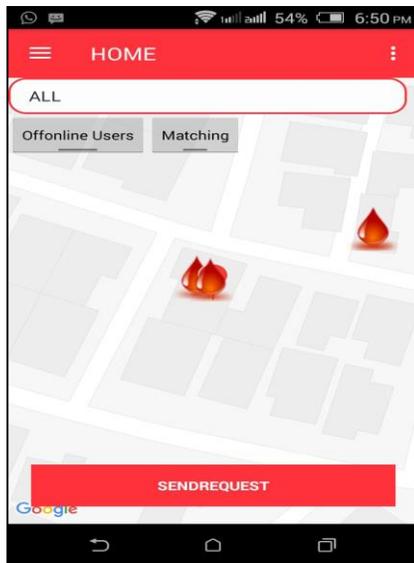


Fig 9. Availability of Donors

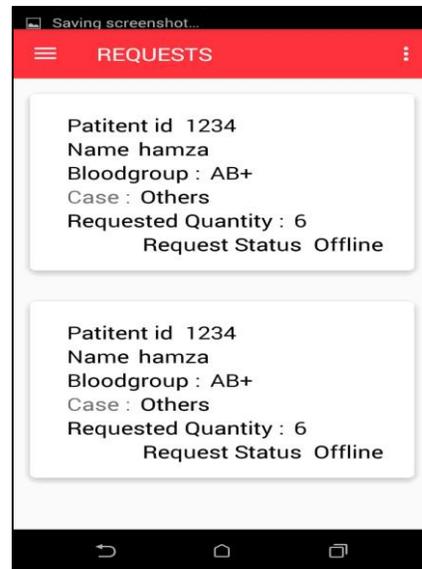


Fig 11. Request Portal

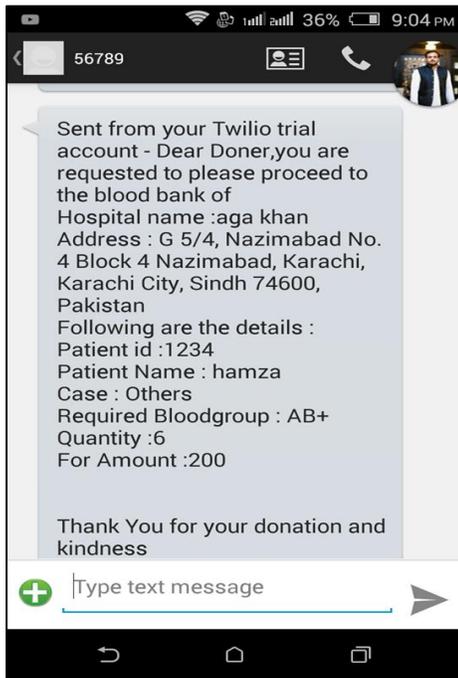


Fig 12. Message sent to the Donor

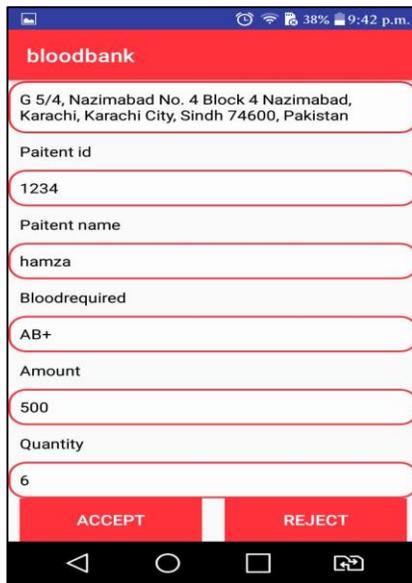


Fig 13. Blood Bank User Interface

5. Implementation

The implementation of the features is covered in this section of the report. The distinctive features include notification SMS broadcasting, firebase cloud messaging, Google maps for hospital and donor, donor history, dismissal of request, request generation by hospital, User account verification by SMS code. The code sections are written here for an understanding of the functionality of the application.

5.1. Notification SMS Broadcasting

```
[HttpPost]
public async Task<string> sendsms(smsmodel sms)
{
    var data = new historydetails();
    string abc = string.Empty
    var sid =
    "AC08036fa1e7d2e8fdf63e2ad27cd35613";
    var token =
    "aa68be533a2b6cd8b39da25ada9057b5";
    TwilioClient.Init(sid, token);
    string address = string.Empty;
    var firebase1 = new
    FirebaseClient("https://bloodbank-
    fd858.firebaseio.com/Hospital/");
    var data1 = await
    firebase1.Child(sms.info.hospitalid).OnceSingle
    Async<hospital>();
    requestinfo req = sms.info;
    var firebase2 = new
    FirebaseClient("https://bloodbank-
    fd858.firebaseio.com/");
    var postinfo = await
    firebase2.Child("history").PostAsync(sms.info);
    string key = postinfo.Key;
    var firebase3 = new
    FirebaseClient("https://bloodbank-
    fd858.firebaseio.com/Hospital/" +
    sms.info.hospitalid + "/history");
    await firebase3.Child(key).PutAsync("true");
    foreach (var i in sms.donerssms)
    {
        abc = sms.info.No_of_bottels;
        var to = new
        PhoneNumber(i.doners.phonenumber);
        var from = new
        PhoneNumber("+16182241163");
        var message = MessageResource.Create(
        to: to,
```

```

from: from,
body: "Dear Doner,you are requested to please
proceed to the blood bank of \n" +
"Hospital name :"+data1.Hospitalname+ "\n" +
"Address : "+data1.Address+"\n" +
"Following are the details :\n" +
"Patient id : " + sms.info.Patentid + "\n" +
"Patient Name : " + sms.info.Patentname + "\n" +
"Case : " + sms.info.Case + "\n" +
"Required Bloodgroup : " + sms.info.Type + "\n"
+
"Quantity : " + sms.info.No_of_bottels + "\n" +
"For Amount : " + sms.info.Amount + "\n\n" +
"Thank You for your donation and kindness"
);
var firebase5 = new
FirebaseClient("https://bloodbank-
fd858.firebaseio.com/history/" + key);
await firebase5.Child(i.key).PutAsync(true);
var firebase4 = new
FirebaseClient("https://bloodbank-
fd858.firebaseio.com/Doners/" + i.key +
"/history");
await firebase4.Child(key).PutAsync(data);
var firebase6 = new
FirebaseClient("https://bloodbank-
fd858.firebaseio.com/history/" + key + "/users/");
await firebase6.Child(i.key).PutAsync(true);
}
return key;
}

```

5.2. Firebase Cloud Messaging

```

[HttpPost]
public async Task<string> FCM(smsmodel
sms)
{
    var data1 = new List<string>();
    foreach (var i in sms.donerssms)
    {
        data1.Add(i.doners.Token);
    }
    var firebase1 = new
FirebaseClient("https://bloodbank-
fd858.firebaseio.com/");
    var postinfo = await
firebase1.Child("history").PostAsync(sms.info);
    string key = postinfo.Key;
    var data = new FCMpayload
    {
        Hospitalid = sms.info.hospitalid,

```

```

Historyid = key
});
using (var sender = new
Sender("AAAA9w1UOgA:APA91bEoW5yV9aj
q8MZVnKiKUNewpiIsPXeulOjEch2BkGQBzb
HNBOS1Vyxyl7e2_ZuFXyJaSX5DCzo007EsUf
pjipKv6OlbU3DA6dcbsxy4L3mzobNZp92ivN60
wtGPE48KaXb33S"))
{
    var message = new Message
    {
        RegistrationIds = data1,
        Data = data,
        Notification = new Notification
        {
            Title = "REQUEST FOR
BLOOD",
            Body = "Blood required" +
sms.info.Type,
            ClickAction =
"com.example.hamzaa.bloodbank.getinfo",
            Sound =
"TYPE_NOTIFICATION"
        }
    };
    var result = await
sender.SendAsync(message);
    string response =
result.MessageResponse.Success.ToString();
    if (response == "1")
    {
        var firebase2 = new
FirebaseClient("https://bloodbank-
fd858.firebaseio.com/Hospital/" +
sms.info.hospitalid + "/history");
        await
firebase2.Child(key).PutAsync(true);
        var firebase3 = new
FirebaseClient("https://bloodbank-
fd858.firebaseio.com/Hospital/" +
sms.info.hospitalid + "/ongoingrequest");
        await
firebase3.Child(key).PutAsync(true);
    }
    return key;
}
}
public async Task<string> FCMnear(string
token,string name)
using (var sender = new
Sender("AAAA9w1UOgA:APA91bEoW5yV9aj

```

```

q8MZVnKiKUNewpilsPXeulOjEch2BkGQBzb
HNBOSIVxyyl7e2_ZuFXyJaSX5DCzo007EsUtl
pjipKv6OlbU3DA6dcbsxy4L3mzobNZp92ivN60
wtGPE48KaXb33S"))
    {
        var message = new Message
        {
            RegistrationIds = new
List<string>{token},
            Notification = new Notification
            {
                Title = "REAHCED",
                Body = "DONER
"+name+"REACHED",
                ClickAction =
"com.example.hamzaa.bloodbank.getinfo",
                Sound =
"TYPE_NOTIFICATION"
            }
        };
        var result = await
sender.SendAsync(message);
        string response =
result.MessageResponse.Success.ToString();
        return response;
    }

package com.example.hamzaa.bloodbank.FCM;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.graphics.BitmapFactory;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Vibrator;
import
android.support.v4.app.NotificationCompat;
import android.util.Log;

import com.example.hamzaa.bloodbank.R;
import
com.example.hamzaa.bloodbank.navigationdraw
er;
import
com.google.firebase.messaging.FirebaseMessagi
ngService;
import
com.google.firebase.messaging.RemoteMessage;

import org.json.JSONException;

import org.json.JSONObject;
/**
 * Created by hamzaa on 11/25/2018.
 */

public class FcmMessageService extends
FirebaseMessagingService {
    String type="";
    @Override
    public void
onMessageReceived(RemoteMessage
remoteMessage) {
        super.onMessageReceived(remoteMessage);
        type=remoteMessage.getData().toString();
        sendNotification(remoteMessage.getNotification(
).getBody(),remoteMessage.getNotification().get
ClickAction(),type);
    }
    private void sendNotification(String s,String
action,String type) {
        String Hospitalid="",Historyid="";
        if(type!=null) {
            try {
                JSONObject jsonObject = new
JSONObject(type);
                Hospitalid =
jsonObject.getString("Hospitalid");
                Historyid =
jsonObject.getString("Historyid");
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        Intent intent = new Intent(action);
        if(Hospitalid!=""&&Historyid!="") {
            intent.putExtra("Hospitalid", Hospitalid);
            intent.putExtra("Historyid", Historyid);
        }
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEA
R_TOP);
        PendingIntent
pendingIntent=PendingIntent.getActivities(this,0,
new
Intent[] {intent},PendingIntent.FLAG_ONE_SH
OT);
        NotificationCompat.Builder builder=new
NotificationCompat.Builder(this);
        builder.setTitle(getString(R.string.app_na
me));
        builder.setText(s);
        Uri sounduri =
RingtoneManager.getDefaultUri(RingtoneManag
er.TYPE_NOTIFICATION);

```

```

builder.setSound(sounduri);

builder.setSmallIcon(R.mipmap.ic_launcher);
//
BitmapFactory.decodeResource(this.getResources(), R.drawable.logo);
builder.setLargeIcon(BitmapFactory.decodeResource(this.getResources(),R.drawable.ic_launcher_foreground));
builder.setAutoCancel(true);
Vibrator
v=(Vibrator)this.getSystemService(Context.VIBRATOR_SERVICE);
v.vibrate(1000);
builder.setContentIntent(pendingIntent);
NotificationManager
notificationManager=(NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

notificationManager.notify(0,builder.build());
}
}

```

6. Results and Discussions

The android application developed is aimed to change the way how blood donation is done using the mobile application. This application is the solution to overcome all the difficulties faced by donors or needy people in regard to a mobile application. This will make it easy for collectors and donors since there is a certain level of trust developed by introducing some new and innovative features.

This application serves a good purpose to both the hospitals and donors. Hospitals can manage donors' history, donors' information, track their location in real-time, and generate requests in the form of notifications and SMS. For donors, this application is a factor of authenticity. We have seen that donors certainly not trust needy individuals on the basis that they are strangers and they cannot be trusted. By introducing a hospital-end module donor can trust where they are going to donate and to whom. Similarly, from the point of needy individuals, it provides a guarantee for them if the donors are approaching the hospital or not with the

feature of real-time tracking. We also have included a finance factor by the hospital end which is a motivational factor for the donor to really approach to the needy individual

References

- [1] Osborne, D.J., Bradley, S., and Lloyd-Griffiths, M., 1978. The anatomy of a volunteer blood donation system. *Transfusion*, 18(4), pp.458-465.
- [2] Wiwanitkit, V., 2000. A study on attitude towards blood donation among people in a rural district, Thailand. *Age (years)*, 11(20), pp.21-30.
- [3] Buciuniene, I., Stonienė, L., Blazevičienė, A., Kazlauskaitė, R., and Skudienė, V., 2006. Blood donors' motivation and attitude to non-remunerated blood donation in Lithuania. *BMC Public Health*, 6(1), p.166.
- [4] Shenga, N., Pal, R., and Sengupta, S., 2008. Behavior disparities towards blood donation in Sikkim, India. *Asian journal of transfusion science*, 2(2), p.56.
- [5] Osborne, D. J., S. Bradley, and M. Lloyd-Griffiths. "The anatomy of a volunteer blood donation system." *Transfusion* 18.4 (1978): 458-465.
- [6] Sahinvezan, Feyza Güliz, Bahar Y. Kara, and Mehmet Rüstü Taner. "Selective vehicle routing for a mobile blood donation system." *European Journal of Operational Research* 245.1 (2015): 22-34.
- [7] Buciuniene, Ilona, et al. "Blood donors' motivation and attitude to non-remunerated blood donation in Lithuania." *BMC Public Health* 6.1 (2006): 166.
- [8] Shenga, Namgay, Ranabir Pal, and Subhabrata Sengupta. "Behavior disparities towards blood donation in Sikkim, India." *Asian journal of transfusion science* 2.2 (2008): 56.
- [9] Gillespie, Theresa W., and Christopher D. Hillver. "Blood donors and factors impacting the blood donation decision." *Transfusion Medicine Reviews* 16.2 (2002): 115-130.
- [10] Mostafa, A.M., Youssef, A.E. and Alshorbagy, G., 2014. A framework for a smart social blood donation system based on mobile cloud computing. *arXiv preprint arXiv:1412.7276*.
- [11] Marantidou, O., Loukopoulou, L., Zervou, E., Martinis, G., Egglezou, A., Fountouli, P., Dimoxenous, P., Parara, M., Gavalaki, M. and Maniatis, A., 2007. Factors that motivate and hinder blood donation in Greece. *Transfusion Medicine*, 17(6), pp.443-450.

- [12] Bas, S., Carello, G., Lanzarone, E., Ocak, Z. and Yalcındağ, S., 2016. Management of blood donation system: literature review and research perspectives. In *Health Care Systems Engineering for Scientists and Practitioners* (pp. 121-132). Springer, Cham.
- [13] Buvx, A.M., 2009. Blood donation, payment, and non-cash incentives: classical questions drawing renewed interest. *Transfusion Medicine and Hemotherapy*, 36(5), pp.329-339.
- [14] Alfouzan, N., 2014. Knowledge, attitudes, and motivations towards blood donation among King Abdulaziz Medical City population. *International journal of family medicine*, 2014.
- [15] Papagiannis, D., Rachiotis, G., Symvoulakis, E.K., Anvfantakis, D., Douvlataniotis, K., Zilidis, C., Markaki, A. and Hadiichristodoulou, C., 2016. Blood donation knowledge and attitudes among undergraduate health science students: A cross-sectional study. *Transfusion and Apheresis Science*, 54(2), pp.303-308.
- [16] World Health Organization, 2012. Blood donor selection: guidelines on assessing donor suitability for blood donation. World Health Organization.
- [17] Ouhbi, S., Fernández-Alemán, J.L., Toval, A., Idri, A. and Pozo, J.R., 2015. Free blood donation mobile applications. *Journal of medical systems*, 39(5), p.52.
- [18] BalaSenthilMurugan, L. and Julian, A., 2015. March. Design and implementation of Automated Blood Bank using embedded systems. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-6). IEEE.
- [19] Adsul, A.C., Bhosale, V.K. and Autee, R.M., 2018. January. Automated blood bank system using Raspberry PI. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)* (pp. 252-255). IEEE.
- [20] Esmail, M.Y. and Osman, Y.S.H., 2018. August. Computerized Central Blood Bank Management System (CCBBMS). In *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (pp. 1-5). IEEE.
- [21] Cheema, A.S., Srivastava, S., Srivastava, P.K. and Murthy, B.K., 2015. December. A standard compliant blood bank management system with enforcing mechanism. In *2015 International Conference on Computing, Communication and Security (ICCCS)* (pp. 1-7). IEEE.
- [22] Kumari, D.M.S. and Wijavanavake, A.N., 2016. October. An efficient inventory model to reduce the wastage of blood in the national blood transfusion service. In *2016 Manufacturing & Industrial Engineering Symposium (MIES)* (pp. 1-4). IEEE