

BLOOD DONOR APPLICATION

Submitted in partial fulfillment of the requirements for the award
Of Bachelor of science in Computer Science

By

ABDUL RAHMAN AR (Reg.No.38290001)

AND

BHARANI KUMAR G (Reg.No.38290901)



DEPARTMENT OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SATHYABAMA

**INSTITUTE OF SCIENCE AND
TECHNOLOGY (DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600119

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119

www.sathyabama.ac.in



DEPARTMENT OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **ABDUL RAHMAN A R** (Reg No 38290001) and **BHARANI KUMAR G** (Reg No. 38290901) who carried out the project entitled "**BLOOD DONOR APPLICATION**" under my/our supervision From November 2020 to April 2021.

Internal Guide

Dr. J. ALBERT MAYAN M.E., Ph.D.,

Head of the Department

Submitted for Viva voice Examination held on

Internal Examiner

External Examiner

DECLARATION

I, **ABDUL RAHMAN A R (Reg No : 38290001)** here by declare that the Project Report entitled **“BLOOD DONOR APPLICATION”** done by me and my Partner **BHARANI KUMAR G (Reg No : 38290901)** under the guidance of **Dr.ALBERT MAYAN J**, is submitted in partial fulfillment of the requirements for the award of **Bachelor Of SCIENCE IN COMPUTER SCIENCE**

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks for **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean** , School of Computing , **Dr. S. Vigneshwari M.E., Ph.D. and Dr. L. Lakshmanan M.E., Ph.D., Professor and Head** of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. J. ALBERT MAYAN** for his valuable guidance , suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non- teaching staff members of the **Department of Bachelor of science in Computer Science** who were help full in many ways for the completion of the project.

BLOOD DONOR APPLICATION

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT LIST OF FIGURES LIST OF SYMBOLS LIST OF ABBREVIATIONS	(i) (ii) (iii) (iv)
1.	CHAPTER 1 : INTRODUCTION 1.1 GENERAL 1.2 EXISTING SYSTEM 1.2.1 EXISTINGSYSTEM DISADVANTAGES 1.2.2 LITERATURE SURVEY 1.3 PROPOSED SYSTEM 1.3.1 PROPOSED SYSTEM ADVANTAGES 1.4 TABLE OF EXISTING AND PROPOSED SYSTEM	1 1 2 2 3 6 6 7
2.	CHAPTER 2 :PROJECT DESCRIPTION 2.1 MODULES NAME 2.1.2 MODULES EXPLANATION AND DIAGRAM 2.2 DATABASE DESIGN 2.3 TECHNIQUE OR ALGORITHM	9 9 9 13 13
3.	CHAPTER 3 : REQUIREMENTS 3.1 GENERAL 3.2 HARDWARE REQUIREMENTS 3.3 SOFTWARE REQUIREMENTS	14 14 14 14

4.	CHAPTER 4 : SYSTEM DESIGN 4.1 GENERAL 4.1.2 USE CASE DIAGRAM 4.1.3 CLASS DIAGRAM 4.1.4 ACTIVITY DIAGRAM 4.1.5 SEQUENCE DIAGRAM 4.1.6 COMPONENT DIAGRAM 4.1.7 COLLOBORATION	15 15 16 17 18 19 20 22
5.	CHAPTER 5 :SOFTWARE SPECIFICATION 5.1 GENERAL 5.1.1 HISTORY 5.2.1 INTRODUCTION OF ANDROID 5.2.2 ANDROID VERSIONS 5.2.3 ANDROID ARCHITECTURE 5.2.4 SECURITY 5.2.5 FEATURES OF ANDROID 5.2.6 ADVANTAGES OF ANDROID 5.2.7 DISADVANTAGE OF ANDROID 5.2.8 LIMITATIONS OF ANDROID	23 23 23 23 25 25 26 26 27 27 28
6.	CHAPTER 6 : SOFTWARE TESTING 6.1 GENERAL 6.2 DEVELOPING METHODOLOGIES 6.3 TYPES OF TESTING	29 29 29 29
7.	CHAPTER 7 : SNAPSHOTS 7.1 GENERAL 7.2 VARIOUS SNAPSHOTS AND CODING	32 32 32

8.	CHAPTER 8 :APPLICATIONS AND FUTURE ENHANCEMENT 8.1 APPLICATIONS 8.2 FUTURE ENHANCEMENTS	47 47 47
9.	CHAPTER 9: 9.1 REFERENCES	48 48

Abstract :

The Android Blood Bank project is one of the great technical life saver applications.

The main aim of this application is to reduce the time spent in searching for blood donors in case of emergency.

There is an assumption that the blood will consistently be there when it is truly required. Blood donor comprise the primary stockpile source in a compelling blood inventory network the board. They feed blood stocks through their gift.

In a crisis circumstance, if the stocks are inadequate, the lone wellspring of blood supply will be individuals who go to the wellbeing community and give the blood on an intentional premise. It is sure that time is a vital part in such circumstance.

Thus, the medical care community should call the closest accessible benefactor to guarantee to get the help as fast as could really be expected.

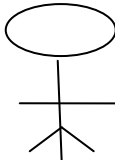
This system will keep track of all the latest updates relevant to the people who donate blood and who had donated blood in case of emergency.

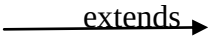
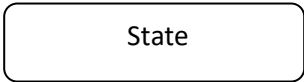
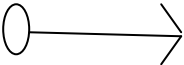
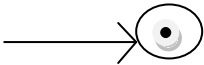
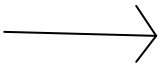
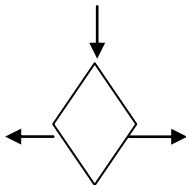
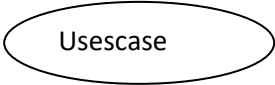
An advanced mobile application is created to encourage the ID of the closest accessible blood contributor volunteer and the correspondence with him/her in the crisis circumstances where the blood can't be provided through the blood donation centers' stocks. In this paper this application will be introduced.

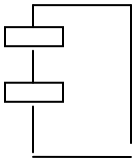
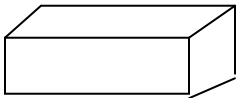
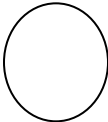
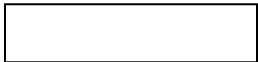
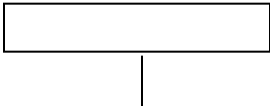
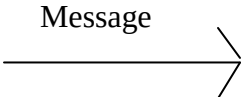
LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO.
2.3.2	Module Diagram	9-12
4.1.2	Use case Diagram	16
4.2.2	Class Diagram	17
4.2.3	Activity Diagram	18
4.2.4	Sequence Diagram	19
4.2.5	Component Diagram	20

LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	<i>+ public</i> <i>- private</i> <i># protected</i> <i>Class Name</i> <i>attribute</i> <i>- attribute</i> <i>+ operation</i> <i>+ operation</i> <i>+ operation</i>	Represents a collection of similar entities grouped together.
2.	Association	NAME Class A Class B Class A Class B	Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single classes.
5.	Aggregation	Class A ↑ Class B Class A ↑ Class B	Interaction between the system and external environment

5.	Relation (uses)	uses	Used for additional process communication.
6.	Relation (extends)		Extends relationship is used when one use case is similar to another use case but does a bit more.
7.	Communication		Communication between various use cases.
8.	State		State of the process.
9.	Initial State		Initial state of the object
10.	Final state		Final state of the object
11.	Control flow		Represents various control flow between the states.
12.	Decision box		Represents decision making process from a constraint
13.	Usecase		Interact ion between the system and external environment.

14.	Component		Represents physical modules which is a collection of components.
15.	Node		Represents physical modules which are a collection of components.
16.	Data Process/State		A circle in DFD represents a state or process which has been triggered due to some event or action.
17.	External entity		Represents external entities such as keyboard, sensors, etc.
18.	Transition		Represents communication that occurs between processes.
19.	Object Lifeline		Represents the vertical dimensions that the object communications.
20.	Message		Represents the message exchanged.

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1.	.dex	Dalvik Executable File
2.	DVM	Dalvik Virtual Machine
3.	ADT	Android Development Tool
4.	SDK	Software Development Kit
5.	.apk	Application Package File
6.	JRE	Java Runtime Environment
7.	Adb	Android Debug Bridge
8.	DDMS	Dalvik Debug Monitor Service

CHAPTER 1

INTRODUCTION

1.1 GENERAL:

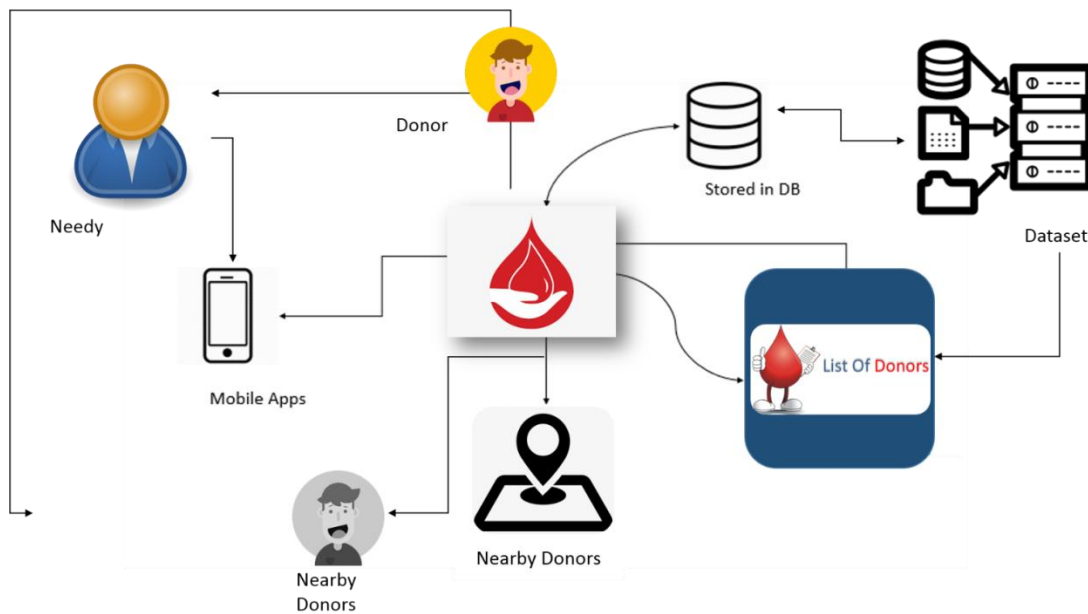
Blood giver is android application which utilizes the Cloud Real-time Database in order to amass and sort out the data of all of the blood supports with a smart and beneficial search for. (A google-services. JSON) record should be made for the android application utilizing the Firebase Console). This undertaking by then goes presumably as fundamental occupation to save the presence of individuals and which is also in like way to its fundamental issue. The undertaking of the android (The blood provider) framework was made with the objective that the clients see the data about the enrolled blood allies, for example, name, address, and some other such close to the home data nearby their simple segment of the blood social affairs and the other valuable data of parts with/supporter. This task likewise has one login page where the client's expected to select the accreditations and at definitely note thus, all things considered would he have the option to see the transparency of require blood and equivalently may comparatively then enlist himself to leave behind the necessary blood in an event which he wishes to. By then this undertaking requires web access approval permitted and a while later in this course there happens a load of the tremendous web dissatisfaction.

This android application picks the correct support online by then and flicker utilizing and accommodating unnoticeable segments which are nearby the available blood pack. It is the fundamental motivation behind making this android application to reduce opportunity, as it was which was spent for finding the correct supplier/provider and moreover the accessibility of the correct red blood required. As needs be, this android application leaves behind the particular required data paying little heed to minutes and a short time later next to helps for a snappier fundamental position. The blood is a specific kind of regular liquid which gives huge substances to body cells for example supplement and oxygen. The blood sponsor android application's a save/supporter of the red blooded blood areas gathered by virtue of blood provider which deals with and is then set something to the side for any later use forward red blood bonding, in like manner, is the blood course of action for patients, likewise ought to be agreed to any closeness justification the red blood holding. Its conceivable several conditions where the patient can't get any necessary extent of the necessary blood at the game plan time because of nonattendance of cover relationship kind of any planned informational collection among its red blood blessing centers which by then prompts nonappearance of the learning of the empowered record of all blood advertisers.

Today advantageous and besides adaptable based android applications have been changed into a piece of everyone's customary presence. With the upheaval and adaptable enrolling different, also, mind blowing highlights have been added to these fields and the mobiles have smaller, quicker and besides, better as numerous years pass. This android application's conveyed into a simple quest for the red blood needed in the coterminous zones for the crisis. In the android application, one can without a very remarkable stretch get all the indisputable permission to the important blood sensibly and at the correct spot.

1.2 EXISTING SYSYTEM :

In existing system given arrangement the clients need to initially get themselves the planned with various pressure acknowledgment and the executives techniques, in light of their last objective, regardless of whether that is pressure identification, observing and imagining, or the board. The Existing concept is a blood can't be made creation lines can basically start from liberal donors.



1.2.1 EXISTING SYSTEM DISADVANTAGES :

- Developed in Web technologies.
- Network could be exploited.

1.2.2 LITERATURE SURVEY :

- **Title** : Electronic Medical Records Adoption and Use
- **Author:** Raghid El-Yafouri, Leslie Klieb
- **Year** : 2014
- **Description:**

• The aim of this research is to explore the motives behind the adoption or rejection of Electronic Health Records (EHR) systems in the USA by medical offices. The current health care system in the United States suffers from high expenditures and poor quality. The Patient Protection and Affordable Care Act, passed in 2010, attempts to save costs and improve quality of care by offering incentives to use Electronic Health Records systems. Part of the reform by this law is dependent on the use of technology in managing patient medical and health records. The objective is to reduce redundancy and increase quality by sharing medical information amongst different health organizations like hospitals, physician offices, laboratories and clinical institutions. The success of such reform requires the participation and collaboration of all these entities and their patients. Prior research shows that adoption of Electronic Medical Records systems by hospitals and physician offices has been evident but at a rate that is slower than in other countries. Aside from financial barriers, technical, psychological, social/legal and organizational barriers exist. In order to understand the impact of those barriers on the adoption of Electronic Health Records management by small physician offices better, a five-level adoption model is presented to define the stages of diffusion of EHR systems. Fifteen consolidated barriers are mapped to each adoption level. This research concentrates on smaller physician offices because hospitals and larger institutions are more ready and capable of adoption, according to previous research. The Diffusion of Technology Model by Rogers, the Theory of Planned Behavior by Ajzen and Fishbein, and Davis' Technology Acceptance Model are combined and extended. This model will be used to empirically measure physicians' attitudes, knowledge, social and legal influences, subjective norm and systems' ease of use and usefulness amongst other variables. These variables are applied as mediators or moderators of the intention and decision to adopt or move into subsequent levels of adoption with the goal of seeing what drives those decisions.

- **Title** : Research of Access Control in Electronic Medical Record Based on UCON

- **Author:** Ju Jianguo, Chen Weihe, Zhan Yongzhao

- **Year** : 2011

- **Description:**

- Access control mechanisms are vital to the privacy preservation and information security in electronic medical record system. In this paper, we analyze the existing drawbacks in traditional access control models firstly and outline the characteristics of next generation access control UCON. Then we apply the idea of UCON to electronic medical record system to meet the challenge of confidentiality, privacy preservation and data integrity. Then the novel access control formalisms of electronic medical record are presented. Various applications are presented to demonstrate its effectiveness.

- **Title** : Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption

- **Author:** Ming Li, Shucheng Yu, Yao Zheng, Kui Ren and Wenjing Lou

- **Year:** 2013

- **Description:**

- Personal health record (PHR) is an emerging patient-centric model of health information exchange, which is often outsourced to be stored at a third party, such as cloud providers. However, there have been wide privacy concerns as personal health information could be exposed to those third party servers and to unauthorized parties. To assure the patients' control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing. Yet, issues such as risks of privacy exposure, scalability in key management, flexible access, and efficient user revocation, have remained the most important challenges toward achieving fine-grained, cryptographically enforced data access control. In this paper, we propose a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semitrusted servers. To achieve fine-grained and scalable data access control for PHRs, we leverage attribute-based encryption (ABE) techniques to encrypt each patient's PHR file. Different from previous works in secure data outsourcing, we focus on the multiple data owner scenario, and divide the users in the PHR system into multiple security domains that greatly reduces the key management complexity for owners and users. A high degree of patient privacy is guaranteed simultaneously by exploiting multiauthority ABE. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation

and break-glass access under emergency scenarios. Extensive analytical and experimental results are presented which show the security, scalability, and efficiency of our proposed scheme.

- **Title** : An Efficient Cloud-based Personal Health Records System Using Attribute-Based Encryption and Anonymous Multi-Receiver Identity-Based Encryption

- **Author** : Chang-Ji Wang, Xi-Lei Xu, Dong-Yuan Shi, Wen-Long Lin

- **Year**: 2014

- **Description**:

- As an emerging patient-centric model of health information exchange, cloud-based personal health record (PHR) system holds great promise for empowering patients and ensuring more effective delivery of health care. In this paper, we propose a novel patient-centric cloud-based secure PHR system, which allows patients to securely store their PHR data on the semi-trusted cloud service providers, and selectively share their PHR data with a wide range of users, including health care provider like doctors and nurses, family members or friends. To reduce the key management complexity for owners and users, we divide the users in the cloud-based PHR system into two security domains named public domain and personal domain. Different from previous cloud-based PHR system, PHR owners encrypt their PHR data for the public domain using ciphertextpolicy attribute-based encryption scheme, while they encrypt their PHR data for the personal domain using anonymous multi-receiver identity-based encryption scheme. Only authorized users whose credentials satisfy the specified ciphertextpolicy or whose identities belong to dedicated identities can decrypt the encrypted PHR data, where ciphertext-policy or dedicated identities are embedded in the encrypted PHR data. Extensive analytical and experimental results are presented which show the patient-centric cloud-based secure PHR system is secure, scalable and efficient.

- **Title** : Designing Privacy Information Protection of Electronic Medical Records

- **Author:** Tzu-Wei Tseng ,Cheng-Yi Yang, Chien-Tsai Liu*

- **Year:** 2016

- **Description:**

- In recent years, the interoperability, exchange, privacy, and security of electronic medical records (EMR) across healthcare institution has become an important international issue within medical informatics. Currently Taiwan's electronic medical records exchange center (EEC) system lacks a mechanism for patients to consent to releasing only portions of their EMR. Currently patients must consent to total access when transferring EMR across healthcare institution. This study uses the IHE BPPC profile and HL7 Confidentiality code to develop privacy policies matrices and patient privacy protect mechanism. Although adding confidentiality codes in Taiwan's electronic medical records would increase the workload for physicians and other healthcare professionals, it would also ensure greater security in the transmission of sensitive information , thereby enhancing the level of security for patients.

1.3 PROPOSED SYSTEM :

In this paper we propose a mobile application is created to encourage the ID of the closest accessible blood donor and the correspondence with him/her in the crisis circumstances where the blood can't be provided through the blood donation centers' stocks. A proposed setting mindful pressure the executive's framework. The vast majority of the proposed frameworks use setting information to give constant pressure checking and representation.

1.3.1 PROPOSED SYSTEM ADVANTAGES :

- ❖ The advantage to using these tools is that you can develop the content once and then revise as necessary.
- ❖ The remote strategy empowers stream of info that'll work a lot more quickly and advantageously.

1.4 TABLE OF EXISTING AND PROPOSED SYSTEM:

EXISTING SYSTEM	PROPOSED SYSTEM
EXISTING CONCEPT:- The Existing concept is a blood can't be made creation lines can basically start from liberal donors.	PROPOSED CONCEPT:- In this paper we propose a mobile application is created to encourage the ID of the closest accessible blood donor and the correspondence with him/her in the crisis circumstances where the blood can't be provided through the blood donation centers' stocks.
EXISTING ALGORITHM:- Sorting Algorithm	PROPOSED ALGORITHM:- K-Means Algorithm
ALGORITHM DEFINITION:- The user prepared the document using Sorting Algorithm	ALGORITHM DEFINITION:- One of the most common clustering algorithms, k-means works by creating a k number of groups from a set of objects based on the similarity between objects. It may not be guaranteed that group members will be exactly similar, but group members will be more similar as compared to non-group members. As per standard implementations, k-means is an unsupervised learning algorithm as it learns the cluster on its own without any external information.

DRAWBACKS:-

- Developed in Web technologies.
- Network could be exploited.

ADVANTAGES:-

- The advantage to using these tools is that you can develop the content once and then revise as necessary.
- The remote strategy empowers stream of info that'll work a lot more quickly and advantageously.

CHAPTER 2

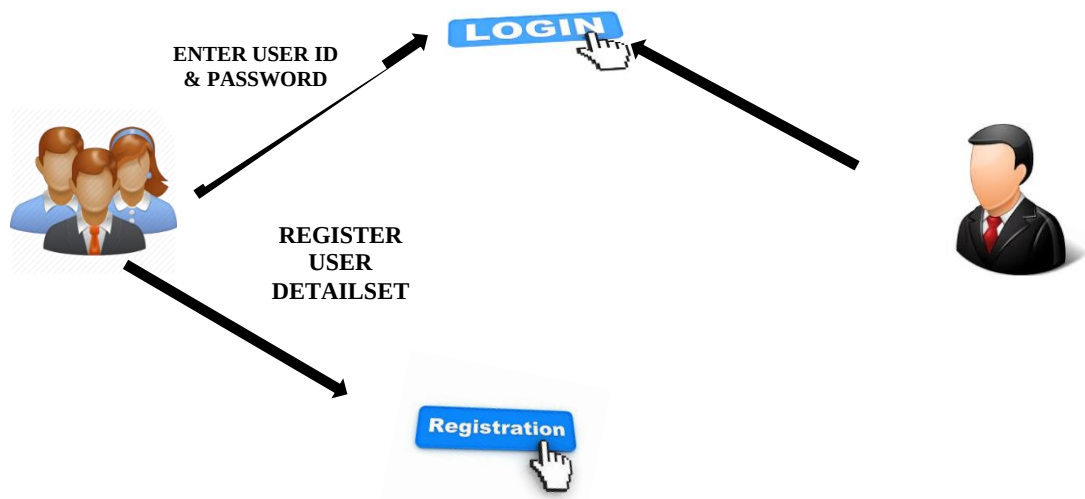
2. MODULES NAME :

- Login / Registration.
- Database Creation.
- Donor Data upload
- Donor data View/Remove
- Security Scanner
- Nearby Donor

2.1 MODULES EXPLANATION AND DIAGRAM :

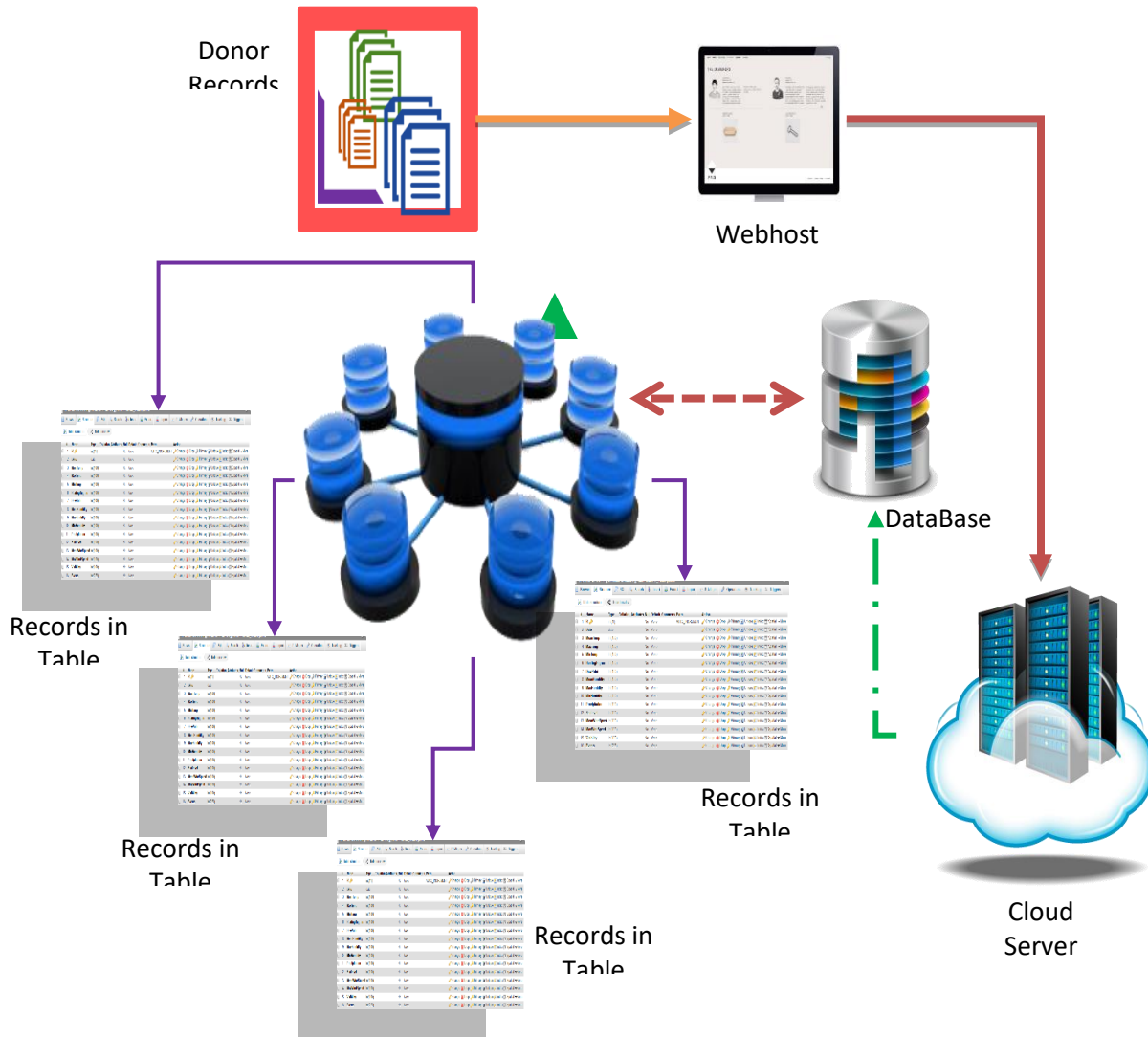
1. LOGIN & REGISTRATION:

In this module we design to develop login and signup screen. Android used xml to develop classical screens in our application. The modules describe signup page contains email id or user name, password and conform password those kind of details should be stored in database. Login screen contains email id or username and password when the user to login the app it should be retrieve the data to the database and combine based on user input if its match user name and password to allow in the app otherwise alert and show a message to the user



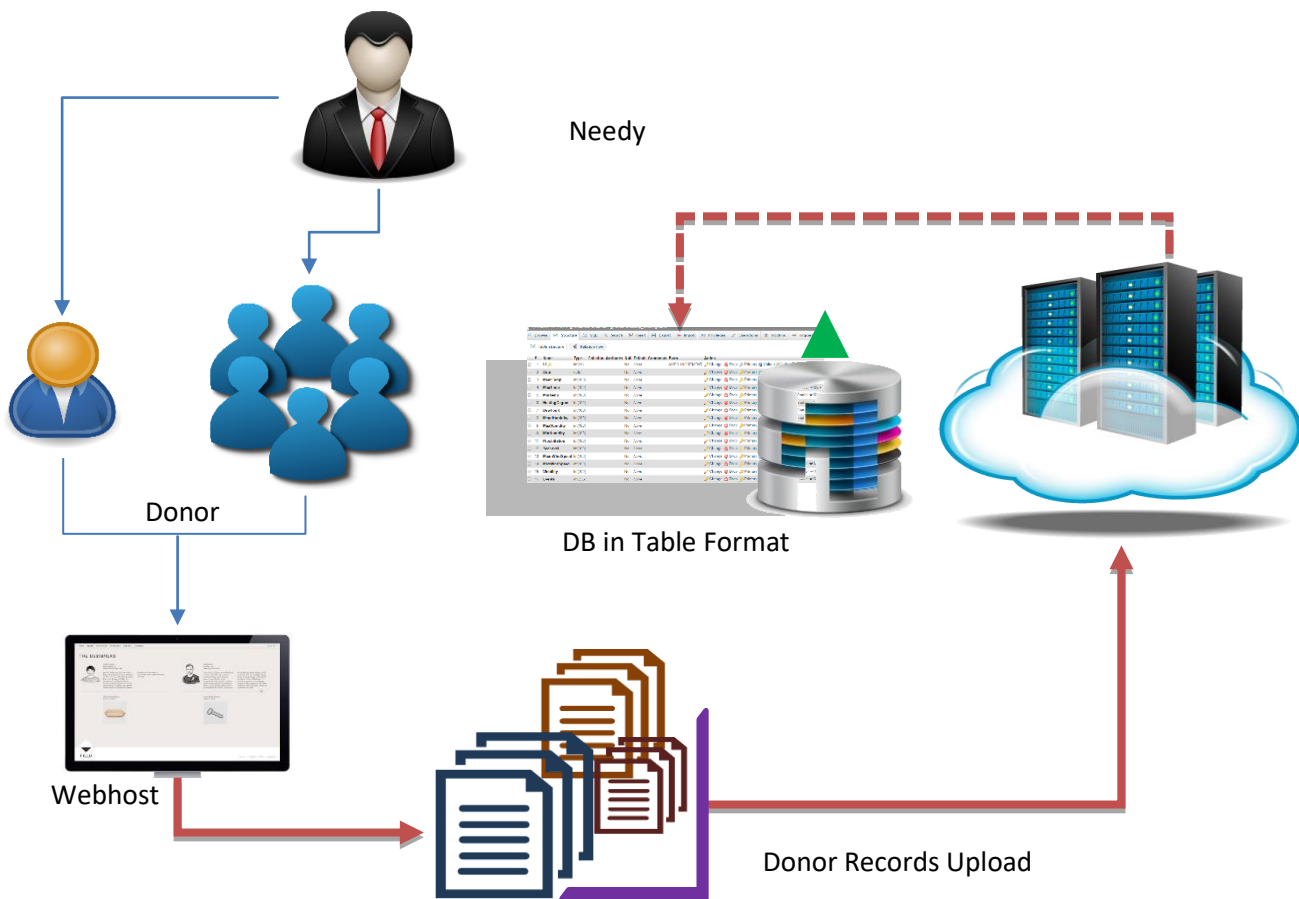
2. DATABASE CREATION:

User email id or user name and password have been stored after registration. Android used SQLite Database for storing and fetching user application details



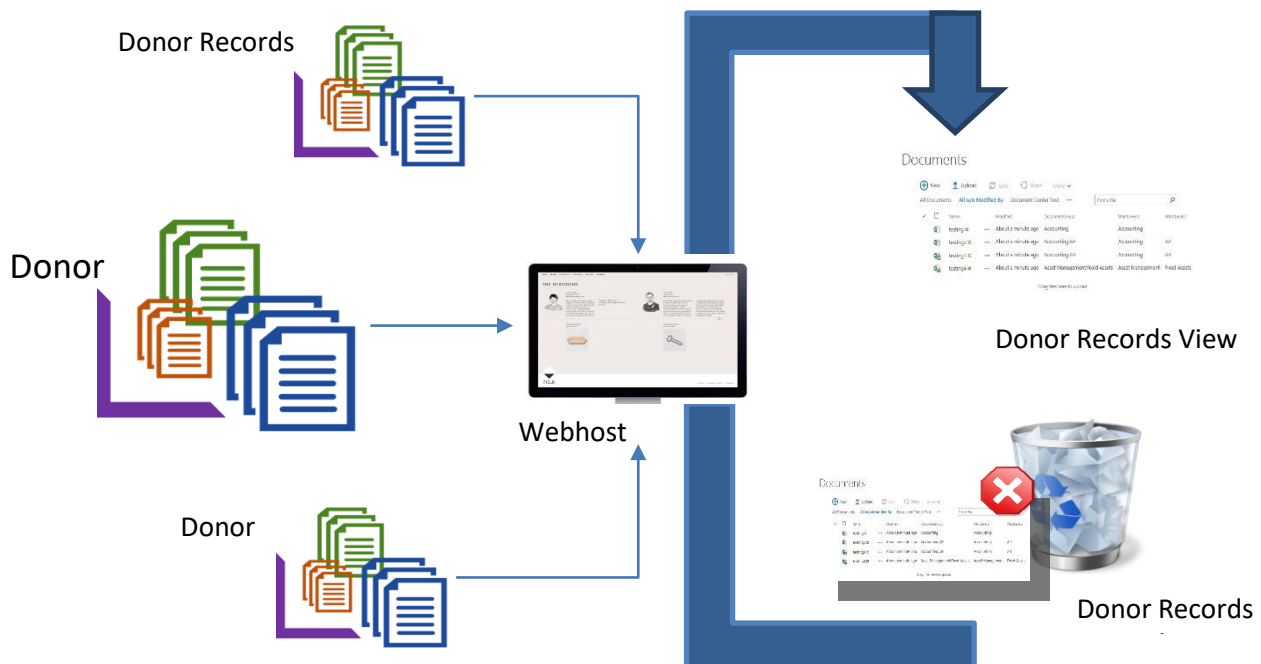
3. DONOR DATA UPLOAD:

In this module, to upload the user information in storage cloud in secure data are user information, Medical record information and patients details etc....



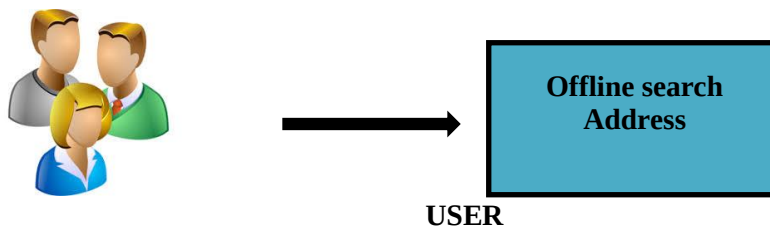
4. DONOR DATA VIEW / REMOVE:

In this module, medical data view/remove that process can be used on logged user view a medical data and incase that data can remove. But original values not delete in permanent.



➤ Nearby Donor

In this modules we have to implement the nearby donor with in a surrounding places the user can be emergencies purpose the can be access the module in our application.



2.2 DATABASE DESIGN :

❖ Registration

Fields	Data Type	Constraints
Id	Integer	Primary key
Category	Varchar	Not null
Last date	Varchar	Not null
Price	Varchar	Not null

2.3 ALGORITHM :

❖ K - Means

One of the most common clustering algorithms, k-means works by creating a k number of groups from a set of objects based on the similarity between objects. It may not be guaranteed that group members will be exactly similar, but group members will be more similar as compared to non-group members. As per standard implementations, k-means is an unsupervised learning algorithm as it learns the cluster on its own without any external information.

CHAPTER 3

REQUIREMENTS ENGINEERING

3.1 GENERAL:

These are the requirements for doing the project. Without using these tools and software's we can't do the project. So we have two requirements to do the project. They are

1. Hardware Requirements.
2. Software Requirements.

3.2 HARDWARE REQUIREMENTS:

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

PROCESSOR	:	Intel Core i3.
RAM	:	4 GB DDR2 RAM
MONITOR	:	15" COLOR
HARD DISK	:	100 GB

3.3 SOFTWARE REQUIREMENTS:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

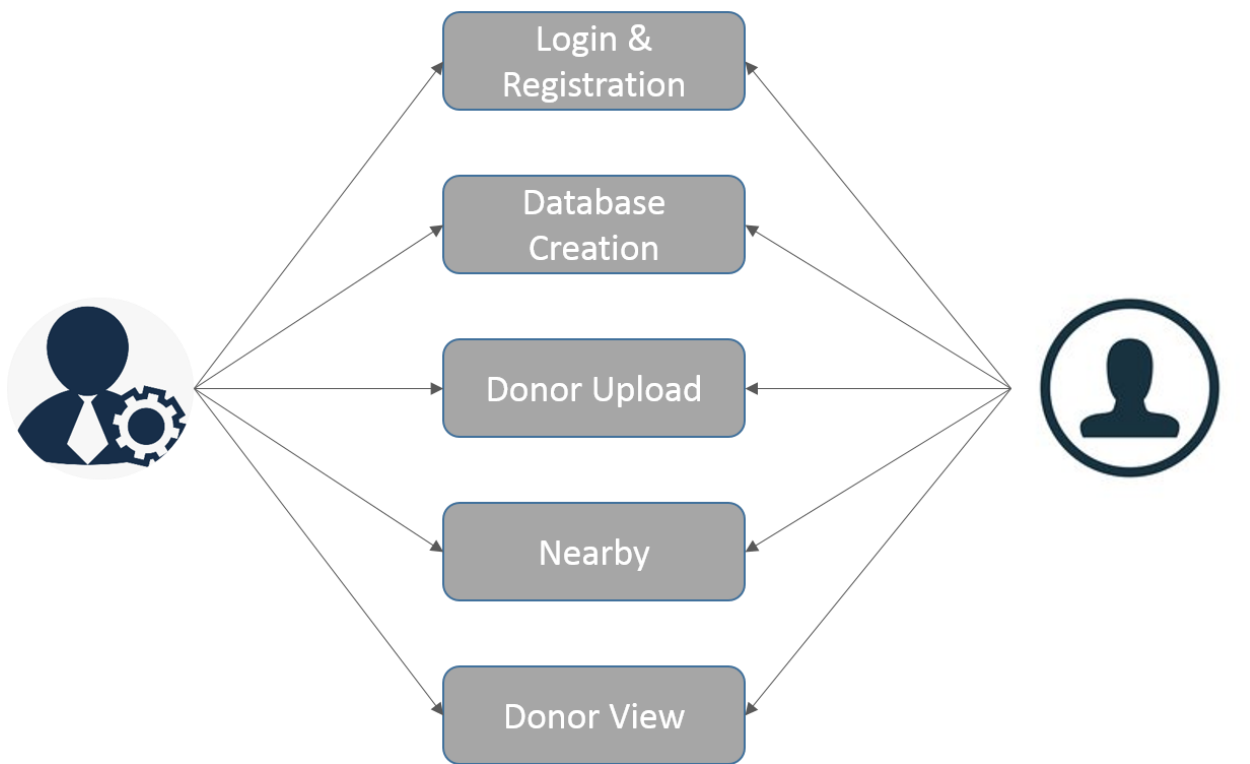
Front End	:	ANDROID XML, JAVA
Back End	:	SQLITE
Operating System	:	Windows 07
IDE	:	Eclipse, Android Studio

CHAPTER 4

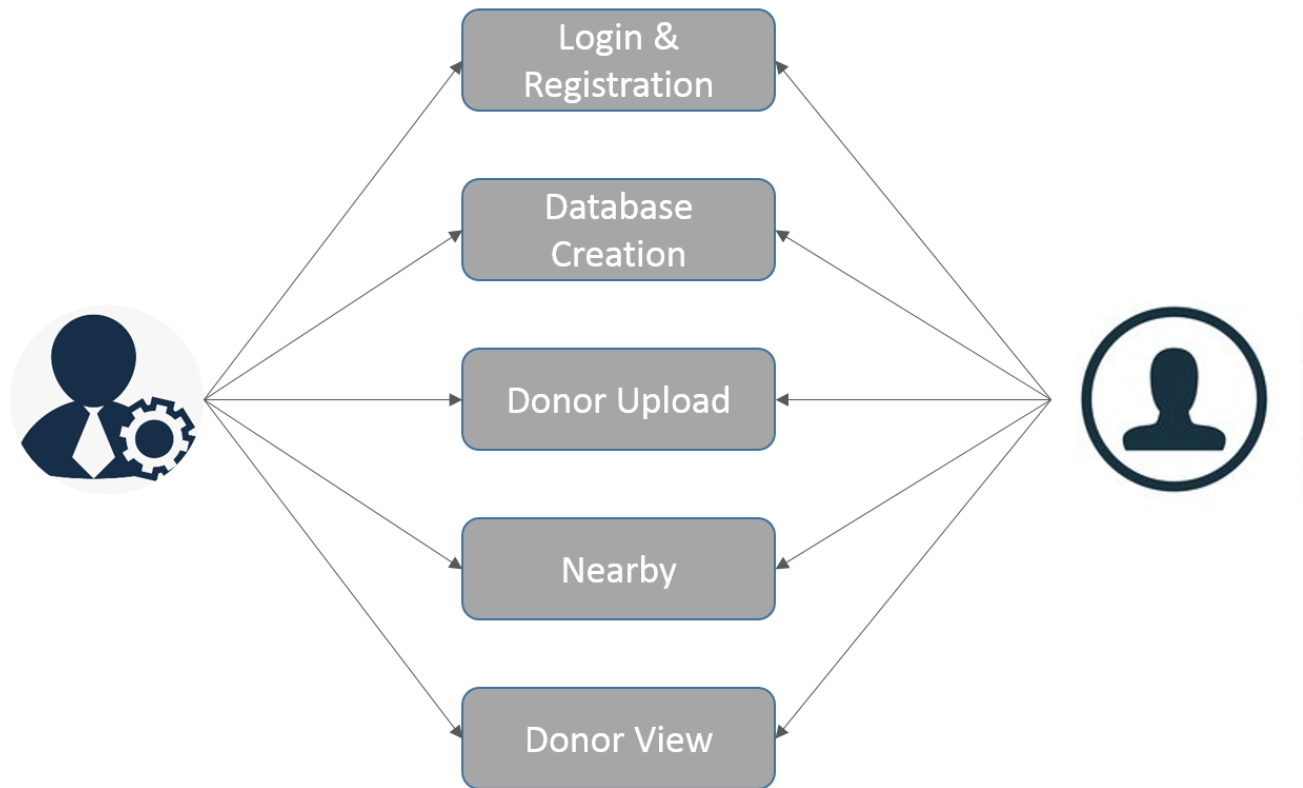
DESIGN ENGINEERING

4.1 GENERAL :

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.



4.1.2 USE CASE DIAGRAM :



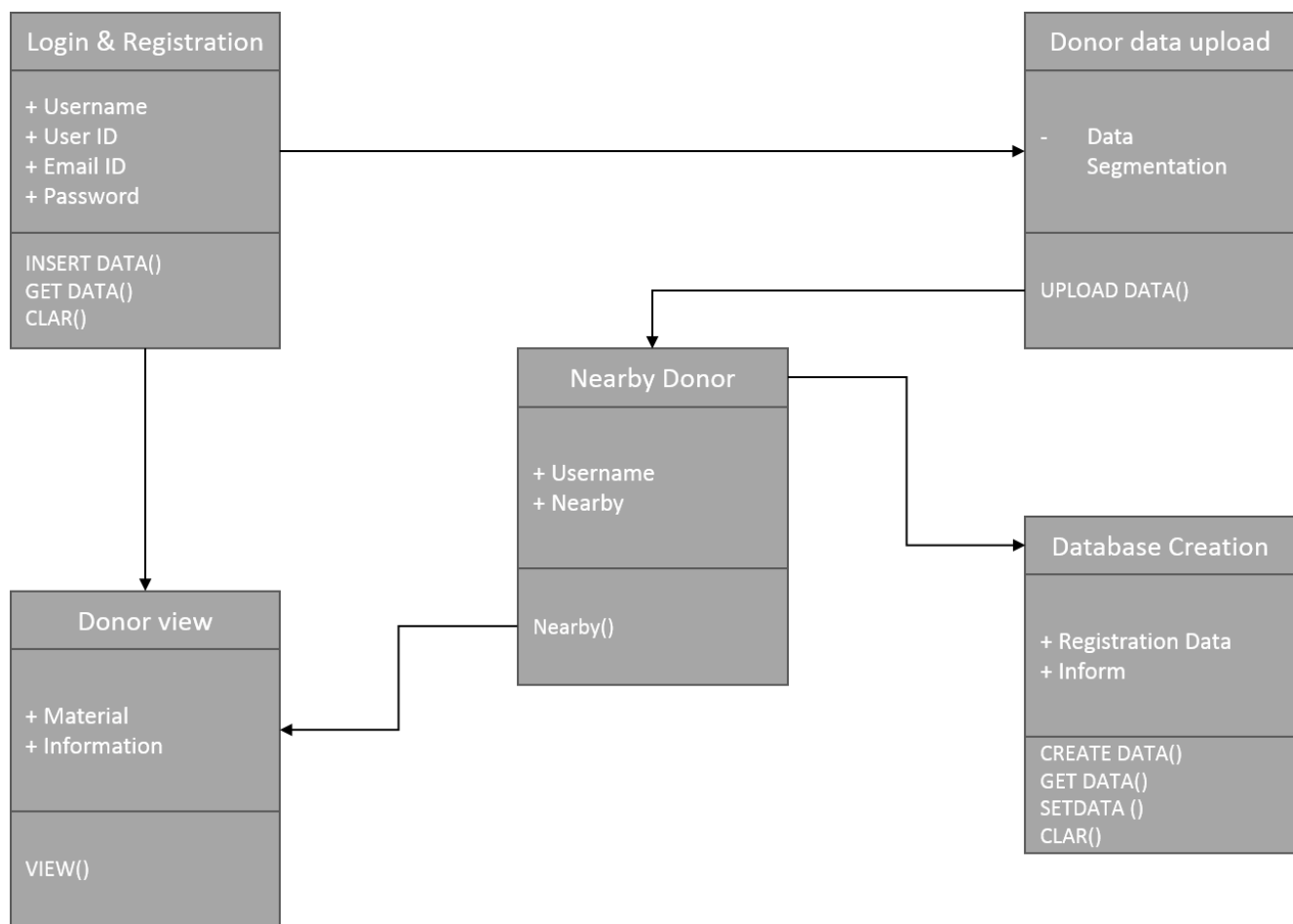
Explanation:

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications.

Functionalities to be represented as a use case, Actors and Relationships among the use cases and actors. The name of a use case is very important. So the name should

be chosen in such a way so that it can identify the functionalities performed. Give a suitable name for actors. Show relationships and dependencies clearly in the diagram. Do not try to include all types of relationships. Because the main purpose of the diagram is to identify requirements. Use note whenever required to clarify some important point.

4.2.2 CLASS DIAGRAM:



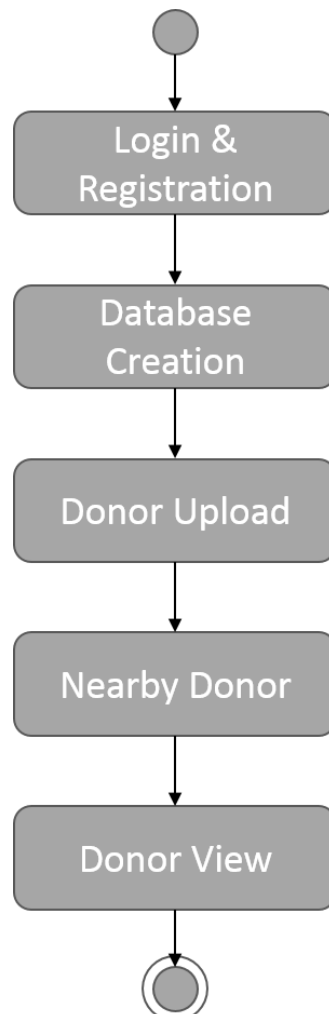
Explanation:

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should

be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified. Because of unnecessary properties will make the diagram complicated.

Use notes whenever required to describe some aspect of the diagram. Because, the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

4.2.3 ACTIVITY DIAGRAM:

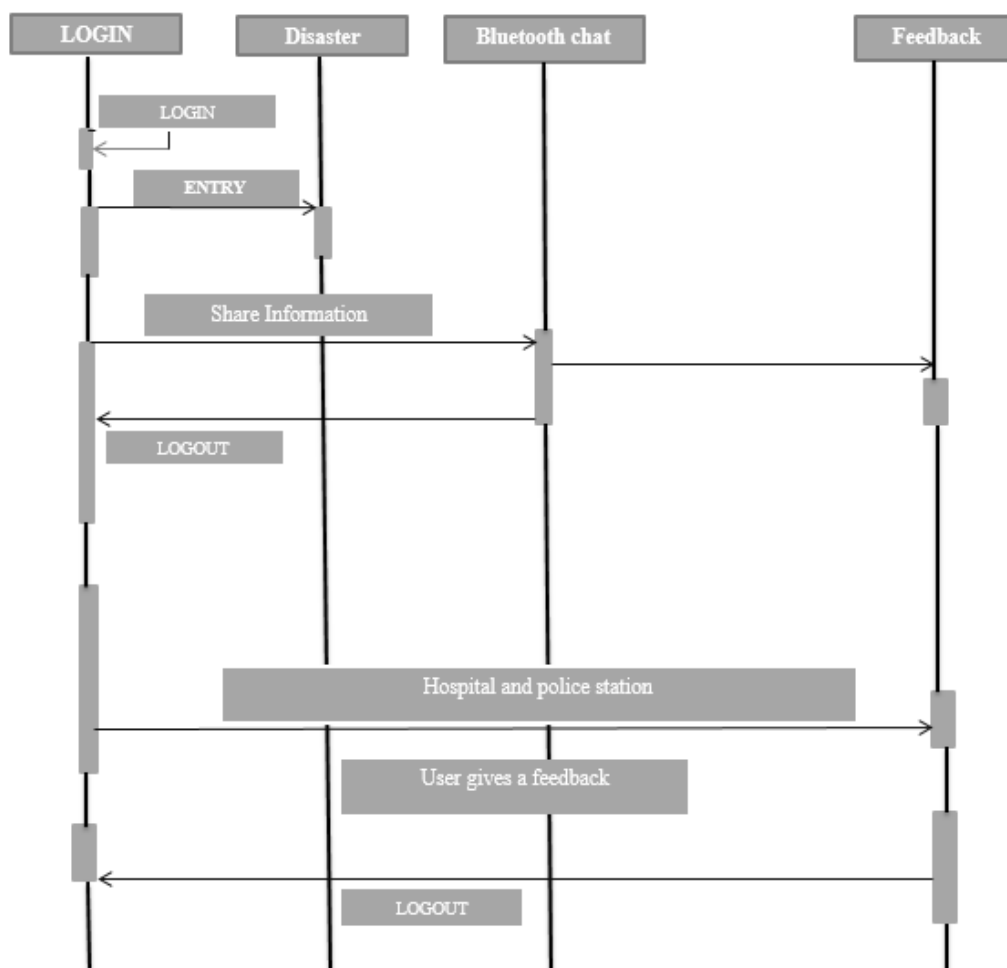


Explanation:

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single

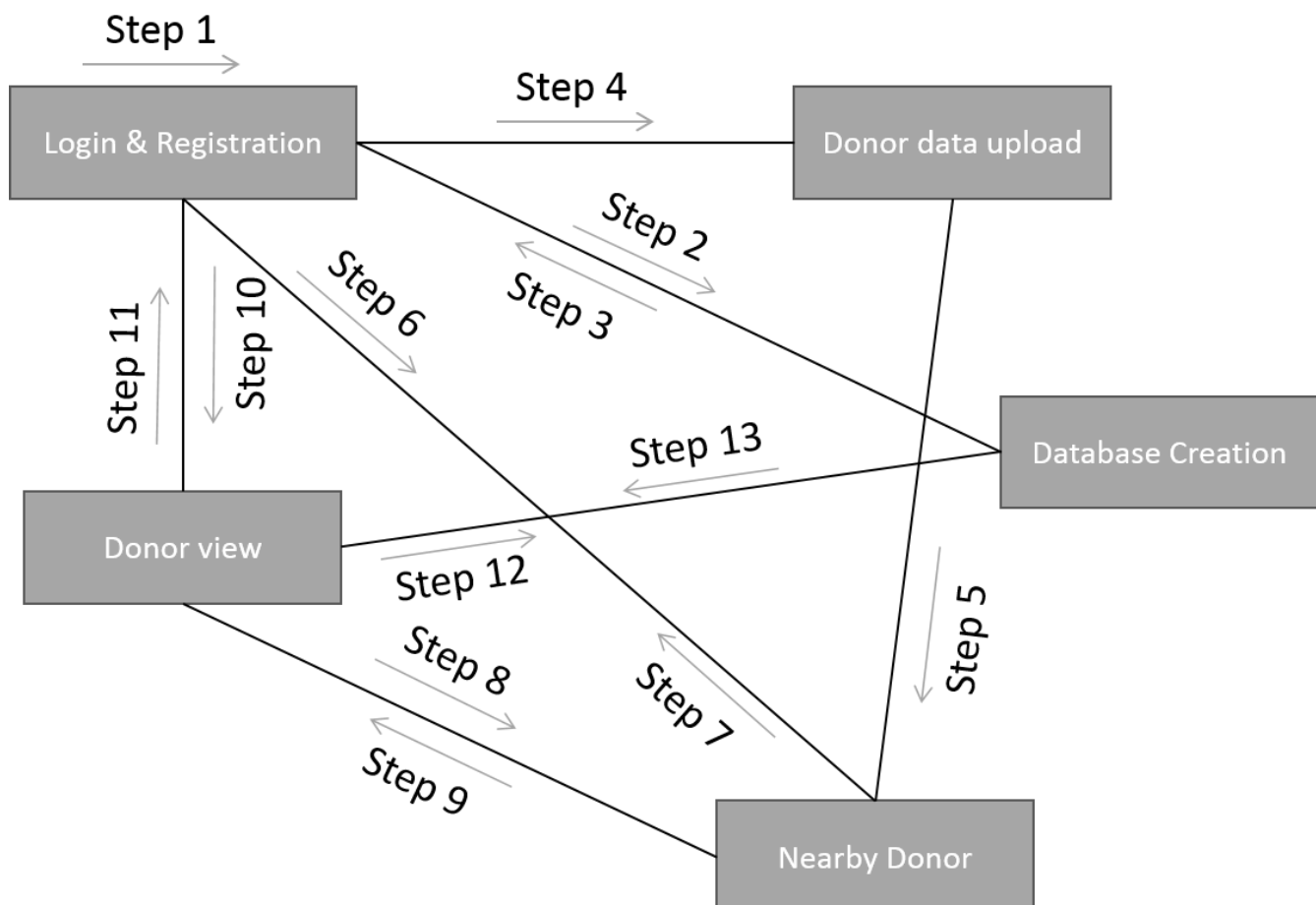
4.2.4 SEQUENCE DIAGRAM :



➤ **Explanation:**

UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Other dynamic modeling techniques include activity diagramming, communication diagramming, diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design-level models for modern business application development.

4.2.5 COMPONENT DIAGRAM :



Explanation:

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executable, libraries etc. So the purpose of this diagram is different, Component diagrams are used during the implementation phase of an application. But it is prepared well in advance to visualize the implementation details. Initially the system is designed using different UML diagrams and then when the artifacts are ready component diagrams are used to get an idea of the implementation. This diagram is very important because without it the application cannot be implemented efficiently. A well prepared component diagram is also important for other aspects like application performance, maintenance etc.

Organization can be further described as the location of the components in a system. These components are organized in a special way to meet the system requirements. As we have already discussed those components are libraries, files, executable etc. Now before implementing the application these components are to be organized. This component organization is also designed separately as a part of project execution.

4.2.6 COLLABORATION :

Explanation:

The second interaction diagram is collaboration diagram. It shows the object organization as shown below. Here in collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization. Now to choose between these two diagrams the main emphasis is given on the type of requirement. If the time sequence is important then sequence diagram is used and if organization is required then collaboration diagram is used.

CHAPTER 5

DEVELOPMENT TOOLS

5.1 GENERAL :

This chapter is about the software language and the tools used in the development of the project. The platform used here is ANDROID.

5.1.1 HISTORY OF ANDROID :

Android Inc. Founded in Palo Alto, California ,united states in October 2003 by Andy Rubin[co-founder of danger],rich miner[co-founder of wildfire communication Inc.],nick sears[once VP at T-Mobile], and Chris white[headed design and interface development at web TV] to develop.

5.2.1 INTRODUCTION OF ANDROID :

It is an open source software platform and operating system for mobile devices

Based on the Linux kernel

Developed by Google and later the Open Handset Alliance (OHA)

Allows writing managed code in the Java language

Android has its own virtual machine i.e. DVM (Dalvik Virtual Machine), which is used for executing the android application.

Google purchased the initial developer of the software, android incorporated in 2005.

OPEN HANDSET ALLIANCE:

The open handset alliance (OHA) is a business alliance of firm to develop open standard for mobile devices.

Devoted to advancing open standards for mobile devices

Develop technologies that will significantly lower the cost of developing and distributing mobile devices and services.

OPEN HANDSET ALLIANCE (OHA) INCLUDED SEVERAL COMPANIES :



5.2.2 ANDROID VERSION :

Android 1.0 (Angel Cake)	-	The first version of the open source software was released back in 2008
Android 1.1 (Battenberg)	-	In Feb 2009, version 1.1
Android 1.5 (Cupcake)	-	Launched in April 2009
Android 1.6 (Donut)	-	released in September 2009
Android 2.0 \ 2.1 (Éclair)	-	released in 26 October 2009
Android 2.2 (Froyo)	-	released in the summer of 2010
Android 2.3 (Gingerbread)	-	Gingerbread landed by the end of 2010
Android 3.0 (Honeycomb)	-	For the first time Google released a software that was totally focused on tablets. This version, released in 2011
Android 4.0 (Ice Cream Sandwich 4.0)	-	released in 2012
Android 4.1 (jelly bean4.1)	-	released in 2013
Android 4.4 (KitKat)	-	released in 2014
Android 5.0 (Lollipop)	-	released in 2014
Android 6.0 (Marshmallow)	-	released in 2015

5.2.3 ANDROID ARCHITECTURE:

The software stack is split into Four Layers:

- The application layer
- The application framework
- The libraries and runtime
- The kernel

5.2.4 SECURITY :

Android is a multi-process system, in which each application (and parts of the system) runs in its own process. Most security between applications and the system is enforced at the process level through standard Linux facilities, such as user and group IDs that are assigned to applications.

- Android is designed having multi-layer security which provides flexibility for this platform. When attackers attempt attack on device, android platform help to reduce the portability of the attack.

There are key components of android security which are described as follows:

- Design review:-when a security model is designed then it will be reviewed by the developers so that risk level will be very less while using the model.
- Code review and penetrating testing:-the goal of this code review is that in which it will be checked that how the system will become strong?
- Open source and community review:-android uses open source technologies that have significant external review such as Linux kernel.
- Incident response:-android team enables the rapid mitigation of vulnerabilities to ensure that potential risks to all android users are minimized.

5.2.5 FEATURES OF ANDROID:

- Background Wi-Fi location still runs even when
- Wi-Fi is turned off
- Developer logging and analysing enhancements
- It is optimized for mobile devices.
- It enables reuse and replacement of components.
- Java support ,media support, multi touch, video calling, multitasking ,voice based features, screen capture, camera, blue tooth, gps, compass and accelerometer,3G

5.2.6 ADVANTAGES OF ANDROID :

- The ability for anyone to customize the Google Android platform
- It gives you better notification.
- It lets you choose your hardware.
- It has better app market(1,80,000 application)
- A more mature platform
- With the support of many applications, the user can change the screen display.
- With Google chrome you can open many windows at once.
- Supports all Google services: Android operating system supports all of Google services ranging from Gmail to Google reader. All Google services can you have with one operating system, namely Android.

5.2.7 DIS-ADVANTAGES OF ANDROID :

- Android Market is less control of the manager, sometimes there are malware.
- Wasteful Batteries, This is because the OS is a lot of "process" in the background causing the battery quickly drains.
- Sometimes slow device company issued an official version of Android your own .
- Extremely inconsistence in design among apps.
- Very unstable and often hang or crash.

5.2.8 LIMITATIONS OF ANDROID:

Development requirements in

- Java
- Android SDK
- Eclipse IDE (optional)
- Bluetooth limitations:-

Android doesn't support:

- Bluetooth stereo
 - Contacts exchange
 - Modem pairing
 - Wireless keyboards
- Firefox mobile isn't coming to android because of android limitations

Apps in Android Market need to be programmed with a custom form of Java

→ Mozilla and the Fennec won't have that

- There are no split or interval times available.
- *Small memory size.*
- Continuous Internet connection is required

CHAPTER 6

SOFTWARE TESTING

6.1 GENERAL :

Cloud computing is a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources. Since an Inter Cloud is a large scale distributed and interconnected computer system, interactions among its sub components (i.e., Clouds) and among stakeholders (i.e., consumers and Cloud providers) can be complex. In an Inter Cloud, computing resources owned and administered by different Cloud providers are pooled to serve multiple consumers, and applications and data are available to and shared by a broad group of cross enterprise and cross platform users. Inter Cloud resource pooling and sharing involve 1) combining resources through cooperation among Clouds, 2) mapping and scheduling shared resources through coordination, and 3) establishing contracts between Clouds and consumers, and among Clouds through negotiation.

6.2 DEVELOPING METHODOLOGIES :

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

6.3 TYPES OF TESTINGS :

Unit testing :

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results

Functional test :

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

System Test :

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Performance Test :

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

Integration Testing :

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Acceptance Testing :

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing for Data Synchronization :

- The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
- The Route add operation is done only when there is a Route request in need
- The Status of Nodes information is done automatically in the Cache Updation process

Build the test plan :

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identify the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

CHAPTER 7

SNAPSHOTS

7.1 GENERAL :

Here in the given document we have provided various snapshots for more deep understanding of our application.

7.2 VARIOUS SNAPSHOTS AND CODING :

SPLASH PAGE :



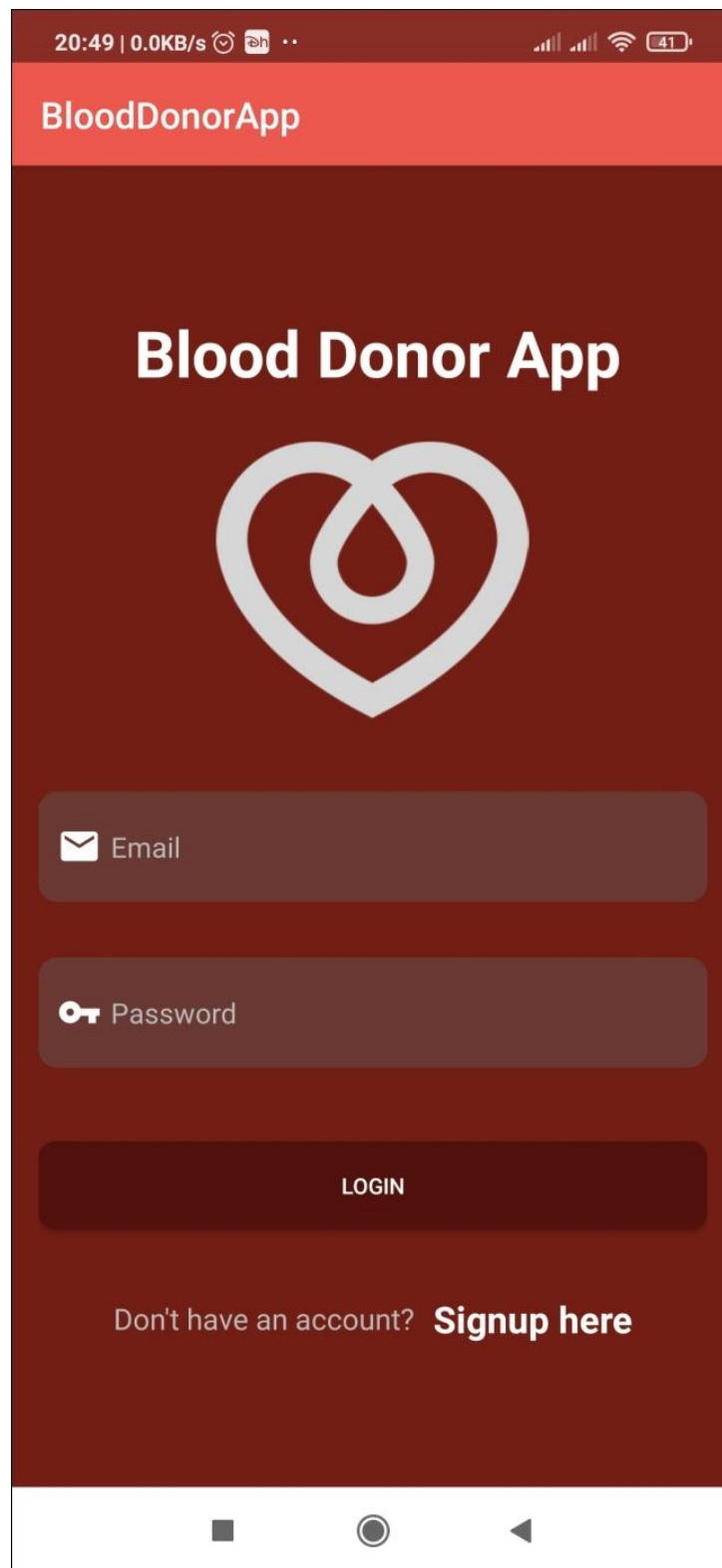
CODING :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".activities.SplashActivity"
    android:gravity="center"
    android:orientation="vertical"
    android:background="@drawable/red_button_bg">

    <LinearLayout
        android:background="@drawable/circle"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center">
        <ImageView
            android:id="@+id/load_image"
            android:layout_weight="1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/spalsh_animation"/>
    </LinearLayout>
    <TextView
        android:layout_marginTop="15dp"
        android:textColor="@android:color/white"
        android:textSize="48sp"
        android:textStyle="bold"
        android:text="@string/app_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:textSize="24sp"
        android:text="Donate Blood, Save Life"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

LOGIN PAGE :



The screenshot shows the login page of the BloodDonorApp. The status bar at the top displays the time 20:49, a data speed of 0.0KB/s, and various connectivity icons. The app's title bar is red with the text "BloodDonorApp" in white. The main background is a dark red color. In the center, the text "Blood Donor App" is written in white, with a large white icon of two interlocking hearts below it. Below the icon are two input fields: "Email" with an envelope icon and "Password" with a key icon. A dark red "LOGIN" button is positioned below these fields. At the bottom, the text "Don't have an account? Signup here" is displayed in white.

20:49 | 0.0KB/s

BloodDonorApp

Blood Donor App



Email

Password

LOGIN

Don't have an account? [Signup here](#)

CODING :

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/red_button_bg"
    tools:context=".activities.LoginActivity">

    <LinearLayout
        android:padding="10dp"

android:gravity="center"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_marginTop="20dp"
            android:layout_gravity="center"
            android:orientation="vertical"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@drawable/circle"
                android:src="@mipmap/blood_bank_icon_round" />

            <LinearLayout
                android:layout_marginTop="10dp"
                android:orientation="horizontal"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <ImageView
                    android:layout_weight="1"
                    android:src="@drawable/facebook_circle"
                    android:layout_width="30dp"
                    android:layout_height="30dp" />

                <ImageView
                    android:layout_width="30dp"
                    android:layout_height="30dp"
                    android:layout_weight="1"
                    android:src="@drawable/twitter_512" />
            </LinearLayout>

        </LinearLayout>

        <EditText
            android:textColor="@android:color/white"
            android:id="@+id/input_username"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textEmailAddress"
            android:hint="@string/prompt_email" />
    </LinearLayout>
</ScrollView>
```

```

<EditText
    android:textColor="@android:color/white"
    android:id="@+id/input_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:hint="@string/prompt_password" />

<Button
    android:textAllCaps="false"
    android:id="@+id/button_login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:textStyle="bold"
    android:background="@drawable/signin_btn"
    android:textColor="@color/primary"
    android:text="@string/action_sign_in_short" />

<Button
    android:textAllCaps="false"
    android:id="@+id/button_register"
    android:layout_width="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_height="wrap_content"
    android:padding="2dp"
    android:textColor="@android:color/white"
    android:background="@drawable/transparent_black_ten_percent"
    android:text="@string/register" />

<Button
    android:textAllCaps="false"
    android:id="@+id/button_forgot_password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="2dp"
    android:textColor="@android:color/white"
    android:background="@drawable/transparent_black_ten_percent"
    android:text="@string/forgotpass" />

```

```

</LinearLayout>

```

```

</ScrollView>

```

REGISTRATION PAGE OF DONOR AND NEEDY:

20:51 | 0.0KB/s

BloodDonorApp

Welcome

Sign Up

Username

Enter username

Number

Enter number

Address

Enter Address

Password

Password

Age

Enter Age

Sign Up

CODING :

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/red_button_bg"
    android:clipToPadding="false"
    android:fillViewport="true"
    android:scrollbarStyle="outsideOverlay"
    tools:context=".activities.ProfileActivity">

    <LinearLayout
        android:padding="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:orientation="vertical">

        <ImageView
            android:layout_gravity="center_horizontal"
            android:id="@+id/image_logo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="?attr/actionBarSize"
            android:layout_marginBottom="40dp"
```

```

        android:background="@drawable/circle"
        android:src="@mipmap/blood_bank_icon_round" />

<LinearLayout
    android:padding="3dp"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:text="Name:"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        android:layout_marginLeft="52dp"
        android:id="@+id/input_fullName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/displayName"
        android:inputType="text"
        android:textColor="@android:color/white" />
</LinearLayout>
<LinearLayout
    android:padding="3dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sex:" />

    <Spinner
        android:layout_marginLeft="67dp"
        android:id="@+id/gender"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:entries="@array/gender">

    </Spinner>
</LinearLayout>
<LinearLayout
    android:padding="3dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Blood Group:" />

    <Spinner
        android:layout_marginLeft="13dp"
        android:id="@+id/inputBloodGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"

```

```

        android:entries="@array/Blood_Group">

        </Spinner>
    </LinearLayout>
<LinearLayout
    android:padding="3dp"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Contact No:" />
    <EditText
        android:layout_marginLeft="20dp"
        android:id="@+id/inputMobile"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/mobile"
        android:inputType="text"
        android:textColor="@android:color/white" />
</LinearLayout>

<LinearLayout
    android:padding="3dp"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Address:" />
    <EditText
        android:layout_marginLeft="40dp"
        android:id="@+id/inputAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/userAddress"
        android:inputType="text"
        android:textColor="@android:color/white" />
</LinearLayout>

    <LinearLayout
        android:padding="3dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Division:" />

        <Spinner
            android:layout_marginLeft="42dp"
            android:id="@+id/inputDivision"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:entries="@array/division_list">

```

```

        </Spinner>
    </LinearLayout>

    <EditText
        android:id="@+id/input_userEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/prompt_email"
        android:inputType="textEmailAddress"
        android:textColor="@android:color/white" />

    <EditText
        android:id="@+id/input_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/prompt_password"
        android:inputType="textPassword"
        android:textColor="@android:color/white" />

    <EditText
        android:id="@+id/input_password_confirm"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/prompt_confirm_password"
        android:inputType="textPassword"
        android:textColor="@android:color/white" />

    <CheckBox
        android:id="@+id/checkbox"
        android:text="Mark this to be a donor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:layout_gravity="center_horizontal"
        android:id="@+id/button_register"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:layout_marginBottom="40dp"
        android:background="@drawable/signin_btn"
        android:text="@string/signup"
        android:textAllCaps="false"
        android:textColor="@color/colorPrimary" />

</LinearLayout>

</ScrollView>

```

PASSWORD RESET PAGE :



CODING :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/red_button_bg"
    tools:context=".activities.RestorePassword">

    <LinearLayout
        android:layout_centerInParent="true"
        android:gravity="center"
        android:orientation="vertical"
        android:layout_width="300dp"
        android:layout_height="wrap_content">

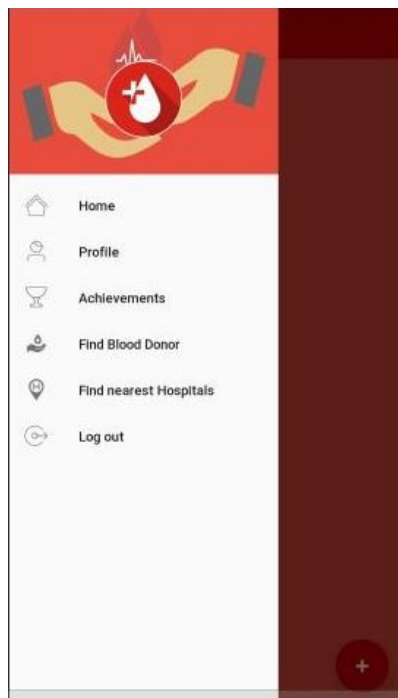
        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <EditText
                android:padding="5dp"
                android:textColor="@android:color/white"
                android:id="@+id/resetUsingEmail"
                android:hint="Enter your registered email"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
        </android.support.design.widget.TextInputLayout>
```

```

        <Button
            android:layout_marginTop="20dp"
            android:id="@+id/resetPassbtn"
            android:textAllCaps="false"
            android:background="@drawable/signin_btn"
            android:text="@string/reset"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
</RelativeLayout>

```

DASHBOARD PAGE :



CODING :

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    class="com.google.android.gms.maps.SupportMapFragment"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context=".activities.Dashboard"
    tools:openDrawer="start">

    <include
        layout="@layout/app_bar_dashboard"

```

```

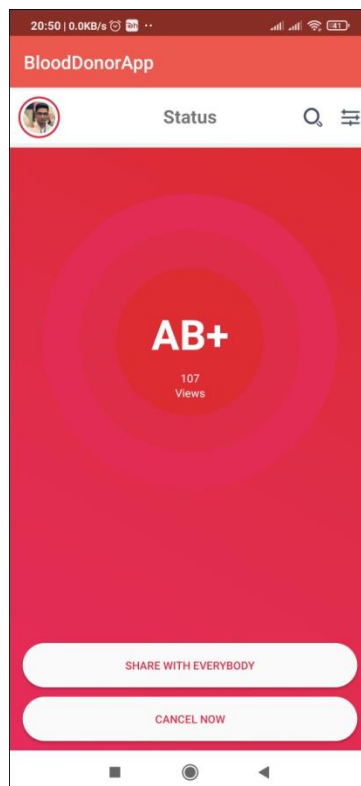
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header_dashboard"
    app:menu="@menu/activity_dashboard_drawer" />

</android.support.v4.widget.DrawerLayout>

```

POSTING AND REQUESTING PAGE :



CODING :

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/red_button_bg"
    tools:context=".activities.PostActivity">

```

```

<LinearLayout
    android:padding="20dp"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_marginTop="2dp"
        android:layout_marginBottom="2dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:padding="5dp"
            android:text="Contact:"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <EditText
            android:id="@+id/getMobile"
            android:padding="5dp"
            android:inputType="phone"
            android:hint="@string/mobile"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </LinearLayout>

    <LinearLayout
        android:layout_marginTop="2dp"
        android:layout_marginBottom="2dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:padding="5dp"
            android:text="Location:"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <EditText
            android:id="@+id/getLocation"
            android:padding="5dp"
            android:inputType="text"
            android:hint="Enter location of place"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </LinearLayout>

    <LinearLayout
        android:layout_marginTop="2dp"
        android:layout_marginBottom="2dp"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:padding="5dp"
            android:text="Blood Group:"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <Spinner
            android:id="@+id/SpinnerBlood"

```

```

        android:entries="@array/Blood_Group"
        android:padding="5dp"
        android:inputType="text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
<LinearLayout
    android:layout_marginTop="2dp"
    android:layout_marginBottom="2dp"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:padding="5dp"
        android:text="Choose Division:"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Spinner
        android:id="@+id/SpinnerDivision"
        android:entries="@array/division_list"
        android:padding="5dp"
        android:inputType="text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>

<Button
    android:layout_marginTop="40dp"
    android:layout_gravity="center"
    android:gravity="center"
    android:id="@+id/postbtn"
    android:text="Post"
    android:backgroundTint="@color/primary_dark"
    android:background="@drawable/signin_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

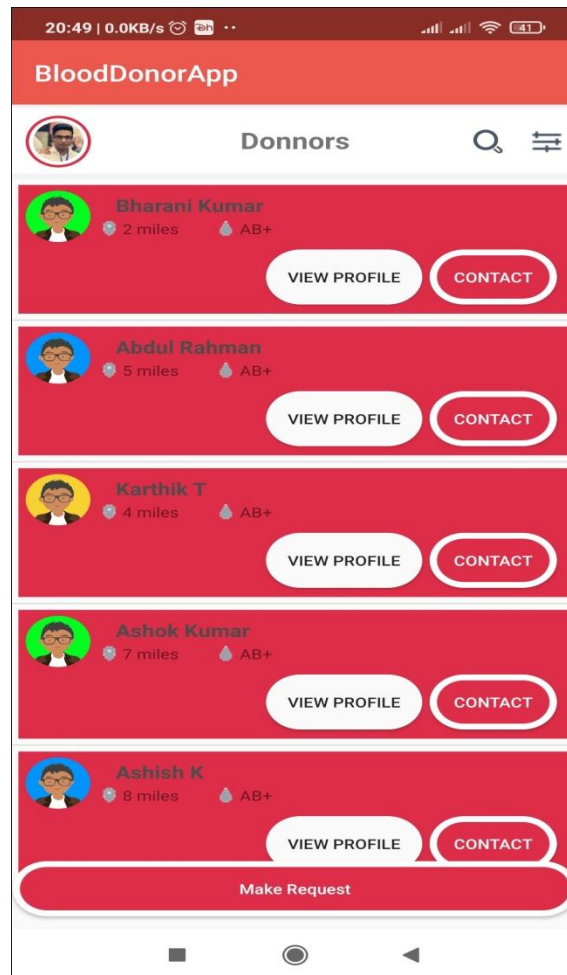
```

```

</LinearLayout>

```

DONORS LIST PAGE :



CHAPTER 8

APPLICATION

8.1 GENERAL :

- The project involves in a main activity of saving humans life
- A probabilistic model is proposed to evaluate the reliability of the sensing data as the users' service qualities.
- It gives more accurate and speeding services

8.2 FUTURE ENHANCEMENTS :

The future involves of adding google maps and adding story board line which is nothing but a donation bar which allows the user to post the medical problems not only the blood also other medical issue where they can request for the money .The requested money is then deposited to the respected person accordingly . This story line enables the cure for the solutions e a s it passes person to person easily.

CHAPTER 9

9.1`REFERENCES :

- [1] Brooks, C., "Introductory Econometrics for Finance," Cambridge University Press, Cambridge, UK. pp. 289, 2002.
- [2] Carney, j., Cunningham, p., "Tuning diversity in bagged ensembles," International Journal of Neural Systems, pp. 267–280, 2000.
- [3] Celik, A.E., Karatepe, Y., "Evaluating and forecasting banking crises through neural network models: an application for Turkish banking sector," Expert Systems with Applications, pp.809–815, 2007.
- [4] Dimitri Pissarenko, "Neural Networks For Financial Time Series Prediction Overview Over Recent Research," 2001-2002.
- [5] Gronholdt, L., Martensen, A., "Analysing customer satisfaction data: A comparison of regression and artificial neural networks," International Journal of Market Research, pp. 121–130, 2005.
- [6] Hansen, J.V., McDonald, J.B., Nelson, R.D., "Time series prediction with geneticalgorithm designed neural networks: An empirical comparison with modern statistical models," Computational Intelligence pp. 171–184, 2002.
- [7] Iebling Kaastra, a., Milton Boyd, b., "Designing a neural network for forecasting financial and economic time series," Neurocomputing, pp. 215-236, 1996.
- [8] Kanas, A., "Neural network linear forecasts for stock returns," International Journal of Finance and Economics, pp. 245–254, 2001.
- [9] Kumar, K., Bhattacharya, S., "Artificial neural network vs. linear discriminant analysis in credit ratings forecast: A comparative study of prediction performances," Review of Accounting & Finance, pp. 216, 2006.

[10] Lai K.K., Yu .L., Wang S.Y, Zhou C.X., “Neural-network-based metamodeling for financial time series forecasting, in: Proceedings of the 9th Joint Conference on Information Sciences,” Atlantis Press, Paris, pp. 172–175, 2006.

J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, pp.68-73, 1992.

