Android Application - RedLife

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

By

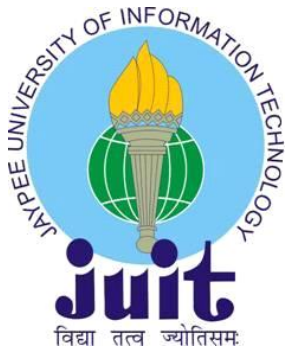
Smily – 131340

Under the supervision of

Dr. Rajni Mohana

Assistant Professor (Senior Grade)

To



Department of Computer Science and Engineering
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"RedLife"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the department of Computer Science**,** Jaypee University of Information Technology Waknaghat is an authentic record of our own work carried out over a period from August 2016 to May 2017 under the supervision of Dr. Rajni Mohana, Assistant Professor (Senior Grade), department of computer science.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Smily (131340)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Rajni Mohana,

Assistant Professor (Senior Grade)

CSE Department

Dated: 09 June, 2017

Certificate

Acknowledgement

Firstly, I would like to express my sincere gratitude to Dr. Rajni Mohana, Assistant professor (Senior Grade), Computer Science department, Jaypee University of Information Technology, Waknaghat, Solan for providing this opportunity and her guidance. Besides my guide, I would like to thank other faculty members of JUIT, Waknaghat, Solan for providing academic inputs, guidance & encouragement throughout the training period to carry out industrial training. It's a great pleasure for me to acknowledge the guidance and contributions of a large number of people to this effort. I would like to thank Mr Sunny Saini (Senior Android Developer), my guide during this internship, who gave me their time and the tools to make my project properly, for spending their time and efforts to develop this project, debug hardware and software and for their guidance about project.

My sincere thanks also goes to Retd. Brigadier Dr. Satyaprakash Ghrera, HOD (Head of Department), Computer Science and Engineering JUIT, Waknaghat, Solan. This report is submitted as a partial requirement for the Bachelor of Technology degree in computer science at JUIT, Waknaghat, Solan. It is substantially the result of my own work except where explicitly shown. Next I would like to thank the staff of Click Labs Pvt Ltd, for their kindness and welcome. Again I thank my project mentors for their constant motivation for the project. I am also grateful to my friends for their valuable feedback.

Table of Contents

List of Figures

List of Tables

Abstract

“BLOOD”, is the most essential necessities of our life. Mostly during accidents the need for blood is the common problem. In emergency, requirements advances in medicine have increased the need for blood in many on-going treatments and elective surgeries. Despite of increase in the requirement for blood, only 5% of the Indian population donates blood. RedLife is a project which aims to develop an online Blood Donor Information System. Through this application any person who is willing to donate the blood can register himself and can receive updates about requirement of blood near them in real time on their device. The person can also make a request for the same along with contact information, type of blood and address. In the same way if any organization wants to register itself with this App, they are welcome. Organizations can inform and spread the word about their events and relevant information about them through this App. The purpose of this App is to keep the track of the registered donors and to maintain such a database that blood could be provided to any user in their need through connecting them with nearest donor of their blood type.

Chapter 1
INTRODUCTION

1.1 INTRODUCTION

The blood is critical for curing patients in restorative field. For every single second somebody needs blood to spare their life either because of mishaps, or because of other serious ailments. The blood donation centers attempts to get blood from different blood givers, to screen and deal with the blood bunches database and to send the required blood amid the need to the healing facility if there should be an occurrence of crises. In creating nations, particularly like India, the blood asset needs in amount which is a boundary to treatment of others life that is for the most part because of number of blood benefactors.

There are numerous inadequacies like unmanaged and diverse nature of giver and required blood is required at genuine circumstances. Physically, it is troublesome in existing framework to track the database for specific blood gathering, which is truly convoluted.

The refreshed data after the gift of the blood by a benefactor about the contributor is entered in the framework. The online blood donation centre administration framework keeps up and deal with the database blood. This builds dependability, adaptation to non-critical failure and accessibility. The online blood donation centre framework with GIS coordinated is imperative since when the lives are at hazard than utilizing the portable application hunting down adjacent blood giver is done where it depends on close-by area. Further, the data is made available to anybody, subsequently solid verification component is required, So that no security ruptures could happen. The fundamental component is utilization of GPS (GLOBAL POSITIONING SYSTEM) in the application. The individuals who enlist utilizing this application, their positions alongside their blood gatherings, address will be shown utilizing this GPS. The fundamental screen of the application will show all the enrolled numbers the individuals who are occupied with giving blood inside the scope of couple of kilometers.

The principle motivation behind this venture is to have one focal and oversaw database of blood giver and assist the healing facility administration with finding the present and precise area of portable of closest contributor by utilizing GPS framework with incredible exactness if there should be an occurrence of crises. Application client can discover the area of any nearest benefactor through RedLife and furthermore make it workable for the intrigued people to enroll themselves as a giver with all points of interest of wellbeing checkup and all other prerequisite expected to enlist so that healing facility administration can utilize that subtle elements in the event of a crisis.

This application will help the destitute patients and their relatives hunting down blood benefactors in their city or range. A man can make a blood ask for by giving their name, contact data, Blood Type required, Units Needed(if more than one), Coordinates got utilizing GPS Technology, and Hospital Name on the demand shape. The submitted demand will reach to the contributors as a warning from where they can contact the individual who made the demand. A Blood Donor can join as a blood giver and get demands for giving blood from Patients, Blood Banks, Health related occasions, and other comparative gatherings and associations. This application will help the needy patients and their family members searching for blood donors in their city or area. A person have the ability to make a blood request by providing their name, contact information, Blood Type required, Units Needed(if more than one), Coordinates fetched using GPS Technology, and Hospital Name on the request form. The submitted request will reach to the donors in the form of a notification from where they can contact the person who made the request. A Blood Donor can sign up as a blood donor and receive requests for donating blood from Patients, Blood Banks, Health related events, and other similar groups & organizations.

1.1.1 Rationale

Blood is essential and necessity for life. It carries nutrients and oxygen throughout the body, fights infections, brings away waste products and helps heal wounds. And everyone has this self-generating resource that can be given to others whenever needed.

- Accidents occur day in and day out that put our companions, family and neighbors in risk.
- Blood gift spare existences of patients amid surgery and constant ailment.
- Premature new born children appeared on the scene frequently requiring blood to survive and giving support.

If not by and by required, sooner or later we as a whole know a companion, neighbor, relative, or colleague who needs a blood transfusion. Truth be told, 1 out of few individuals will require blood in their lifetime. From cataclysmic events to inconspicuous disasters, crisis healing facility techniques to deep rooted fights with ceaseless sicknesses, for example, malignancy the interest for blood is steady and assessed to increment in future.

The asset is short. What's more, there is no substitute to blood, just volunteer blood contributors can approach and spare lives.

RedLife is that stage which gives the poor individual a stage to organize and impart his prerequisite for the blood to the givers along these lines keeping away from the tedious procedure and making the errand of getting blood substantially more attainable and simpler.

1.2 PROBLEM STATEMENT

There is a need of a service using which a needy person can get the blood at the earliest saving a priceless human resource. Our service focuses not only on precision of finding a suitable donor, but also on speed i.e. the requirement should be fulfilled at the earliest. This Android Application is being developed to get the blood easily in case of emergency in nearby places. A person can get access to the required blood in real time and real place using RedLife.

1.2.1 Existing System

Existing system of looking up for blood is a loop of hit and trial which repeats in case of error or failure.
The Needy tries to reach as much people he/she could to find the required type of blood.
Usually the communication channel means phone calls by them or their relatives to their relatives. Some people use social networking sites (Facebook, Twitter, etc.) or IM (WhatsApp, Hike, etc.) services to spread the requirement but then again such messages cannot be easily verified and hence ignored by the recipient.
When a person is eager to donate blood, he tries to reach nearest Blood Banks or Blood
Donation Camps. However, not everyone is informed about the camps and sometimes even banks which is why Donor cannot donate even willingly. The existing system is flawed to an extent and can be improved by implementing major changes which will be discussed in this document in later sections.

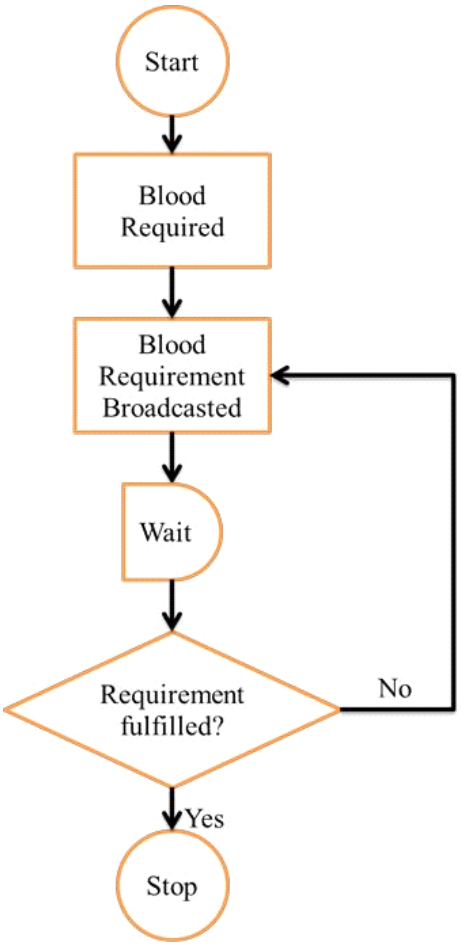


Figure 1.1: Process Flow of existing system of finding blood donor.

1.2.2 Limitations of existing system

Existing System regarding the blood donation scenario have many limitations. Some of them are as follows:

- It takes so much time for the needy find the details of a donor.
- Poor communication between needy/organizations and donors, so here informing donors about new requirement is a hard task.
- Needy people may not get connected to the donors instantly.

1.3 OBJECTIVE

The objective of this application is to provide a required type of blood in emergencies by searching for the donor in nearby area to help patient getting blood transfusion in required period of time.

1.4 METHODOLOGY

I have used Android Studio (2.3) for implementing android application.
Requirement –
Scribble down the necessity that application requires, at that point examine the device for creating application and afterward introduce Android Studio 2.3.

UI outline for application-

UI for portable application can make easy to use application. Through utilize case, arrangement chart and information stream outline and taking a shot at it with the goal that best outcome can be given by client. This application is a multistep procedure with its survey stages. The final product is blue prints that educate and shows last item and how the cooperation ought to felt, moved and stream. This is to be finished by utilizing XML in android studio in planning formats by making movement and parts.
Coding for application-
Develop application by implementing various frontend and backend processes.

Testing the application-
Test the implementation of application so that it is easy to use and learn, this is done by:

- Unit testing
- Performance testing and
- Integration testing

1.5 ORGANIZATION

This thesis is organized as follows.

- The introduction is given in Chapter 1. This section describes the problem statement, objective, methodology and organization.
- Chapter 2 reviews various existing technologies as well as new technology and also the tools that are related to the work presented in this thesis under the title literature survey.
- Chapter 3 System Development specifies all the technical aspects of the project, the concepts used and the theory revolving around it.
- Chapter 4 describes the performance analysis of the complete implementation of the project.
- Chapter 5 is conclusion section. It gives the conclusion of the project and also future enhancements in the project.

And the last section Appendix and References. I have mentioned the annotated bibliography. This section is a list of the journals and research papers used for the preparation of this project report.

CHAPTER 2
LITERATURE SURVEY

2.1 BACKGROUND

There are short comings in web application in the blood donation system:

- Obstacles in handling emergency conditions.
- No security for personal information may lead to misuse of personal information.
- Security breaches by third parties.
- No updates regarding recent information.
- Needs intermediate to work manually on information update.
- Leads to error in results.

2.2 RELATED WORK

Traditionally, the blood demander are depending on hospitals and blood banks to fulfil their needs. This makes the system work slow and less efficient. Recent efforts are being made to automate this process. Some of the works in this category are given below:

2.2.1 Blood Banking by Indian Red Society

This Blood Banking application records all your information, tracks your account, and provides timely reminders when your next donation is due. Most importantly, you can make blood transfers to the needy at a click of a button.

2.2.1.1 Advantages of Indian red Society

- Blood can be deposited in advance.
- Appointment can be scheduled for donation of blood in the nearby Blood Bank.
- Blood Units can be shared to other account holders.

2.2.1.2 Disadvantages of Indian Red Society

- Not favourable to use since obtaining process is long and tedious.
- Red Cross works on the principal of give and take, i.e. the units of blood taken must be returned by the family members of the patient.
- The app is usable only for the people living in Bangalore, India.

2.2.2 Blood Donor

Blood banks and plasma centre helps you find a nearby place if you get injured and need blood. However, it doesn‘t directly interacts with the donor, it only provides to donors about where they can donate blood.

2.2.2.1 Advantages of Blood Donor

- Blood Donor app is an android app that motivates the more and more persons to donate the blood.
- They give information about the emergency blood bank nearby in the case of accidents.
- This application has both a website and android application.

2.2.2.2 Disadvantages of Blood Donor

- Inefficient search algorithm and requires highly connective network.
- Server is ill equipped to handle large traffic.
- Sloppily maintained database gets updates and modified at a slow pace.
- The contact information of donors is not safely stored and can be easily stolen.

2.2.3 Indian Blood Donors

Indian Blood Donors is an application that provides us with donor‘s name, contact number and other information narrowing the donor‘s down using STD Code or the Postal Code of the area.

2.2.3.1 Advantages

- It locks down the donor for 90 days once his/her name is popped open in the app.
- It is easier and faster to look for donors on Indian Blood Donors Application.

2.2.3.2 Disadvantages

- A query made unknowingly or mistakenly or for testing purpose will lead to waste a precious potential donor depending upon the number of queries made by the user.
- The contact information of donors is not safely stored and can be easily stolen.

2.3 PROPOSED SOLUTION

The blood is a specialized fluid that contains necessary substances like Oxygen and Nutrients to the body.

Absence or shortage of blood in human body can lead to serious health issues which may directly or indirectly lead to death of person. Life is always full of uncertainties.

A person living right now may cease to exist five minutes later. These days, road accidents prove to be the fatal accidents where usually a person dies for the reason that excess loss of blood.

There is a need of a service using which a needy person can get the blood at the earliest saving a priceless human resource. Our service focuses not only on precision of finding a suitable donor, but also on speed i.e. the requirement should be fulfilled at the earliest. This Android Application is being developed to get the blood easily in case of emergency in nearby places. In this Android Application, one will get access to required blood in real time and real place.

2.4 ANDROID

It is a mobile operating system which is created by Google on the Linux kernel for the touch screen device such as mobile and tablet. It consist of manipulation on screen objects, with additional feature of virtual keyboard for entering the text .It has functionality of real world actions ,touch gestures such as pinching , tapping, swiping.
Before portraying the Android application essentials, the Android OS design will be clarified. There are diverse parts in the Android OS.

- Applications: There are distinctive centre applications in view of Java, for example, email customer, maps, programs, and so forth
- Application Framework: Developers could take advantage of their access to the framework APIs used by the core applications. It makes the components reuse easy.
- Libraries: A number of different libraries based on C/C++ will be used by many of the system's components.
- Android Runtime: There are various centre libraries which give the functionalities of the centre libraries of the Java. The Dalvik VM gives the gadget the capacity of running numerous virtual machines. It depends on Linux Kernel for a portion of the functionalities.
- Linux Kernel: It is an abstraction layer between the hardware and whatever is left of software stack. A few functionalities of the bit are threading, low-level memory overseeing, prepare overseeing, arrange stack, and so forth. Android working framework is a pile of programming segments which is generally isolated into five segments and four principle layers as appeared in the design chart (Figure 2.1).

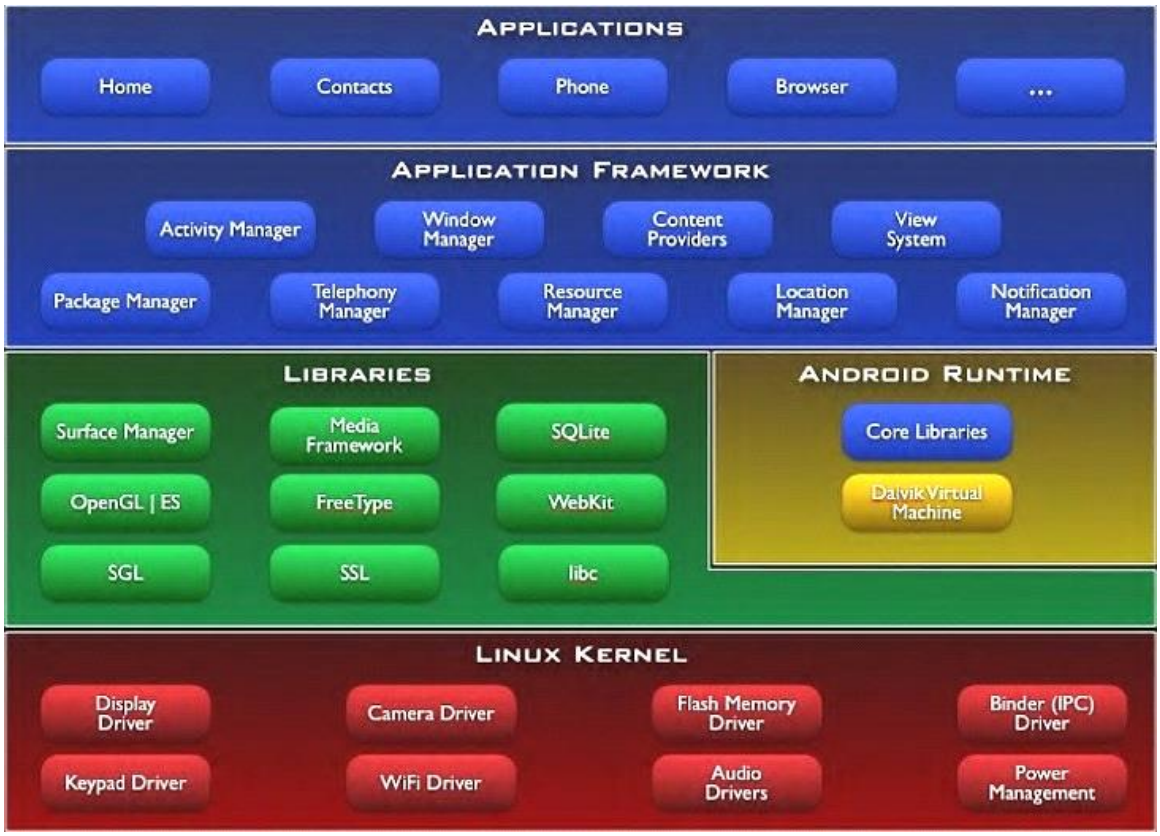


Figure 2.1: Android Architecture

In the demands of having some data about the system's architecture, the application components could be presented. They are building blocks which are necessary in the application and are diverse passage points for system to enter the application. The following are distinctive types of components and clarification of their correct part in the application's general conduct.

- Activities: Different activities could all work together to improve an application's usefulness, yet they are as yet autonomous from each other.

An application could begin some of other application's exercises on the off chance that it has enough benefit to do as such. Every action is actualized as a subclass of Activity class and accompanies a UI which is fundamentally a solitary screen.

The going with diagram exhibits the basic state methods for an Activity. The square rectangles address callback strategies you can execute to perform operations when the Activity moves between states. The shaded ovals are noteworthy states the Activity can be in.

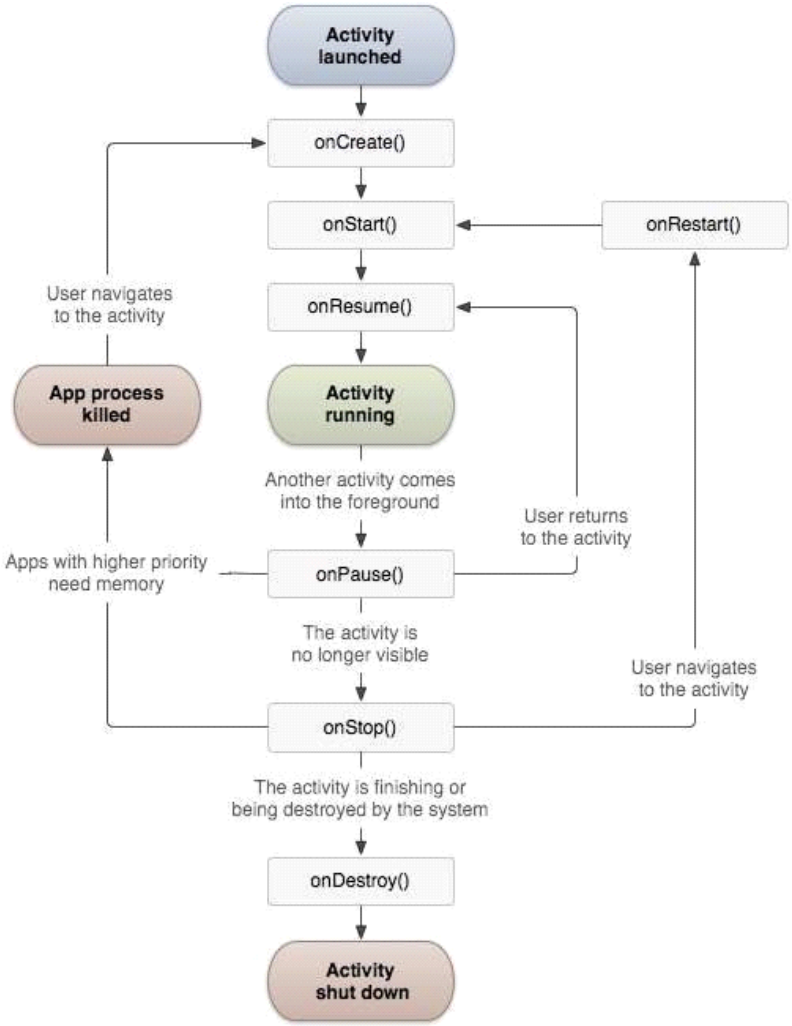


Figure 2.2: Describes Activity Life cycle of android activity

- Services: Conversely with exercises, a service does not accompany a UI. Instead is going on that a segment like an action could begin a service and let it keep running out of sight, collaborate with it by restricting itself to it or make it perform for remote procedures. Each service is actualized as a subclass Of service class and could be either made with startService() or bindService(). In figure 2.3.

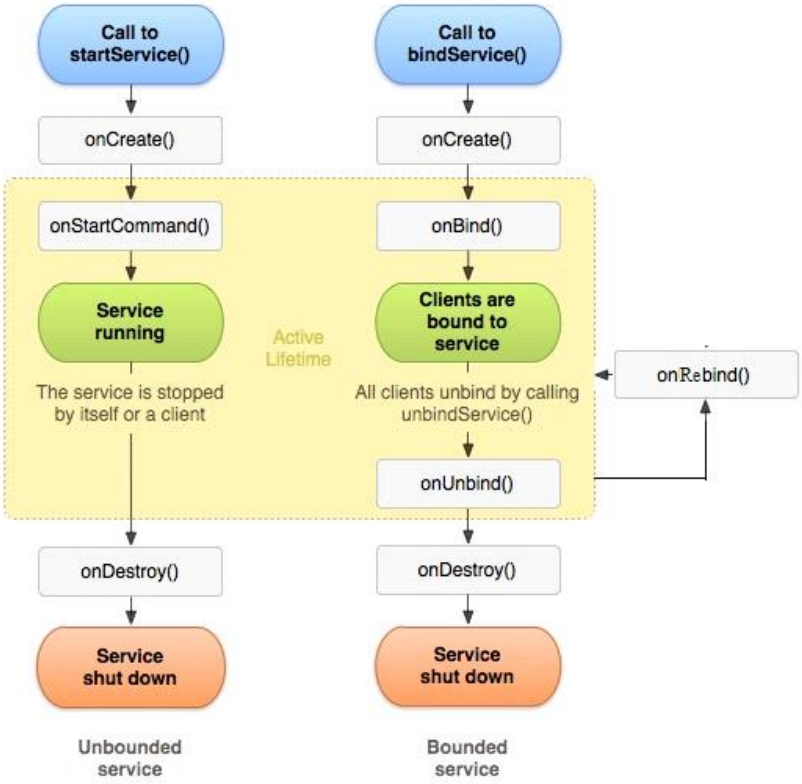


Figure 2.3: Tells about the service life cycle of activity.

- Broadcast receivers: It is a portable working framework which is made by Google on the Linux piece for the touch screen gadget, for example, versatile and tablet. It comprise of control on screen objects, with extra element of virtual console for entering the content .It has usefulness of true activities ,touch signals, for example, squeezing , tapping, swiping. Before depicting the Android application fundamentals, the Android OS configuration will be elucidated. There are differing parts in the Android OS.
- Applications: There are particular focus applications in perspective of Java, for instance, email client, maps, projects, et cetera. [16].

- It makes it less demanding to refresh entirely unexpected attributes of the application while not redoing the source code. XML documents that are intended for different designs, may be just adjusted for protest.

- Alternative assets could be utilized for various gadget setups. A few dialects could without much of a stretch be taken care of by referencing diverse assets for example.

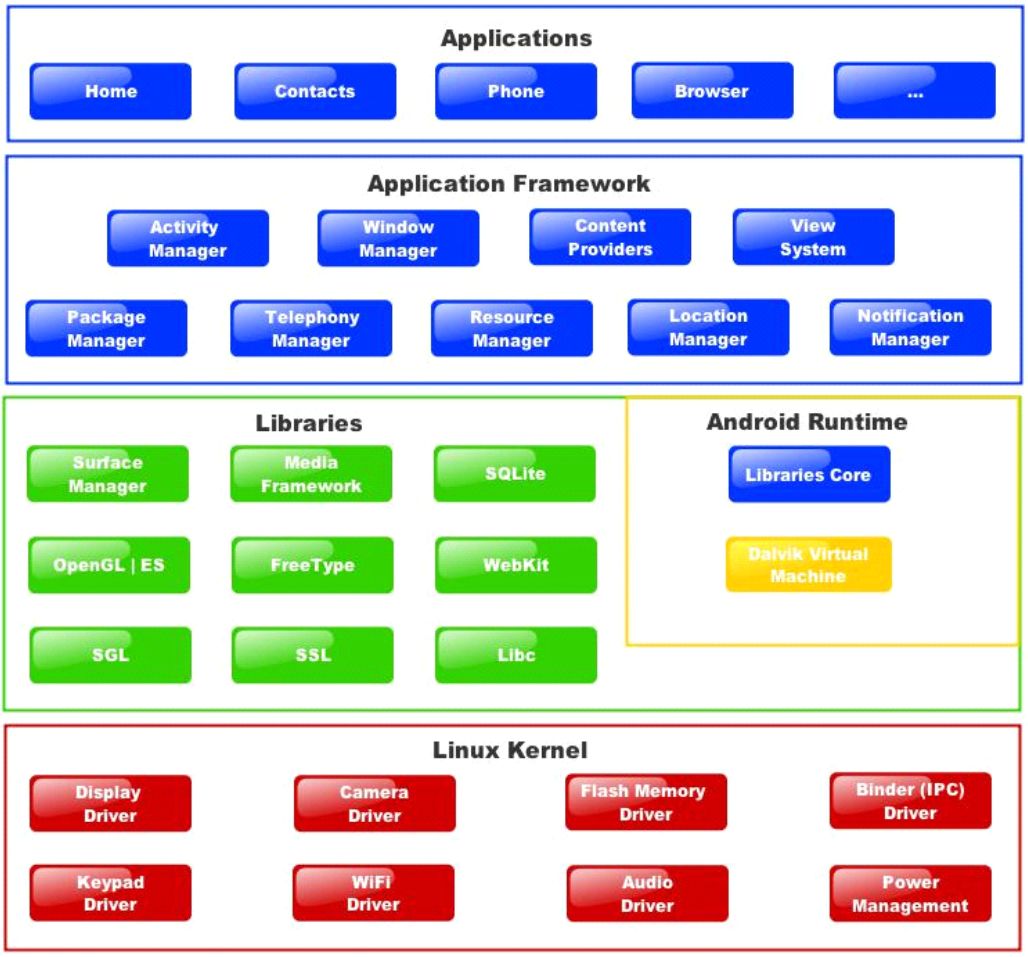


Figure 2.4: Android Architecture (Detailed View)

2.4.1 Linux Kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

2.4.2 Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, includes C/C++ libraries. SSL libraries responsible for Internet security etc.

2.4.3 Android Libraries

This section encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. It facilitates easy interface through java. A summary of some key core Android libraries available to the Android developer is as follows –

- android.app − It facilitates easy access to the application model and is the cornerstone of all

Android applications, which unites all other application.

- android.os − It Provides applications access to standard operating system services including messages, system services and inter-process communication.

- android.text − Used to render, manage and manipulate text on a device display.

2.4.4 Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine (Dalvik VM) which acts as a translator between application side and the operating system, and is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine and facilitates communication.

The Android runtime also provides a set of core libraries which enable Android developers to write Android applications using standard Java programming language.

2.4.5 Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes, this layer has been designed to provide ease and reuse of components in android. Application developers are allowed to make use of these services in their applications.

Activity Manager − Controls and manage all aspects of the application lifecycle and activity stack.

- **Content Providers** − Allows applications to publish and share data with other applications which is required to be maintained in an application.

- **Resource Manager** − Provides access to non-code embedded resources such as strings, colour settings, dimensions of text and user interface layouts.

- **Notifications Manager** − Allows applications to display alerts and notifications to the user through messages and sound alert.

- **View System** – An extensible set of views containing various other classes, which is used to create application user interfaces.

2.4.6 Applications

We will find all the Android application at the top layer. We write our application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Facebook, WhatsApp etc.

CHAPTER 3
SYSTEM DEVELOPEMENT

I have used Android platform for this system. This system heavily relies and need as many users as we can get to remove the gap between the patients and blood donors. Android constitutes more than 82.8% of smartphone market in Q2 of 2015.
This project uses LAMP stack which acts as a dynamic server for various events occurring in the app.

3.1 LAMP STACK

LAMP stack is a popular open source web platform commonly used to run dynamic web sites and servers. It includes Linux, Apache, MySQL, and PHP/Python/Perl and is considered the platform of choice for development, design and deployment of high performance web and mobile applications which require a solid and reliable foundation.

Figure 3.1: LAMP Stack

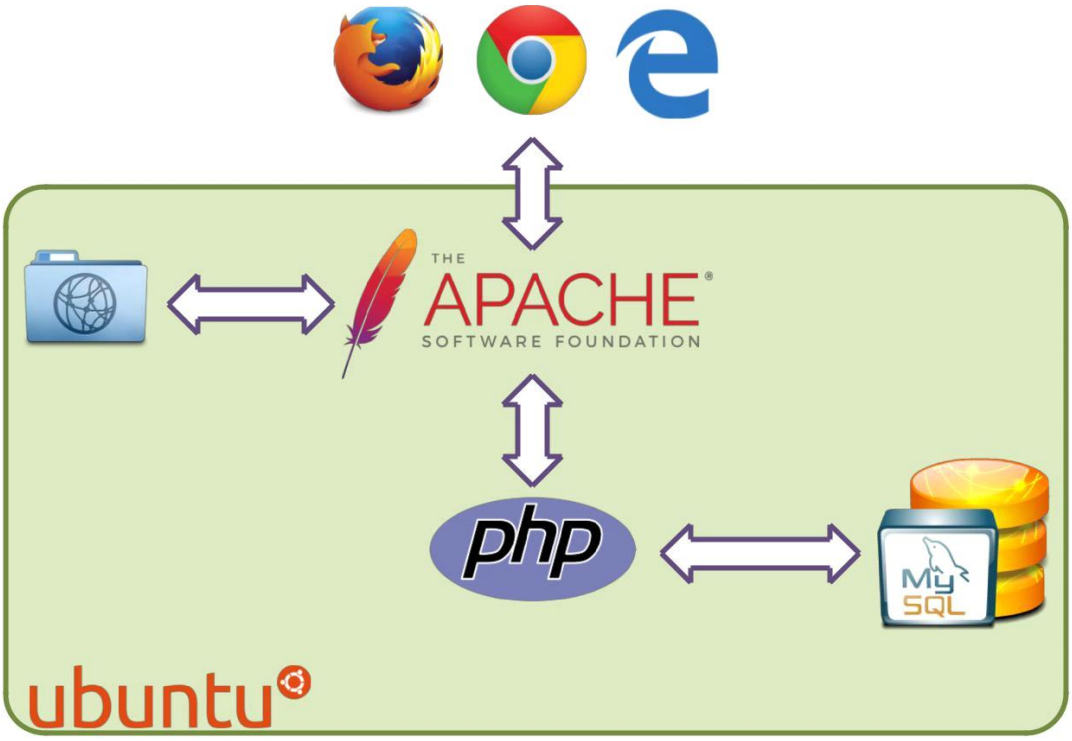


Figure 3.2: Lamp stack (Flow Diagram)

3.1.1 Linux

Linux is the operating system on which the applications execute and run. As an operating system,
Linux is obviously acts as much more than a Web application platform, but it has many benefits too. Aside from the FOSS approach of Linux, the flexibility of the Linux operating system show its importance and how it can be customized means it is fairly easy to produce a highly optimized platform on which to deploy mobile as well as web applications.

The open source approach of Linux has meant that the operating system has been developed for a wide range of platforms, from desktop PCs to massive servers and mainframe hardware, all the way down to embedded systems. Thus, "old" or unusual hardware that is no longer OEM-supported can now be used to run Linux and to deploy LAMP applications.

3.1.2 Apache

The Apache project is a generic name given to a number of high-quality, open-source-based application projects, but its original project was the HTTPD Web server. Apache HTTPD is a powerful Web serving platform that can be used to provide basic file-based serving, which can work with CGI to provide interactive applications or be combined with loadable modules to support a very efficient Web application service.

Within the Apache HTTPD project, this project includes modules for caching data, providing different authentication, authorization and security environments, and supporting a wide range of management functionality. For Web application support, modules enhance the power of the Apache HTTPD by allowing users to embed an interpreter for one of the LAMP languages (Perl, Python, or PHP) into the Apache HTTPD, significantly increasing the speed with which these applications are executed and therefore improving the responsiveness of the mobile as well as web application.

Apache can be easily installed and configured and then largely forgotten. It is safe, secure, easy and self-managing, and it needs little attention to keep it running and following. In general, once the initial configuration has been completed, an Apache installation need not to be be touched ever again.

3.1.3 MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses.

MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

3.2 SERVER SIDE REQUIREMENTS

3.2.1 Hardware Requirements

Hardware requirements are as follows:

- **RAM:** 4GB or above
- **Hard Disk:** 8 GB
- **Processor:** 22GHz processor and above
- **Operating System:** windows 7 or above
- **Network:** 4 MegaBit/sec or higher

3.2.2 Software Requirements

Following software are required for the application to work perfectly since the back-end of our application heavily relies on this part

- Android studio
- JDK (java development kit) 1.7
- **Web Server:** Apache httpd Server
- **Database:** MySQL
- **Scripting Language:** PHP
- **phpMyAdmin**

3.3 CLIENT SIDE REQUIREMENTS

Client is required to have an android smartphone. A detailed requirement draft for client side device is as follows:

- **API level:** 19

- **Android Version:** 4.4 (KitKat) or above
- **Device capabilities:** The features that our app uses:
    - Network Connectivity
    - GPS
    - File Read/Write Access
    - Access Contacts
- **RAM:** 1 GB, 40 MB Free
- **CPU Speed:** 1 GHz
- **Disk:** 20 MB, 10 MB minimum
- **Display Size:** Any size.

3.4 SYSTEM ANALYSIS

First in the system development process is preliminary Investigation. Preliminary Investigation is conducted in the following phases:

- Project Clarification
- Feasibility Study
- Project Appraisal

Project clarification is the process of selecting a project request for further study. When a system development or modification request is made, the first systems activity, the preliminary investigation, begins the activity has three parts: Request clarification, feasibility study and project appraisal. Many request from employees and users in organization are not clearly stated.

Therefore, before any systems investigation can be considered, the project request must be examined to determine preciously what the originator wants. This is called Request clarification.

As important outcome of the preliminary investigation is the determination that the system request in feasible.

3.5 SYSTEM DESIGN

System design is the process of planning a new system or to replace the existing system. Simply, system design is like the blueprint for building, it specifies all the features that are to be in the finished product. System design phase follows system analysis phase. Design is concerned with identifying functions, data streams among those functions, maintaining a record of the design decisions and providing a blueprint the implementation phase.

Design is the bridge between system analysis and system implementation. Some of the essential fundamental concepts involved in the design of application software are:

- Abstraction: Abstraction is used to construct solutions to problem without having to take account of the intricate details of the various component sub problems. Abstraction allows system designer to make step-wise refinement, which at each stage of the design may hide, unnecessary details associated with representation or implementation from the surrounding environment.
- Modularity: Modularity is concerned with decomposing of main module into well-defined manageable units with well-defined interfaces among the units. This enhance design clarity, which in turn eases implementation, Debugging, Testing, Documenting and Maintenance of the software product. Modularity viewed in this sense is a vital tool in the construction of large software projects.

Some of the important factors of quality that are to be considered in the design of application software are:

- Reliability:
  The product ought to carry on entirely as per the first particular and ought to work easily under ordinary conditions.
- Extensibility:
  The product ought to be fit for adjusting effectively to changes in the particular.

- Reusability:

The product ought to be created utilizing a measured approach, which licenses modules to be reused by other application, if conceivable. The System Design quickly portrays the idea of framework outline and it contains four areas. The primary segment quickly depicts the elements that the framework will give to the client and the yields that the proposed framework will offer, the second section namely Logical Design describes the Data Flow Diagrams, which show clearly the data movements, the processes and the data sources, and sinks, E-R diagrams which represent the overall logical design of the database, and high-level process structure of the system.

3.6 ARCHITECTURE

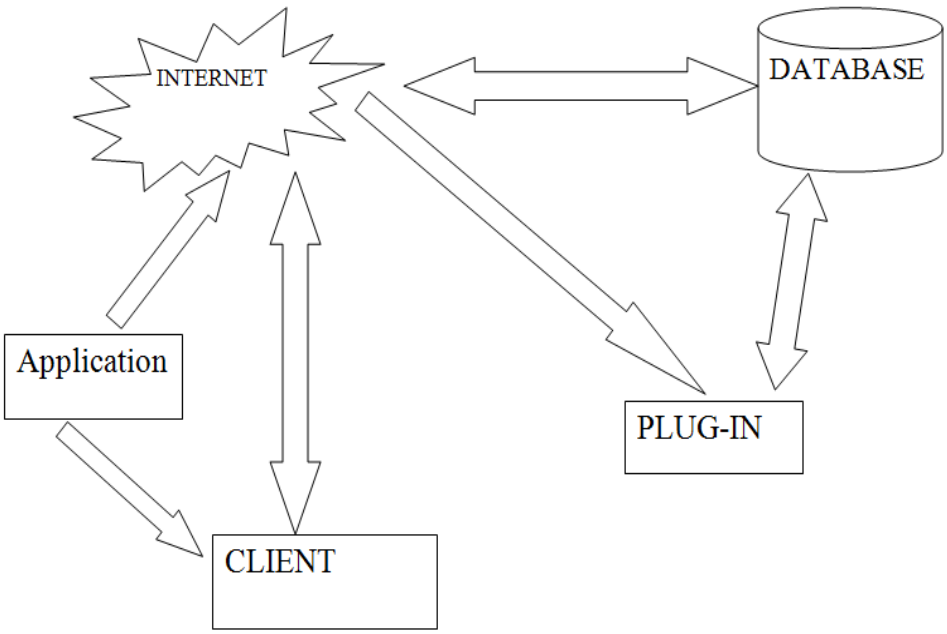


Figure 3.3 Architecture

This application depends on customer server design, in which customer as a client utilize the application and tap on the catch for continue further, however when client tap on catch at that point ask for experience database. On the off chance that there isn't right information embed by client then application is not open further and if information is right then customer/client continue further.

RedLife Blood Donor Lookup Service is a system that manages the records of willing donors and organizations working in the relevant field. This application will help the needy patients and their family members searching for blood donors in their city or area. A person have the ability to make a blood request by providing their name, contact information, Blood Type required, Units needed (if more than one).

While making a request, coordinates will be fetched using GPS in the background while the user fills request form with information like Full Name of person of contact, name of patient, contact number, blood group required, units needed, period within which request should be fulfilled, and Hospital Name & Address. The submitted request will reach to the donors in the form of a notification from where they can contact the person who made the request. A Blood Donor can sign up as a blood donor and receive requests for donating blood from Patients, Blood Banks, Health related events, and other similar groups & organizations.

There is an ability to carry out following operations in this system:

- Making a request for blood

- Accepting/Denying request by user
- Notifications to all subscribed users
- Donation History
- User Profile
- Organizations and their events, announcements, etc.

There are three types of users in this project:

- Donor
- Organization
- Administrator

There is a main role of the admin in this system. Admin have complete access to the whole system, while a donor is only required to have access to the requests and/or make requests for the blood. Organizations, in addition to the functions given to the Donors, also have ability to create events and make announcements.

The Admin role can be as follows:

- Create Organizations
- Delete fake/completed/invalid requests
- Back up of Database
- Delete Organizations
- Delete Events or Announcements.
- Check spamming in the system.

The Donor role can be as follows:

- Edit profile information
- Receive Push Notifications of requests
- Create requests
- Delete own requests
- Accept/Deny requests by other users

The Organization role can be as follows:

- Edit profile information
- Receive Push Notifications of requests
- Create Requests
- Delete own requests
- Accept/Deny requests by Donors
- Create an Event
- Make an announcement

3.6.1 Design

Before start developing the project, one must first analyse the needs and requirements. We make a rundown of essential needs that we need our application to fulfill. Additionally, list down the optional needs. The Internet is the greatest wellspring of online exchange and trade that has altered business exercises over the globe. It helps in simple exchanges which empower guide access to both specialist co-ops and customers for compelling trade of products and enterprises.

3.6.1.1 Use Case Diagram

A Use Case Diagram is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

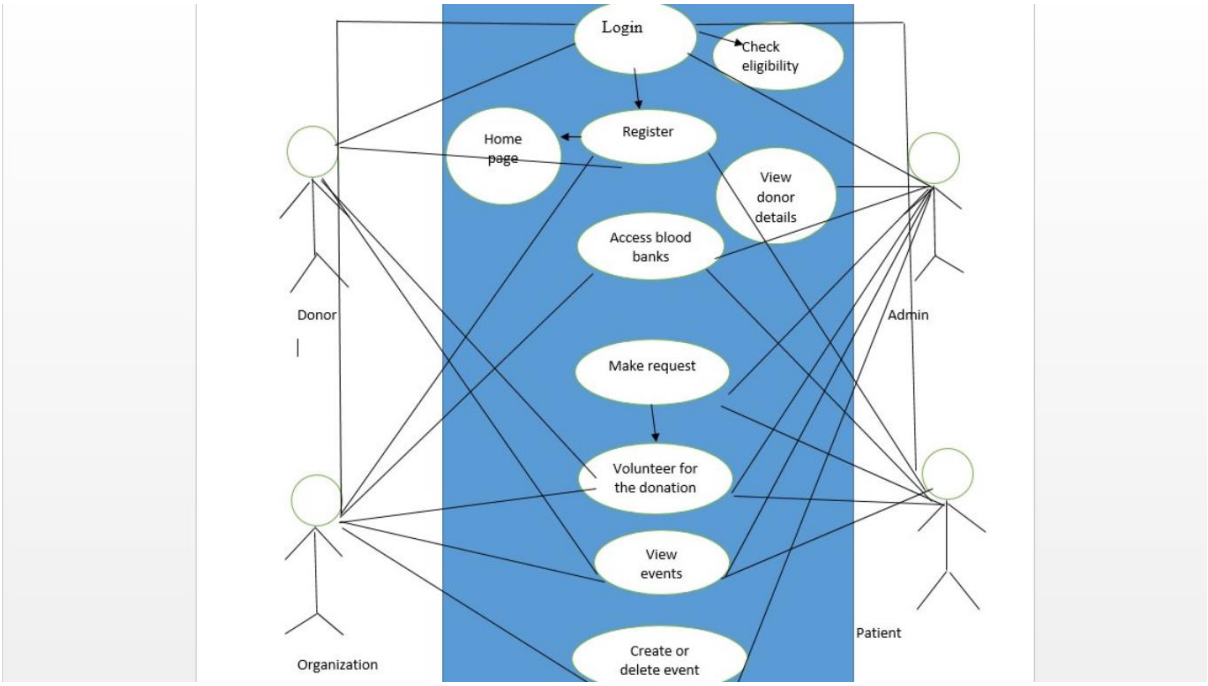


Figure 3.4: Use Case Diagram for RedLife

3.6.1.2 Use Case Specifications

- Actors
  1. Donor
  2. Organization
  3. Receiver
  4. Admin

- Use Cases

1. Donor
   a. Access Welcome Page
   b. View Application Info
   c. View Requests
   d. Search User/Organization
   e. View Events

f. Accept/Deny Requests

2. Organization
   a. View Requests
   b. Make Requests
   c. Search Donor/Organization
   d. View Events
   e. Create/Delete Events
3. Receiver
   a. Access Welcome Page
   b. View Requests
   c. Make Requests
   d. Search Donor/Organization
   e. View Events
   f. Create/Delete Events
4. Admin
   a. Access Welcome Page
   b. Edit/Delete Requests
   c. View Requests
   d. Make Requests
   e. Mark Requests as Completed
   f. Search Donor/Organization
   g. View Events
   h. Create/Delete Events

3.6.1.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

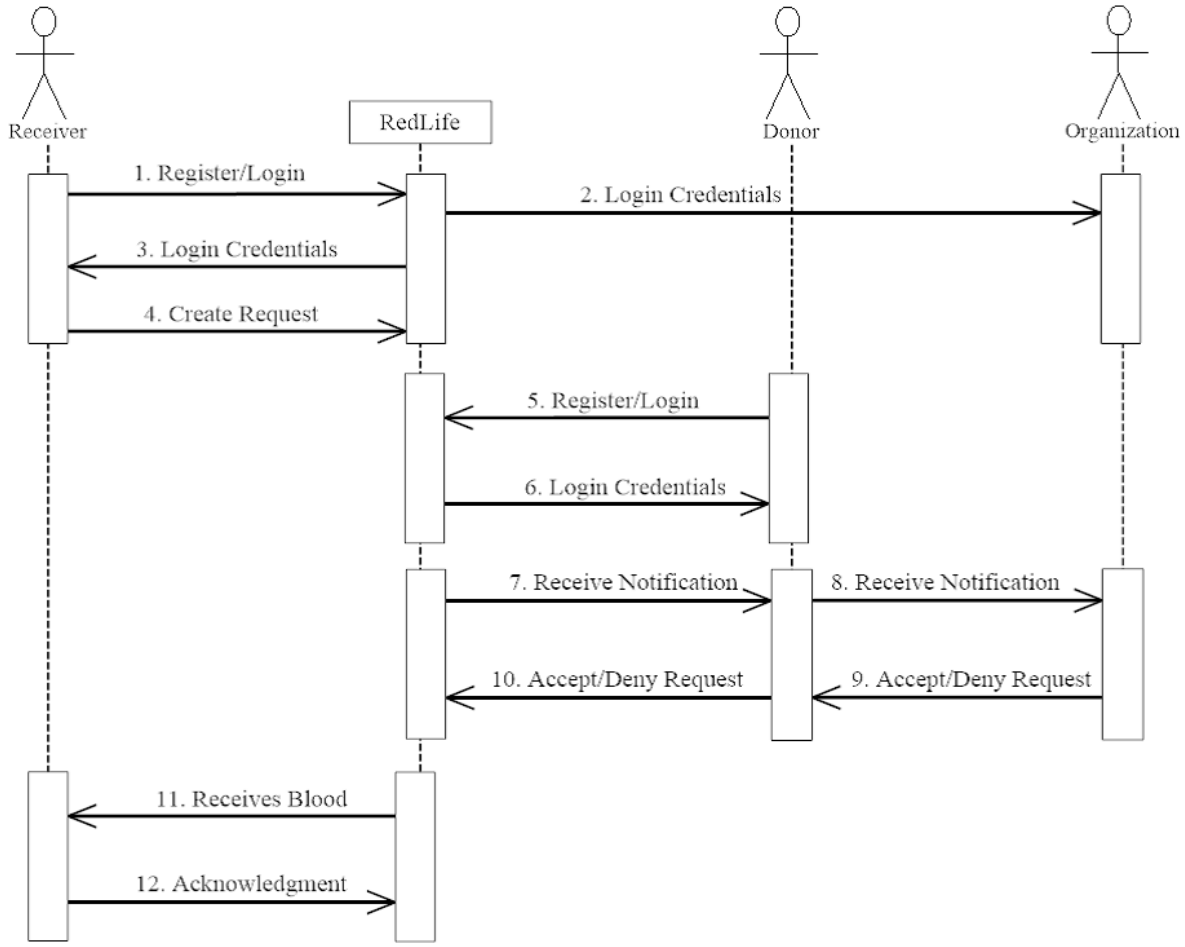


Figure 3.5: Sequence Diagram for RedLife.

3.6.1.4 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

- Circle - It represents the process.
- Squares - It represents the source and destination.
- Arrows - It represents the data flow.

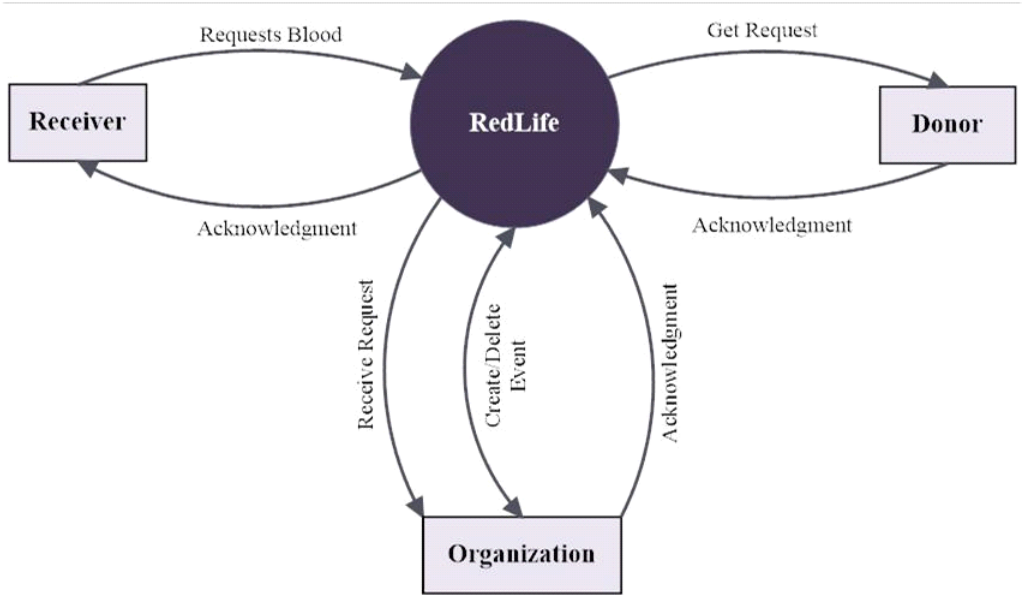


Figure 3.6: Data Flow Diagram Level 0 for RedLife

CHAPTER 4
IMPLEMENTATION AND PERFORMANCE ANALYSIS

4.1 APPLICATION IMPLEMENTATION

Below are shown screenshots of screen mentioned in RedLife Android Applications.

4.1.1 Splash Screen screenshot

Figure 4.1 Splash Screen

4.1.2 Login Screenshot

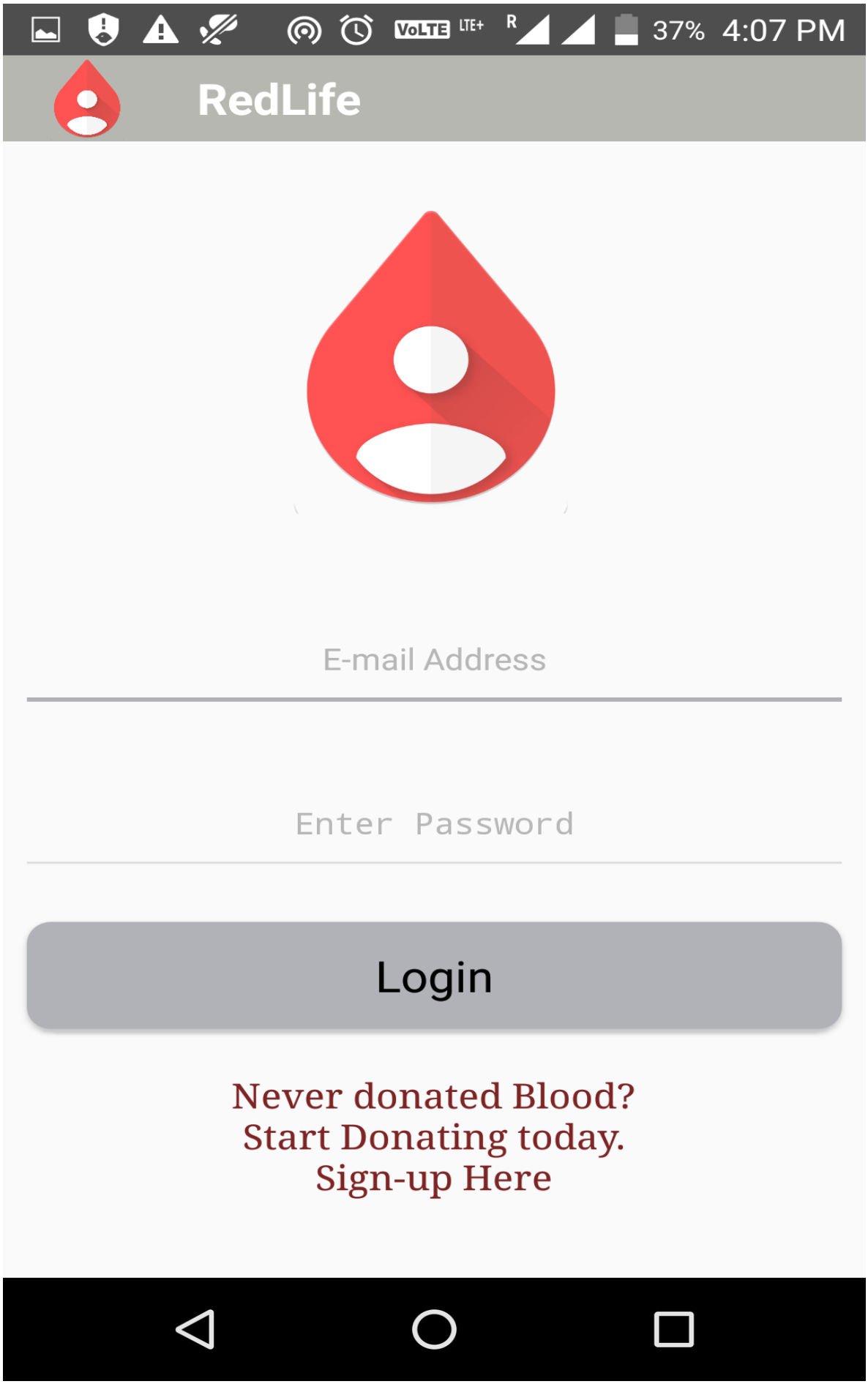


Figure 4.2 Login Screen

4.1.3 Registration Screen

Figure 4.3 Registration Screen

4.1.4 Home Page (With 1st Image)

Figure 4.4 Home Screen 1

4.1.5 Home Page (With 2nd Image)

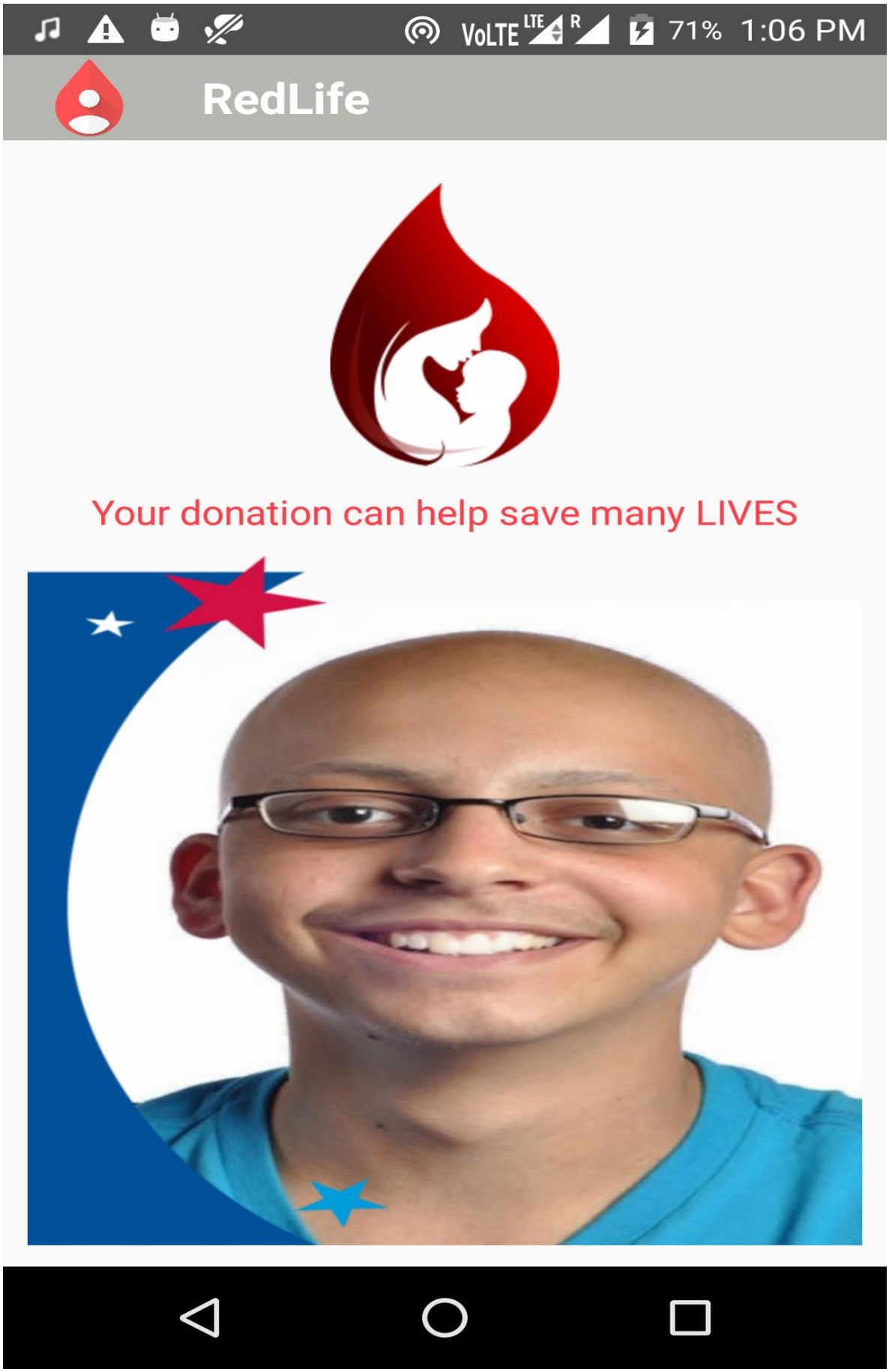


Figure 4.5 Home Screen 2

4.1.6 Home Page (With 3rd Image)

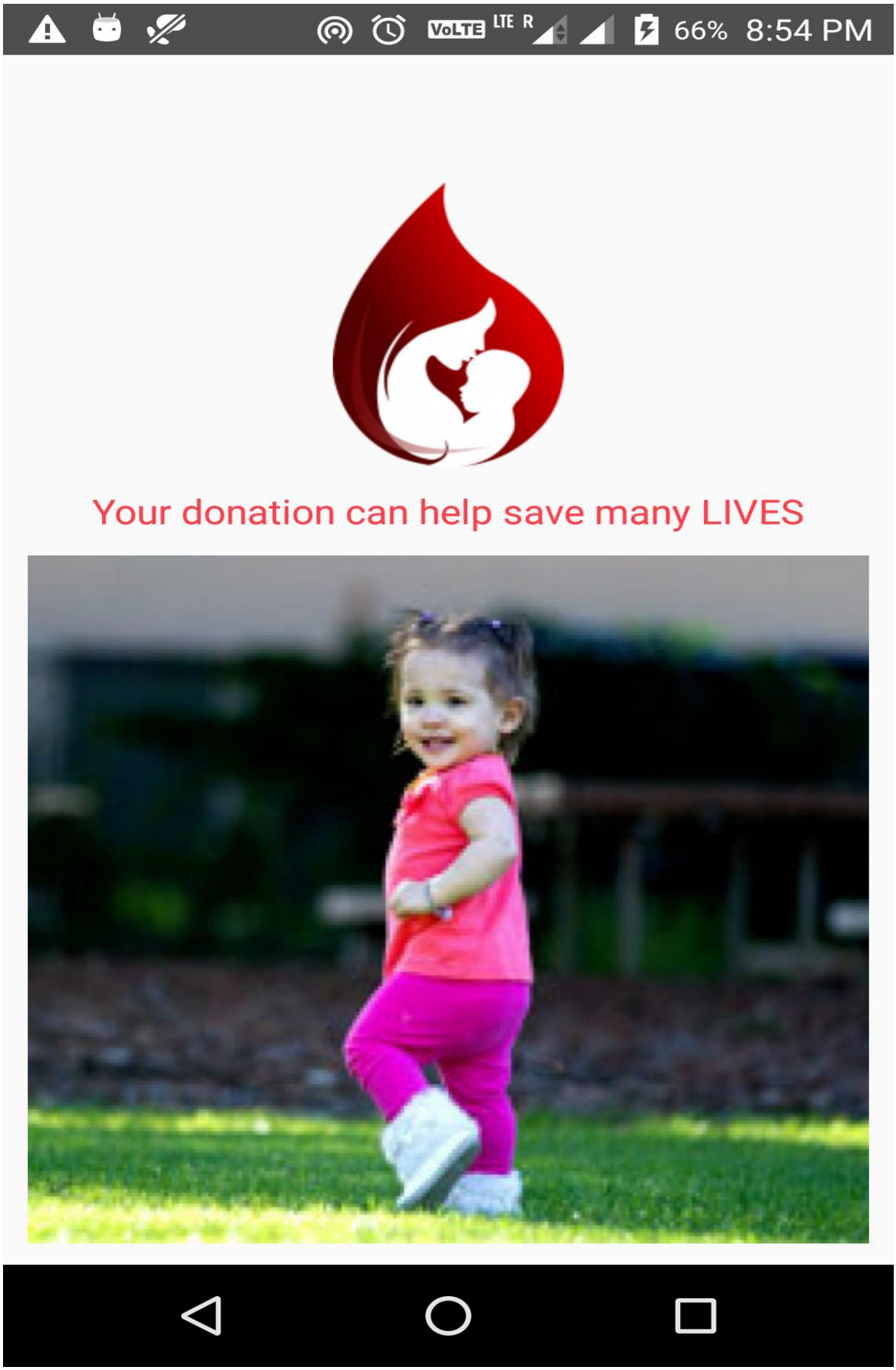


Figure 4.6 Home Screen 3

4.1.7 Home Page with Navigation Drawer

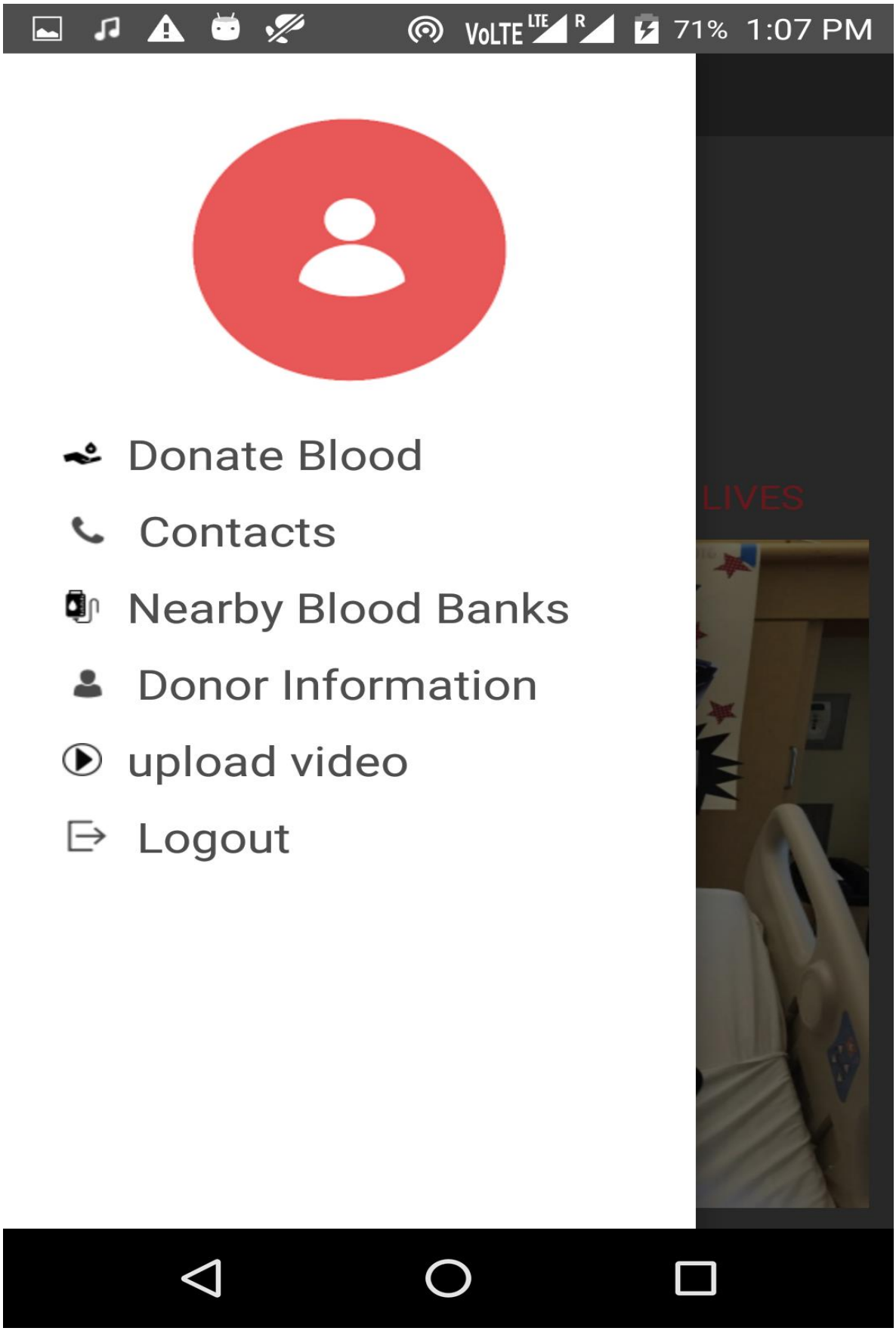


Figure 4.7 Navigation Drawer

4.1.8 Screen display on selecting Donate Blood

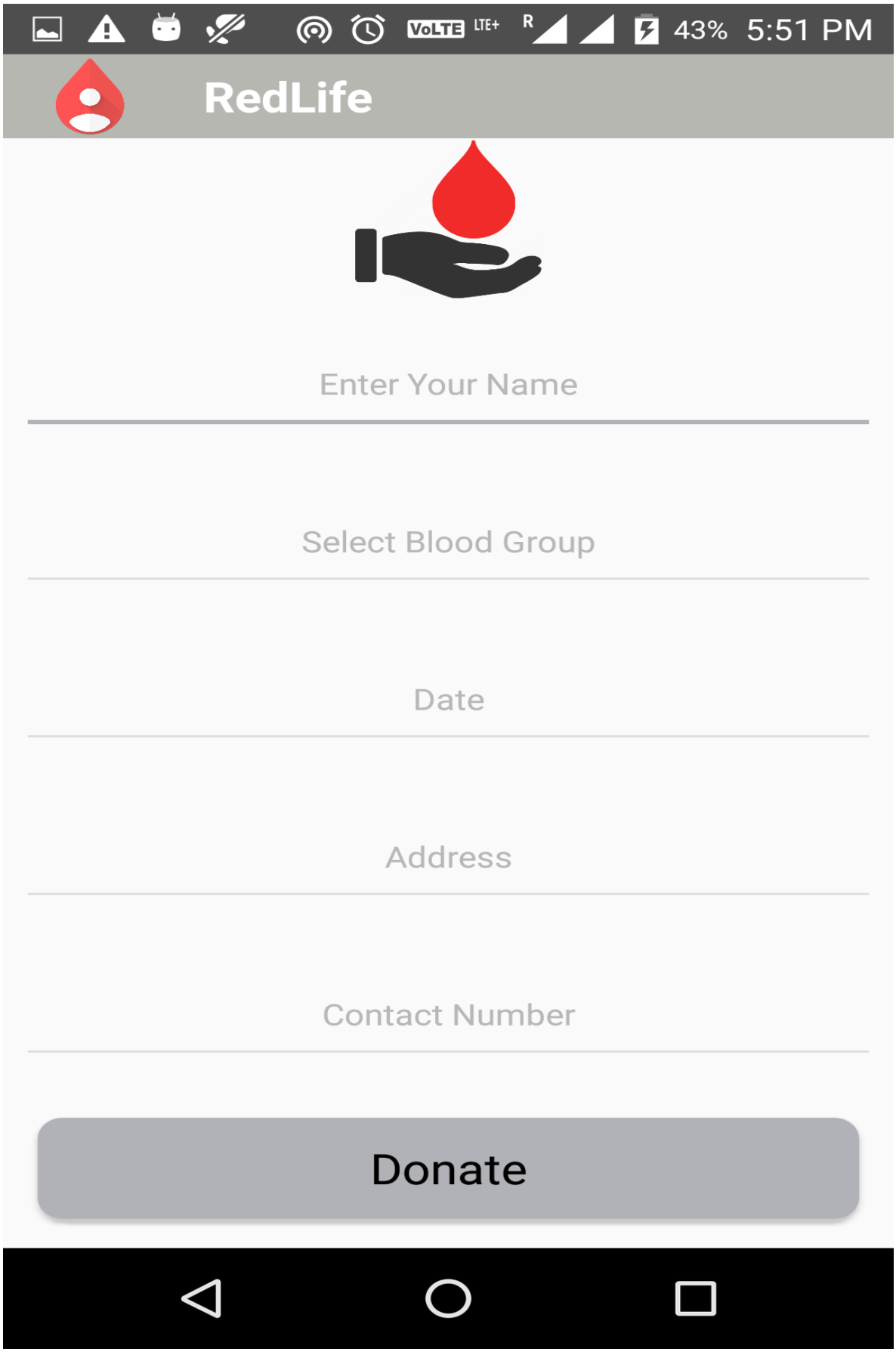


Figure 4.8 Donate Blood

After form filling

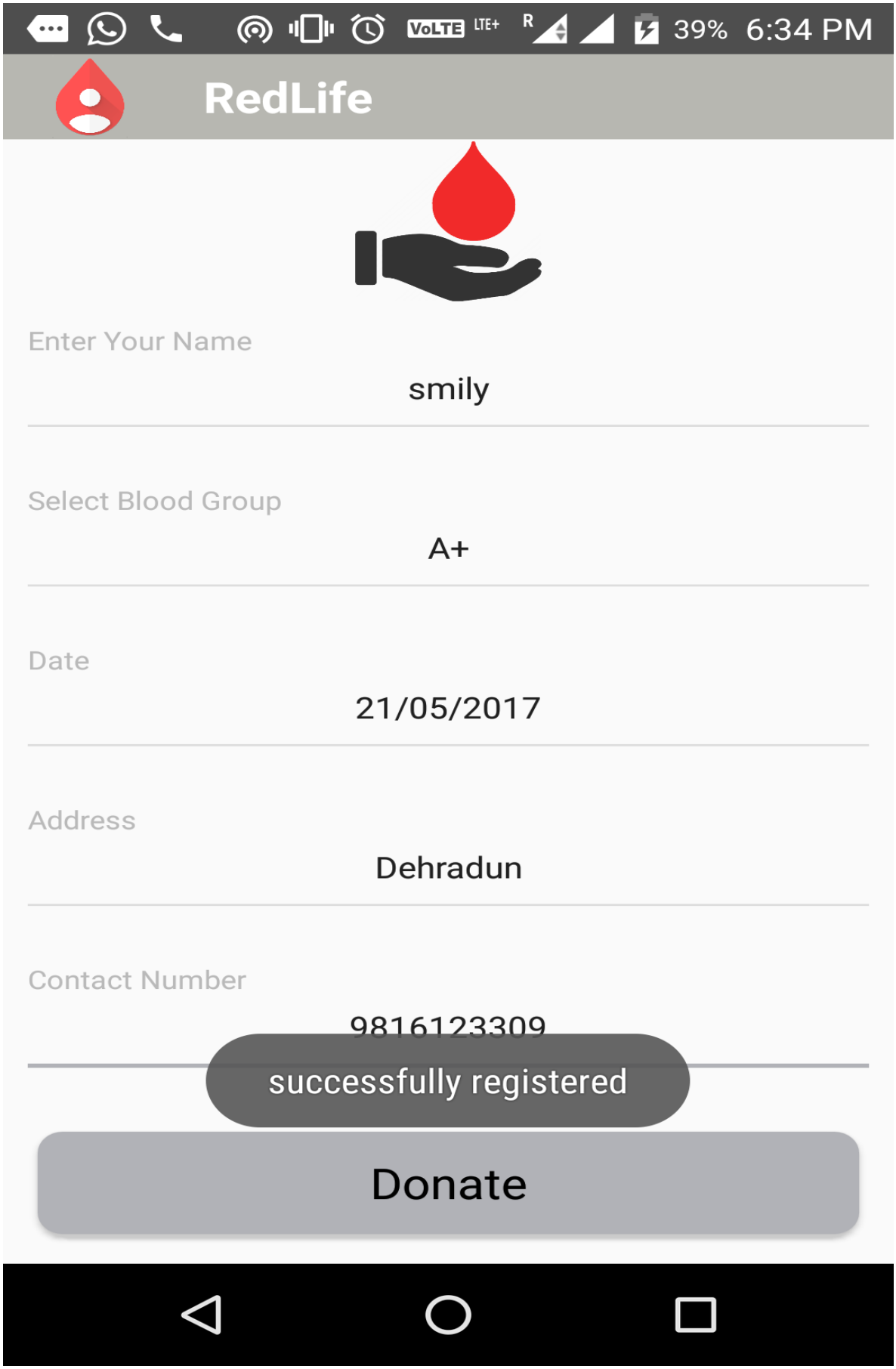


Figure 4.9 Filled Donate Blood Form

4.1.9On selecting Contacts menu

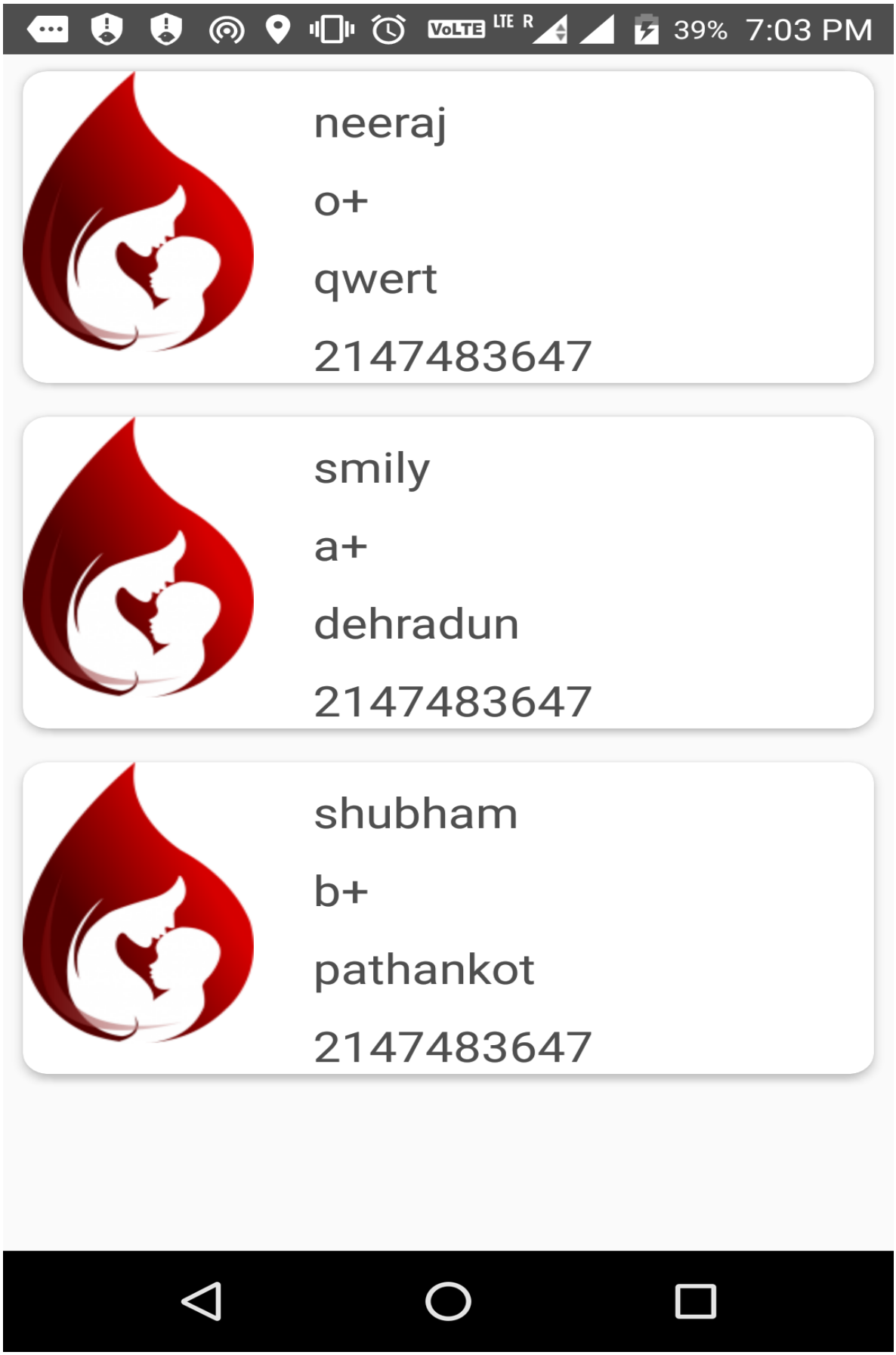


Figure 4.10 Contacts

4.1.10On selecting Nearby Blood Banks

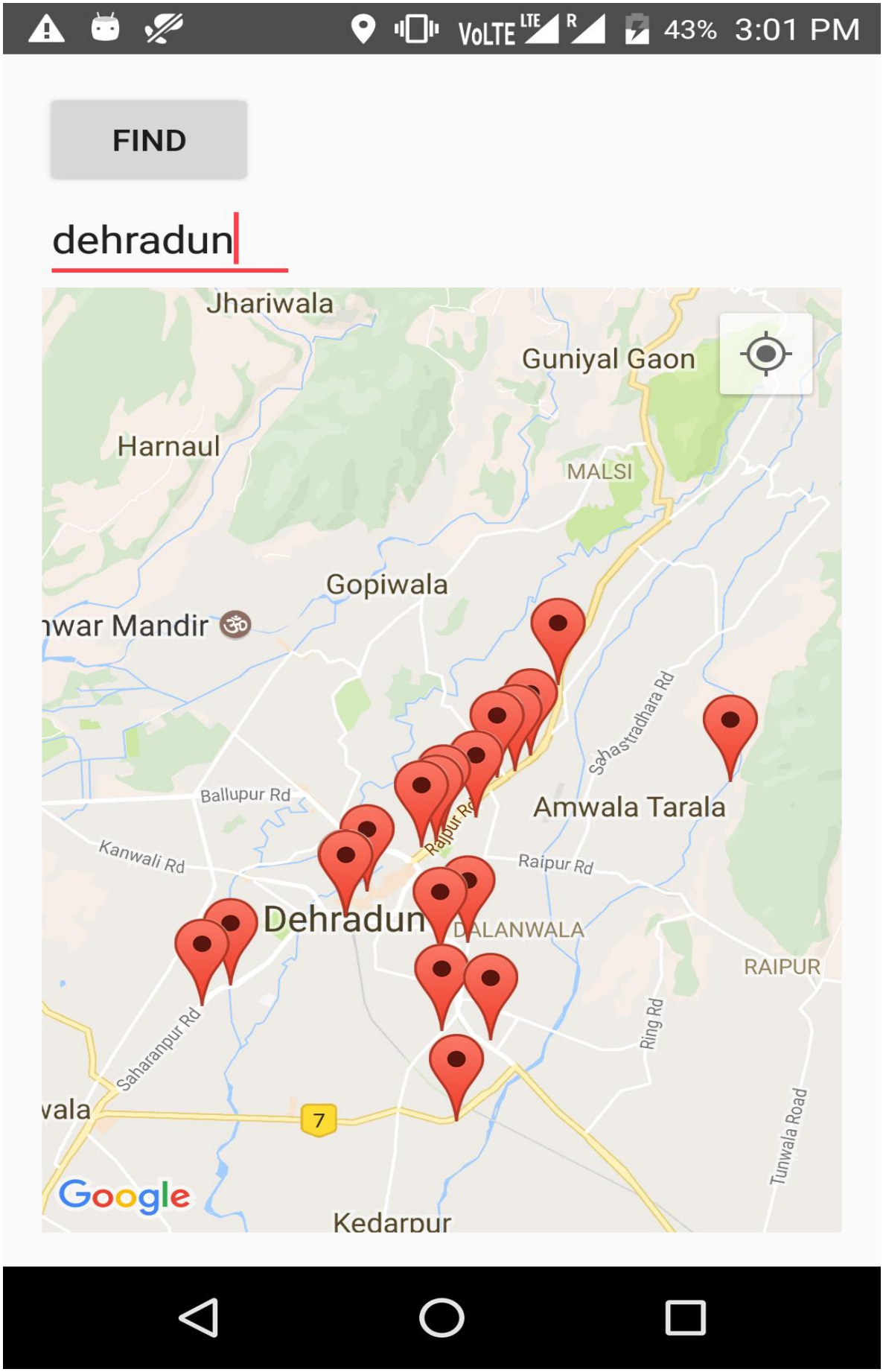


Figure 4.11Nearby Blood Banks

4.1.11On selecting upload video

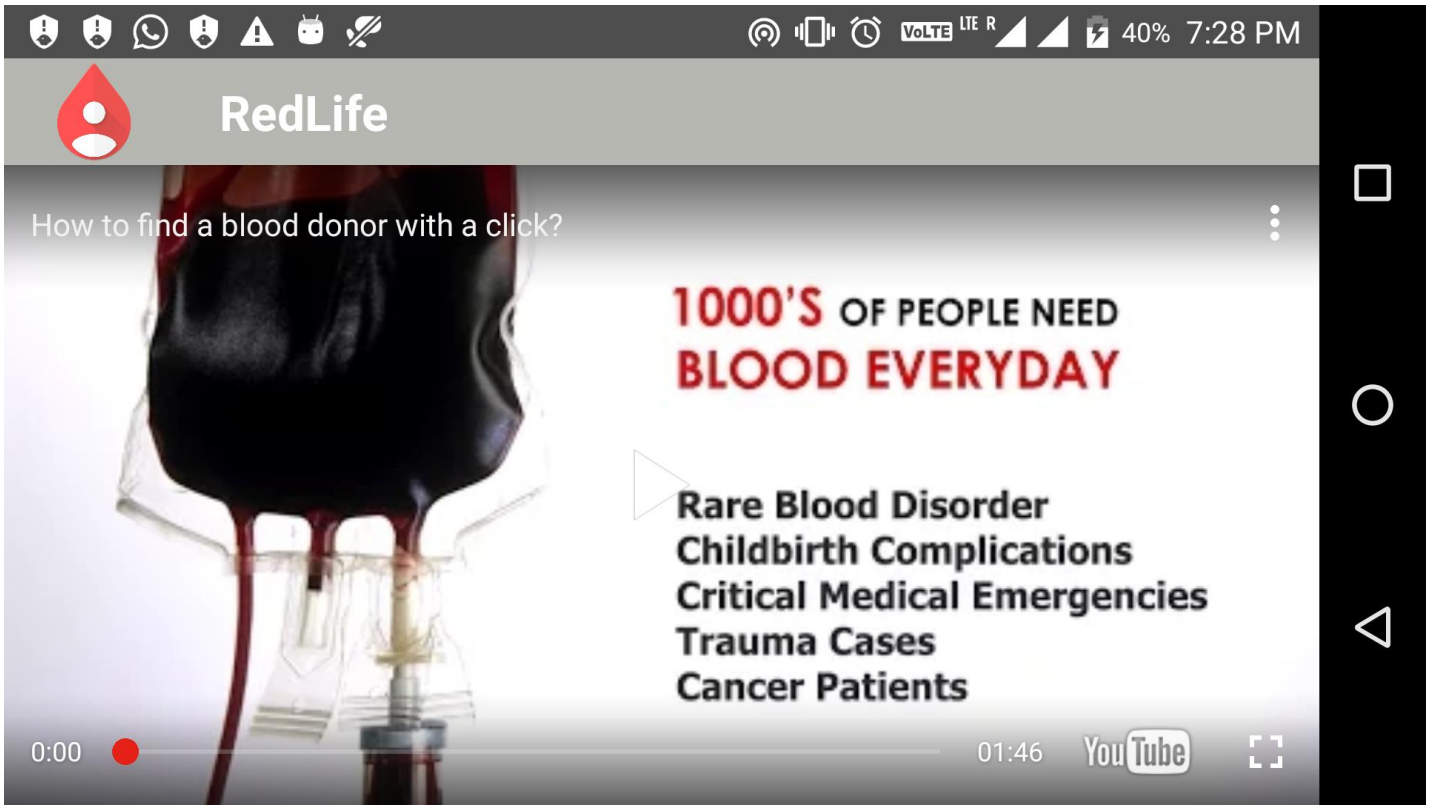


Figure 4.12 Upload Video

4.1.12On selecting Logout menu

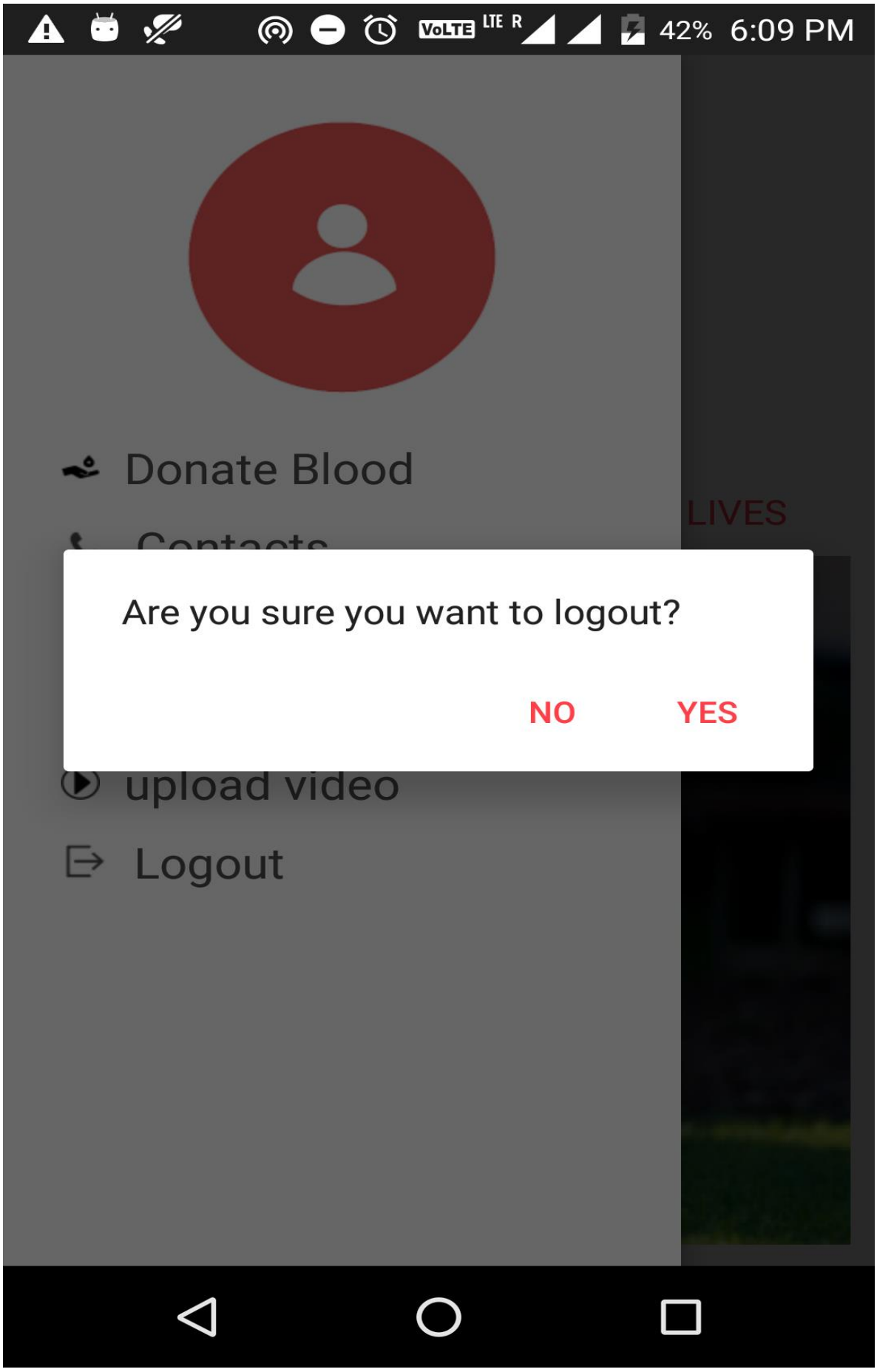


Figure 4.11 Logout

4.2TESTING PLAN

Software Testing has a dual function; it is used to identify the defects in program and it is used to help judge whether or not program is usable in practice. Thus software testing is used for validation and verification, which ensure that software conforms to its specification and meets need of the software customer.

Developer adopted Alpha testing, which usually comes in after the basic design of the program has been completed for testing by the same party. The project scientist will look over the program and give suggestions and ideas to improve or correct the design. They also report and give ideas to get rid of around any major problems. There is bound to be a number of bugs after a program have been created.

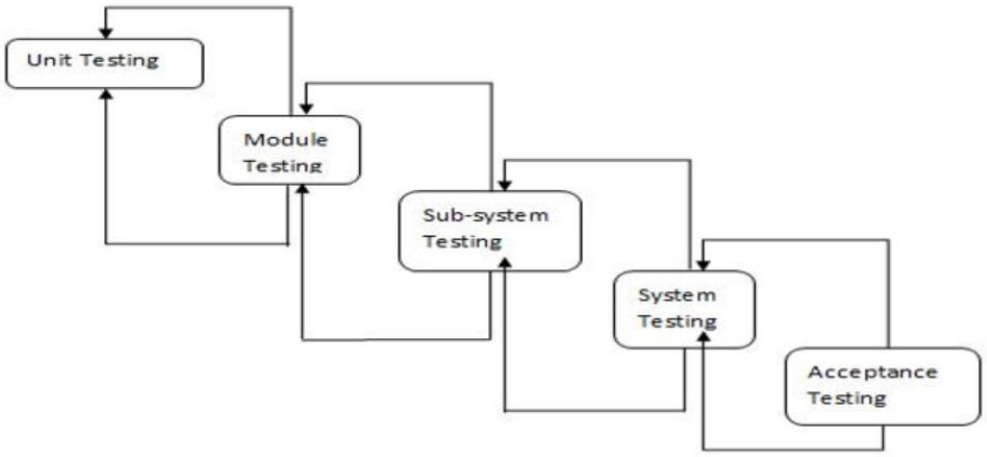


Figure 4.12Testing steps

Branch coverage:
It is strategy in which test cases are designed to make each branch condition assume true & false values.

Conditional coverage:
In this testing test cases are designed to make each component of composite conditional expression both true & false.

The Testing Process：
In this software process activities such as Design, Implementation, and Requirement Engineering is tested. Because, design errors are very costly to repair once system has been started to operate, it is quite obvious to repair them at early stage of the system. So analysis plays most important role in any project.

4.2.1 Unit Testing
The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

4.2.2 Integration Testing
After our individual modules were tested out we proceed to the integration testing to create a complete system. This integration process involves building the system and testing the resultant system for problems that arise from component interactions.

We have applied top-down strategy to validate high-level components of a system before design and implementations have been completed. Our development process started with high-level components and we worked down the component hierarchy.

4.2.3 System Testing

System testing is actually a series of tests whose purpose is to fully exercise the computer-based system. It verifies that system elements have been properly integrated and perform allocated functions. It checks whether the system as a whole works as per requirement.

4.2.4 Performance Testing

This is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout each steps in the testing process. Our system is checked for high load as well as low load.

4.2.5 Statistical Testing

Statistical Testing is used to test the program's performance and reliability and to check how it works under operational conditions. Tests are performed to detect and analyse the actual user inputs and their frequency.

4.3 TEST CASES

A test case is a set of conditions or variables and inputs that are developed for a particular goal or objective to be achieved on a certain application to judge its capabilities or features. It might take more than one test scenario to determine the true functionality of the application being tested. Every requirement or objective to be achieved efficiently needs at least one test case. Some software development methodologies like Rational Unified Process (RUP) recommend creating at least two test cases for each requirement or objective, one for performing testing through positive perspective and the other through negative perspective so that application can be tested from all aspects.

4.3.1Test Case Structure

A formal written test comprises of three test-

Information -

Information consists of general information about the test case. Information incorporates Identifier, name of the test case, purpose or brief description and test case dependencies.

Activity-

Activity consists of the actual test case scenarios. Activity contains information about the test case environment, activities to be done at test case initialization, activities to be done after test case is performed, and step by step actions to be done while testing and the input data that is to be supplied for testing.

4.3.2Designing Test Cases

- Software requirements and Hardware requirements (if any)
- Specific setup or configuration requirements
- Description on how to perform the test(s)
- Expected results or success criteria for the test

Planning experiments is tedious in a testing plan, yet they merit giving time since they can stay away from pointless retesting or investigating. Associations can adopt the test situations strategy in their own specific

manner and as indicated by their own particular points of view. Some take after a general strides while others may adjust for a more organized and complex approach. It is imperative to settle on the two extremes and judge on what might work the best. Outlining appropriate experiments is extremely essential for your product testing arranges as a great deal of bugs, ambiguities, irregularities, complexities and slip ups can be recuperated in time as additionally it helps in sparing time on investigating and re-testing experiments.

Table 4.1: Login Activity Test

Test Case	
Test scenario name	Login activity testing
Description	This scenario tests that whether the login activity works correctly or not
Module Name	Normal User
Status	Created

ID	Test Case	User Input	Expected Result	Test Result
Andro1	Login Activity Testing	User must tap login Button without Entering Any Credentials	Error Message should be shown on a toast	Successful
		User inputs invalid Login credentials.	Error message should be shown on a toast	Successful
		User Inputs valid login Credentials	Home page should be opened	Successful

Table 4.2: Register Activity Test

Test Case				
Test scenario name	Register activity testing			
Description	This scenario tests that whether the Register activity works correctly or not			
ID	**Test Case**	**User Input**	**Expected Result**	**Test Result**
Andro_2	Register Activity Testing	User taps Register button without entering any information	Error Message should be toasted	Successful
		User inputs incomplete registration form	Error message should be shown on a toast	Successful
		User Inputs valid information and registers	User should be added to database and success message should be shown on a toast	Successful

Table 4.3: Donate Blood Fragment Test

Test Case

Test scenario name	Donate Blood activity testing
Description	This scenario tests that whether the Donate Blood activity works correctly or not

D	Test Case	User Input	Expected Result	Test Result
Andro_3	Donate Blood Activity Testing	User Taps donate button without entering any information	Error Message should be toasted	Successful
		User inputs incomplete donation form	Error message should be shown on a toast	Successful
		User Inputs valid information and clicks donate button	Donor should be added to database and success message should be shown on a toast	Successful

Table 4.4: Find Blood Banks Fragment Test

Test Case

Test scenario name	Find Blood Banks activity testing
Description	This scenario tests that whether the Find Blood Banks activity works correctly or not

ID	Test Case	User Input	Expected Result	Test Result
Andro_4	Find Blood Banks Activity Testing	User taps find button without entering any Information	Error Message should be toasted	Successful
		User Inputs valid information and clicks search button	The entered query and relevant results should be marked on the map	Successful

Table 4.5: Contact Blood Donors fragment Test

Test Case				
Test scenario name		Need Blood activity testing		
Description		This scenario tests that whether the Need Blood activity works correctly or not		
ID	**Test Case**	**User Input**	**Expected Result**	**Test Result**
Andro_5	Need Blood Activity Testing	-	Available Donor list should be displayed	Successful

CHAPTER 5
CONCLUSION

5.1 CONCLUSION

The fundamental reason for this venture is to create and outline a framework that will associate all contributors. The framework will help and control a blood transfusion benefit and make an all around kept up database to hold information on loads of blood gatherings and blood in every zone as information on contributors in every city. Moreover, individuals will have the capacity to check which patients require blood supplies through the blood giving application. They will have the capacity to enroll as blood benefactors and get ask for from their nearby customers who needs blood to give blood in instances of need.

5.2 FUTURE ENHANCEMENTS

Future enhancements include addition of new features and updating some older modules or components as follows:

- Integration of SMS Gateway to inform and reach out more users not having smartphones.
- Inclusion of More cities.
- Changing the preferred database as the number of users grows.

APPENDICES

USER MANUAL

1. We have made an app on Blood Donation service.
2. It is a two way app in which blood donor can donate and receiver can receive it at the same time.
3. In it Donor can fill details like (address and phone No.) to create his donation on the Database
4. From the same database Receiver can check the availability according to the need otherwise can create its request for the requirement.
5. Every registered user can check History, Eligibility etc.
6. The Receiver can get address and phone No. of Donor and can him/her easily

REFRENCES

[1] Ministry of Health and Family Welfare **National AIDS Control Organization Government of India**. Name, Voluntary Blood Donation Programme, 2007.

[2] Medline Plus, U.S.National Library of Medicine, NIH, USA. **Blood Transfusion and Donation**, http://www.nlm.nih.gov/medlineplus/bloodtransfusionan ddonation.html, 2011.

[3] The Armed Services Blood Program (ASBP, http://www.militaryblood.dod.mil/About/default.aspx 2011.

[4] Shyam Sundaram and T. Santhanam, "**Classification of Blood Donors using Data Mining**", Proceedings of the Semantic E-Business and Enterprise Computing , 2009, pp. 145-147.

[5] Shyam Sundaram and T. Santhanam, "**Application of CART Algorithm in Blood Donors Classification**‖, Journal of Computer Science, Vol. 6, No. 5, 2010, pp.548-552.

[6] L. Breiman, J. Friedman, R. A. Olshen and C. J.Stone.Classification and regression trees, Wadsworth, 1984.

[7] Michael Chau, Eddie Cheng and Chi Wai Chan, "**Data Analysis for Healthcare: A Case Study In Blood Donation Center Analysis"**, Proceedings of Sixteenth Americas Conference on Information Systems (AMICS), 2010.

[8] Saberton, P. J., Paez, A., Newbold, K. B., and Heddle, N.M., "**Geographical Variations in the Correlates of Blood Donor Turnout Rates: An Investigation of Canadian Metropolitan Areas**‖, International Journal of Health Geographic, Vol. 8, No. 56, 2009.

[9] http://www.androidhive.info/2012/01/android-login-and-registration-with-php-mysql-and-sqlite/

[10] http://www.tutorialspoint.com/android/android_list_view.htm

[11]http://www.tutorialspoint.com/android/android_user_interface_layouts.htm

[12]http://www.w3schools.com/tags/att_button_form.asp

[13] http://www.androidhive.info/2012/07/android-gps-location-manager-tutorial/

[14] https://www.simplifiedcoding.net/android-login-example-using-php-mysql-and-volley/

[15] https://en.wikipedia.org/wiki/LAMP (software_bundle) [8]https://aws.amazon.com/free

[16] https://aws.amazon.com/ec2/

[17] http://www.androidhive.info/2011/11/android-sqlite-database-tutorial/