

Connect to Blood

Project Report

Submitted in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

Aman Rai (1900950100011)

Namita Raj (1900950100045)

Preeti Sharma (1900950100058)

Ria Sirohi (1900950100065)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MGM's College of Engineering & Technology, Noida

May, 2023

TABLE OF CONTENTS

	PAGE
DECLARATION	i
CERTIFICATE	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
LIST OF FIGURES	vi
LIST OF ABBREVIATIONS	viii

CHAPTER 1 INTRODUCTION AND BACKGROUND OF THE PROJECT

1.1. Literature review.....	1
1.2. Problem definition.....	4
1.3. Brief introduction of the project.....	6
1.4. Technologies used.....	7
1.4.1 Introduction to Xamarin.....	7
1.4.2 Introduction to C-Sharp.....	7
1.4.3 Introduction to SQL.....	8
1.5. Proposed modules.....	8
1.6. Software & Hardware requirements.....	9

CHAPTER 2 SYSTEMS ANALYSIS AND SPECIFICATION

2.1. A functional model.....	11
2.1.1. Data Flow Diagram (DFD).....	11

2.2. A data model.....	15
2.2.1. ER diagram.....	15
2.3. A process-flow model.....	17
2.3.1. Activity diagram.....	17
2.4. A behavioral model.....	18
2.4.1. Sequence diagram.....	18
2.4.2. Class diagram.....	19
2.5. System Design.....	22
2.5.1 Technical feasibility.....	22
2.5.2 Operational feasibility.....	23
2.5.3 Economic feasibility.....	23

CHAPTER 3 (MODULE IMPLEMENTATION & SYSTEM INTEGRATION)

3.1. Module Implementation.....	24
3.2. System Integration.....	32

CHAPTER 4 (TESTING AND EVALUATION).....

4.1 Testing	33
-------------------	----

CHAPTER 5 (TASK TIMELINE AND SCHEDULE OF ACTIVITIES)

5.1 Analysis.....	42
5.2 Design.....	42
5.3 Build.....	42
5.4 Testing.....	43
5.5 Maintenance.....	43

CHAPTER 6 (PROJECT MANAGEMENT)	
6.1 Major risks and contingency plans	44
6.2 Principle learning outcomes	44
CONCLUSION AND FUTURE SCOPE.....	45
REFERENCES.....	46

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name: Aman Rai

Roll No.: 1900950100011

Date:

Signature:

Name: Namita Raj

Roll No.: 1900950100045

Date:

Signature:

Name: Preeti Sharma

Roll No.: 1900950100058

Date:

Signature:

Name: Ria Sirohi

Roll No.: 1900950100065

Date:

CERTIFICATE

This is to certify that Project Report entitled “Connect to Blood” which is submitted by Aman Rai (1900950100011), Namita Raj (1900995000045), Preeti Sharma (190090100058), Ria Sirohi (1900950100065) in partial fulfillment of the requirement for the award of degree B. Tech. in the Department of Computer Science and Engineering of MGM’s College of Engineering and Technology which is affiliated by AKTU Lucknow, is a record of the candidate's own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date:

Supervisor Signature:

Name of Supervisor: Ms. Karamjeet Kaur

Designation: H.O.D

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Ms. Karamjeet Kaur Department of Computer Science & Engineering, MGM CoET, Noida for her constant support and guidance throughout the course of our work. Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only her cognizant efforts that our endeavour have seen the light of the day. We also take the opportunity to acknowledge the contribution of Ms. Karamjeet Kaur Head, Department of Computer Science & Engineering, MGM CoET, Noida for her full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project.

Signature:

Name of Student: Aman Rai

Roll No.: 1900950100011

Signature:

Name of Student: Namita Raj

Roll No.: 1900950100045

Signature:

Name of Student: Preeti Sharma

Roll No.: 1900950100058

Signature:

Name of Student: Ria Sirohi

Roll No.: 1900950100065

Date:

ABSTRACT

The purpose of this study was to develop a blood management information system to assist in the management of blood donor records and ease/or control the distribution of blood in various parts of the country based on the hospital's or Recipient's demands thoroughly. Without quick and timely access to donor can lead to death of the patient .If someone needs the blood, then either the person has to go to the blood bank nearby him/her or else have to buy the blood from the hospital but in case, if both the places don't have the blood of the required group than finding blood of that group might be a tough task at the required time. Also there is a lack of proper documentation about blood donors and their medical history. This may lead to blood bag contamination and may affect the blood transfusion safety. The lack of proper documentation may endanger a patient's health due to the possibility of having contaminated blood bags. Contamination happens when there is an incomplete donor's medical history record and the blood bag's self life is not monitored properly. Entering the details about the blood groups, members, addresses etc. and tracking the database is complicated when the details are maintained manually. This makes the maintenance of schedule erroneous. At the blood house unit, the staff and nurses only are informed about the blood donation schedule for each month on the whiteboard at the blood house. The public did not have knowledge about blood donation. Since most blood banks are still in a paper-based system, various disadvantages are experienced by various stakeholders, which endanger the lives of patients and deter the healthcare system.

Blood donation is a humanitarian, altruist, noble, and solidary act supported by all health organizations in the world. Blood transfusion saves lives and improves health. Providing safe and adequate blood should be an integral part of every country's national health care policy and infrastructure. With the advancement of information and communication technologies, new technological tools such as social networks and applications have been used for the blood donation process on a global scale. The main motive of collecting blood is to supply patients who need blood for surgery or for treatment of any disease.

To fulfill the requirement, we together decided to deliver a platform that will help and give hope to needy ones and their families. This project aims at maintaining all the information pertaining

to blood donors. The blood management information system will offer functionalities to quick access to donor records collected from various parts of the country. It will enable monitoring of the results and performance of the blood donation activity such that relevant and measurable objectives of the organization can be checked. It will provide to management timely, confidential and secure medical reports that facilitates planning and decision making and hence improved medical service delivery. The reports that will be generated by the system will give answers to most of the challenges management faces as far as blood donor records are concerned. This study aims to determine how the use of online bank management systems enhances blood transfusion safety.

These problems can be dealt with by automating the existing manual blood bank management system. In our project we are using Android/iOS development so that users can reach nearby hospitals, blood banks, and volunteer donors in an efficient way. This project will be developed by four perspectives i.e., hospital, blood bank, volunteer donors and patients. A high-end, efficient, highly available and scalable system has to be developed to bridge the gap between the donors and the recipients and to reduce the efforts required to search for blood donors. This application will help to select the nearby hospitals, blood banks, donors online instantly by tracing its location using GPS and also reducing the time to a greater extent while searching for the blood remotely. A digital donor card will be provided to donors and ambulance services will be arranged in case of emergency. Health insurance along with health-related surveys is also a part of our application. Donors can know their blood screening result for each time they donate their blood. With having this function in the system, the donor can monitor his or her health condition. This will make the donor become aware of their health condition.

List of Figures

SNo.	Figure Number	Figure Name	Page Number
1.	2.1.a	DFD Level 0	11
2.	2.1.b	DFD Level 1	13
3.	2.1.c	DFD Level 2	14
4.	2.2.a	ER Diagram	16
5.	2.3.a	Activity Diagram	17
6.	2.4.a	Sequence Diagram	19
7.	2.4.b	Class Diagram	21
8.	3.1.a	Splash Screen	24
9.	3.1.b	User Login Screen	25
10.	3.1.c	User Signup Screen	26
11.	3.1.d	OTP Screen	26
11.	3.1.e	Main Screen	27
12.	3.1.f	Menu Screen	28
13.	3.1.g	Profile Screen	28
14.	3.1.h	Insurance Screen	29
15.	3.1.j	Ambulance Service Screen	29
16.	3.1.k	Privacy Policy Screen	30

17.	3.1.1	Donor Screen	31
18.	3.1.m	Receiver Screen	32
19.	4.1.a	API Testing Image 1	35
20.	4.1.b	API Testing Image 2	36
21.	4.1.c	API Testing Image 3	36
22.	4.1.d	API Testing Image 4	37
23.	4.1.e	Manual Testing Image Setup	38
24.	4.1.f	Errors while Testing	38
25.	4.1.g	Manual Testing 1	39
26.	4.1.h	Manual Testing 2	39
27.	4.1.i	Manual Testing 3	40
28.	4.1.j	Manual Testing 4	40
29.	4.1.k	Manual Testing 5	41

List of Abbreviations

GPS	Global Positioning System
ASP	Active Server Pages
PHP	Hypertext Preprocessor
SQL	Structured Query Language
OBBMS	Online Blood Bank Management System
API	Application Programming Interface
CLI	Command Line Interface
ECMA	European Computer Manufacturers Association
RDBMS	Relational DataBase Management System
RDSMS	Relational Data Stream Management System
ISAM	Indexed Sequential Access Method
VSAM	Virtual Storage Access Method
DDL	Data Definition Language
DQL	Data Query Language
DCL	Data Control Language
DML	Data Manipulation Language
.NET MAUI	Multi-platform App UI
XAML	Extensible Application Markup Language
IDE	Integrated Development Environment
GUI	Graphical User Interface

CHAPTER 1

INTRODUCTION AND BACKGROUND OF PROJECT

INTRODUCTION

This project aims at maintaining all the information pertaining to blood donors. The intent is to automate the complete operation of the blood bank. In our project we are using Android development so that users can reach nearby hospitals, blood banks, and volunteer donors in an efficient way. This project is developed by four perspectives i.e., hospital, blood bank, volunteer donors and patient. The application helps to select the nearby hospitals, blood banks, donors online instantly by tracing its location using GPS and also reduces the time to a greater extent while searching for the blood remotely. A digital donor card is provided to donors and ambulance services are arranged in case of emergency. Health insurance along with health-related surveys is also a part of our application. It enables monitoring of the results and performance of the blood donation activity such that relevant and measurable objectives of the organization can be checked. It provides to management timely, confidential and secure medical reports that facilitates planning and decision making and hence improved medical service delivery. The reports generated by the system give answers to most of the challenges management faces as far as blood donor records are concerned. Blood donation is a humanitarian, altruist, noble, and solidary act supported by all health organizations in the world. Blood transfusion saves lives and improves health. Providing safe and adequate blood should be an integral part of every country's national health care policy and infrastructure. With the advancement of information and communication technologies, new technological tools such as social networks and applications have been used for the blood donation process on a global scale. The main motive of collecting blood is to supply patients who need blood for surgery or for treatment of any disease.

1.1 LITERATURE REVIEW

The Project provides the facility for the donor to register by him or herself as a blood donor. Only citizens in India can register to the system. It also provides a feature where a person or hospital can request the blood bag or blood stock from the Blood Bank. It gives a charge to the

person or patient that is in need of blood. However, the money that is collected is not for the profit but to recover the expenses incurred in recruiting and educating donors. This is also to ensure that the blood transfusion is as safe as possible. It also informs the donor and the public where and when is their next blood donation campaign. Also, it provides facilities for the donor and the patient so they know how many times that they have donated their blood. Also, for the donor, they can know their blood screening result for each time they donate their blood. With having this function in the system, the donor can monitor his or her health condition. This will make the donor become aware of their health condition. In addition, it seems that there is a lack of proper documentation about blood donors and their medical history. This may lead to blood bag contamination and may affect the blood transfusion safety. Generally, this study aims to determine how the use of online bank management systems enhance blood transfusion safety. The lack of proper documentation may endanger a patient's health due to the possibility of having contaminated blood bags. Contamination happens when there is an incomplete donors' medical history record and the blood bags' shelf life is not monitored properly. Hence, a blood bank management system might be needed to address these issues and problems encountered to ensure blood transfusion safe. In this study, the researchers learnt the importance of implementing a web-based blood bank management system in handling records for blood donors and blood donation activities to ensure accurate and readily available information for blood transfusion services. Indeed, the impact of using Information Technology on hospitals provides better healthcare services for the public. Likewise, the researchers learnt that there are programming languages suitable for web-based applications such as ASP.NET, PhP, to name a few.

Blood is a constantly circulating uid providing the body with nutrition, oxygen, and waste[1] removal. Blood is mostly liquid, with numerous cells and proteins suspended in it. Making blood thicker than pure water. The average person has about 5 liters of blood. It forms an integral part in mankind's life, because one the most important function it performs is transportation of needed oxygen to various parts and also helps in removing carbon dioxide from the human body. This knowledge of the importance of human blood plays a very important role in healthcare centers in reviving many patients such as accidental, gun shot or in operation theater. The Department of Transfusion medicine plays a very important role in separating the blood components such as plasma, platelets, and packed red cells. Each and every component present in a blood plays a

specific role. The patients with less Hemoglobin level need to be transfused with packed red cells, and patients with low platelet count need to be transfused with Platelets, and since plasma contains coagulation factors can be very useful during operation and many more patient's needs. The Department of Transfusion medicine plays a very important role in meeting all the blood requirements for all the healthcare systems. To make things more accurate and manage blood requirements more systematically blood bank management systems are introduced. These systems store data of blood donors and as well as blood stored in different blood banks and hospitals. So that whenever blood is required, they can get access to blood easily.

Blood donation and transfusion has been an ever-serious issue and the shortage of blood[2] throughout the world has caused many people to lose their life. The lack of a centralized system for blood donation is majorly responsible for those losses. Now in the era of online and digital processes, the conventional methods of collecting blood are absolute. An automated system is required to manage the centers and to showcase the information to the interested parties. We have developed a website that single handedly solves all these issues related to blood donation and reception. We have designed a SQLite database as an integral part of the integrated framework to store historical blood donation data in a centralized database for analytical processing. The proposed system would enable people to register as a donor to make themselves available whenever in need of their blood type. We have introduced a search tab to search available people ready to donate. In our proposed system in the donor registration, health-related details would be updated in the blood management system database for all to see.

Blood Donation is a tedious process where it involves a lot of manual work with the expense[3] of time. Requesting blood is a highly critical task for the patients who are victims of several kinds of accidents. Their compatible blood types should be made readily available for their further treatment. Blood Donors should also be intimated appropriately for faster blood availability for a specific request of blood. Our project focuses on this blood donation process where Emergency situations, such as accidents, create immediate and critical needs for specific blood types. The blood bank won't give blood for free unless you provide them with another blood. Due to lack of communication between blood receivers and donors, people face problems in the blood transfusion. The donor has to forward the blood request with ease and his/ her privacy should be maintained.

An online Blood Bank Management System can be utilized in any center, clinic, lab, or crisis[4] circumstance which requires blood units for endurance. The Online Blood Bank site is a true exertion of facilitating all cycles rotating around getting and giving blood. The site empowers the client to get to without any problem data with respect to the accessibility of blood classifications in different blood donation centers. Our framework can be utilized to track down the required measures of blood in crisis circumstances from either blood donation centers or even blood benefactors. The goals of proposing such a framework are to annul the frenzy made during a crisis due to inaccessibility of blood. The objective of this undertaking is to give individuals a solitary answer for all the blood-giving and getting issues all at one place in a solitary snap.

The Blood Bank Management System is a web based application that is related to[5] administrative and inventory management in a blood bank which maintains, organizes, retrieves, and analyses data. We developed an application to make the procedure of requesting blood and increase the availability. When a specific blood group is required urgently, you can utilize the app to contact only those who have that blood group. This system is made up of various modules that keep track of blood and blood requests, hospitals. Our web application will attract a huge number of blood donors. Cloud-based systems can be useful in emergency blood supply since they allow for central and fast access to donor data and location from nearly any place and device. Because practically everyone has a cell phone, it allows for real time location tracking and communication. As a result, the 'Online Blood Bank' could be a lifesaver for blood donors.

1.2. PROBLEM DEFINITION

1.2.1 PROBLEM

The percentage of people donating blood is increasing day by day due to awareness to donate blood for those needed. The blood received has to be managed thoroughly so that there will be no negative effect to the blood receiver once they receive blood.

1.2.2 LIMITATIONS OF THE MANUAL SYSTEM

1. It is time consuming.
2. It leads to error prone results.
3. Complication during Blood Donation.
4. It lacks data security.

5. Retrieval of data takes a lot of time.
6. Percentage of accuracy is less.
7. Reports take time to produce.

1.2.3 DETAILS OF THE PROBLEM DEFINITION

Entering the details about the blood groups, members, addresses etc. And tracking the database is complicated when the details are maintained manually. This makes the maintenance of schedule erroneous. The blood donation event schedule is normally advertised to the public so that they are aware of the blood donation campaign period. At the blood house unit, the staff and nurses only are informed about the blood donation schedule for each month on the whiteboard at the blood house. So, they are using manual ways in informing the schedule. The problem arises when the space provided is not enough. The public did not have knowledge about blood donation. Hence, the public are not getting any detailed information about blood donation unless they go to the blood donation house. One of the factors of the public being afraid to donate their blood is they believe in myths. The myths that they always believe are, if they donate their blood, they will become fat and if they donate their blood, their blood will become less in total and they will become pale. Since most blood banks are still in a paper-based system, various disadvantages are experienced by various stakeholders, which endanger the lives of patients and deter the healthcare system. As to overcome this problem we aimed to design, develop, and implement an online blood bank management system (OBBMS}. There is a dire need of synchronization between the blood donors and hospitals and the blood banks. This improper management of blood leads to wastage of the available blood inventory. Improper communication and synchronization between the blood banks and hospitals leads to wastage of the blood available.

1.2.4 OBJECTIVES

1. Maintaining information pertaining to blood donors.
2. Maintaining blood groups available in each blood bank.
3. Make the process of obtaining blood from blood banks hassle free and corruption free.
4. Automate complete operation of blood banks.

5. Suits all types of blood banks in future.
6. Making blood banks systematic and manageable.

1.2.5 SOLUTION TO PROBLEM STATEMENT

These problems can be dealt with by automating the existing manual blood bank management system. A high-end, efficient, highly available and scalable system has to be developed to bridge the gap between the donors and the recipients and to reduce the efforts required to search for blood donors. Therefore, with the use of an online blood bank management system, the blood transfusion process is safe and secured. Threats on improper blood donor documentation, or misplaced records will be totally eradicated. Also, processes involving recording about blood donors, blood bag collection, storage, and inventory will be systematized and organized, hence, improving the healthcare management for blood banks.

1.3 BRIEF INTRODUCTION OF THE PROJECT

1.3.1 BLOOD BANK MANAGEMENT SYSTEM APPLICATION

Blood Bank Management System (BBMS) is an application that can assist the information of blood bags during its handling in the blood bank. With this system, the user of this system can key in the result of a blood test that has been conducted to each of the blood bags received by the blood bank. The result of the test will indicate whether the blood bag can be delivered to the patient or not.

From this system, there are several types of reports that can be generated such as blood stock report, donor's gender report and the total of blood donation according to months and year. The system also can give the information to the donor about blood analysis test results for each time the donor makes a contribution. Hence, it will make the blood bank more systematic and manageable. This system will help the needy person, access to all the donors of the blood of the particular group, nearby to their place, in a minute of time. The details will include the name of the donor, his/her blood group, contact number and address. This app is effective in raising blood donation requests and dissemination, reducing the gap between blood donors and the people in need.

1.4 TECHNOLOGY

We are using Xamarin as frontend and C-Sharp for backend in this project. The database is prepared on SQL server connected through API. The framework that is used for this is .NET framework. Visual Studio is used in order to complete the project.

1.4.1 XAMARIN

Xamarin is a Microsoft-owned San Francisco based software company founded in May 2011 by the engineers that created Mono, Xamarin.Android (formerly Mono for Android) and Xamarin.iOS (formerly MonoTouch), which are cross-platform implementations of the Common Language Infrastructure (CLI) and Common Language Specifications (often called Microsoft .NET). With a C# shared codebase, developers can use Xamarin tools to write native Android, iOS, and Windows apps with native user interfaces and share code across multiple platforms, including Windows, macOS, and Linux. According to Xamarin, over 1.4 million developers were using Xamarin's products in 120 countries around the world as of April 2017. On February 24, 2016, Microsoft announced it had signed a definitive agreement to acquire Xamarin.

1.4.2 C-SHARP

C# (pronounced C sharp) is a general-purpose high-level programming language supporting multiple paradigms. C# encompasses static typing, strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. The C# programming language was designed by Anders Hejlsberg from Microsoft in 2000 and was later approved as an international standard by Ecma (ECMA-334) in 2002 and ISO/IEC (ISO/IEC 23270) in 2003. Microsoft introduced C# along with .NET Framework and Visual Studio, both of which were closed-source. At the time, Microsoft had no open-source products. Four years later, in 2004, a free and open-source project called Mono began, providing a cross-platform compiler and runtime environment for the C# programming language. A decade later, Microsoft released Visual Studio Code (code editor), Roslyn(compiler), and the unified .NET platform (software framework), all of which support C# and are free, open-source, and cross-platform. Mono also joined Microsoft but was not merged into .NET. As of November 2022, the most recent stable version of the language is C# 11.0, which was released in 2022 in .NET 7.0.

1.4.3 SQL

Structured Query Language, abbreviated as SQL, sometimes "sequel" for historical reasons), is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables. SQL offers two main advantages over older read-write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify *how* to reach a record, e.g. with or without an index. Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements, which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update, and delete), data definition (schema creation and modification), and data access control. Although SQL is essentially a declarative language (4GL), it also includes procedural elements.

1.5 PROPOSED MODULES

This system is used for maintaining whole information about Blood Management.

The major modules in this project are:

Admin:

This module focuses on both donors & acceptors. Each member in a donor & acceptor is given a user id and password, which identifies him uniquely. The member is given a login form. He enters the login details user id and password.

Donor:

Each member in a Donor is given a user id and password, which identifies him uniquely. The member is given a login form. He enters the login details user id and password.

Acceptor:

In this module all the information related to Recipient is stored.

Keeping the problem definition in mind the proposed system evolves which is user friendly, easy to update with the new features, data is maintained and reports generated will be more useful for management to take quick business .

Health Report-card:

A report card of users based on their health status is generated.

Once the blood donated by the user is tested, health status is updated on the app.

Survey:

Collecting data from users with the help of surveys to gain insights.

Insurance:

Health insurance will be available for needy people which will include the funds other than operation also like for expensive medicines or in receiving blood.

1.6 SOFTWARE AND HARDWARE REQUIREMENTS

SOFTWARE REQUIREMENTS:

.NET

.NET (pronounced as "dot net"; formerly named .NET Core) is a free and open-source, managed computer software framework for Windows, Linux, and macOS operating systems. It is a cross-platform successor to .NET Framework. The project is mainly developed by Microsoft employees by way of the .NET Foundation, and released under an MIT License. .NET MAUI also supports Android and iOS (also with .NET 6), i.e., .NET Multi-platform App UI (.NET MAUI) is a cross-platform framework for creating native mobile and desktop apps with C# and Extensible Application Markup Language (XAML). .NET 7 (released in 2022, .NET 6 LTS is the

only other currently supported release) is for many features fully compatible, but Microsoft has a long list on incompatible changes, often individual features are no longer binary compatible while source compatibility is retained, in some cases vice versa, and in rare cases neither.

VISUAL STUDIO

Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs including websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works as both a source-level debugger and as a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level - including adding support for source control systems (like Subversion and Git) and adding new tool sets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Azure DevOps client: Team Explorer).

HARDWARE REQUIREMENTS:

1. RAM 4GB
2. Processor Intel® core™ i5
3. WINDOWS 10 64-bit OS
4. For users, the system is designed to include a mobile based application. The user will need to download the application from Google PlayStore

CHAPTER 2

SYSTEMS ANALYSIS AND SPECIFICATION

2.1 FUNCTIONAL MODEL

The purposes of the function model are to describe the functions and processes, assist with discovery of information needs, help identify opportunities, and establish a basis for determining product and service costs.

2.1.1 DATA FLOW DIAGRAM (DFD)

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.

2.1.1.1 ZERO LEVEL DATA FLOW DIAGRAM (0 Level DFD)

This is the Zero Level DFD of Blood Bank Management System, where we have elaborated the high-level process of Blood Bank. It's a basic overview of the whole Blood Bank Management System or process being modeled. It's designed to glance at Donor and Patient showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including Acceptor and Donor.

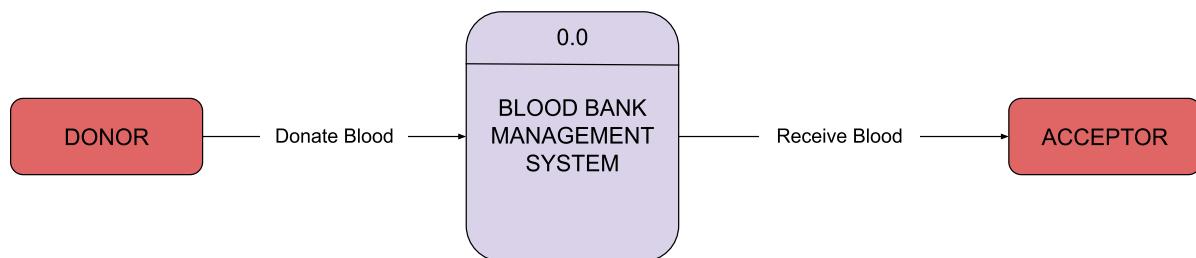


Figure: 2.1.a DFD Level 0

High Level Entities and process flow of Blood Bank Management System:

- Managing all the Blood.
- Managing all the Blood Group.
- Managing all the Donors.
- Managing all the Order.
- Managing all the Acceptors.

2.1.1.2 FIRST LEVEL DATA FLOW DIAGRAM (1st Level DFD)

First Level DFD (1st Level) of Blood Bank Management System shows how

The system is divided into subsystems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Blood Bank Management System as a whole. It also identifies internal data stores of Patient, Order, Donor that must be present in order for the Blood Bank system to do its job, and shows the flow of data between the various parts of Order, Patient, Donor of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD.

Main entities and output of First Level DFD (1st Level DFD):

- Processing Blood records and generating reports of all Blood.
- Processing Blood Group records and generating reports of all Blood Group.
- Processing Donor records and generating reports of all Donors.
- Processing Order records and generating reports of all Order.
- Processing Acceptor records and generating reports of all Acceptor.

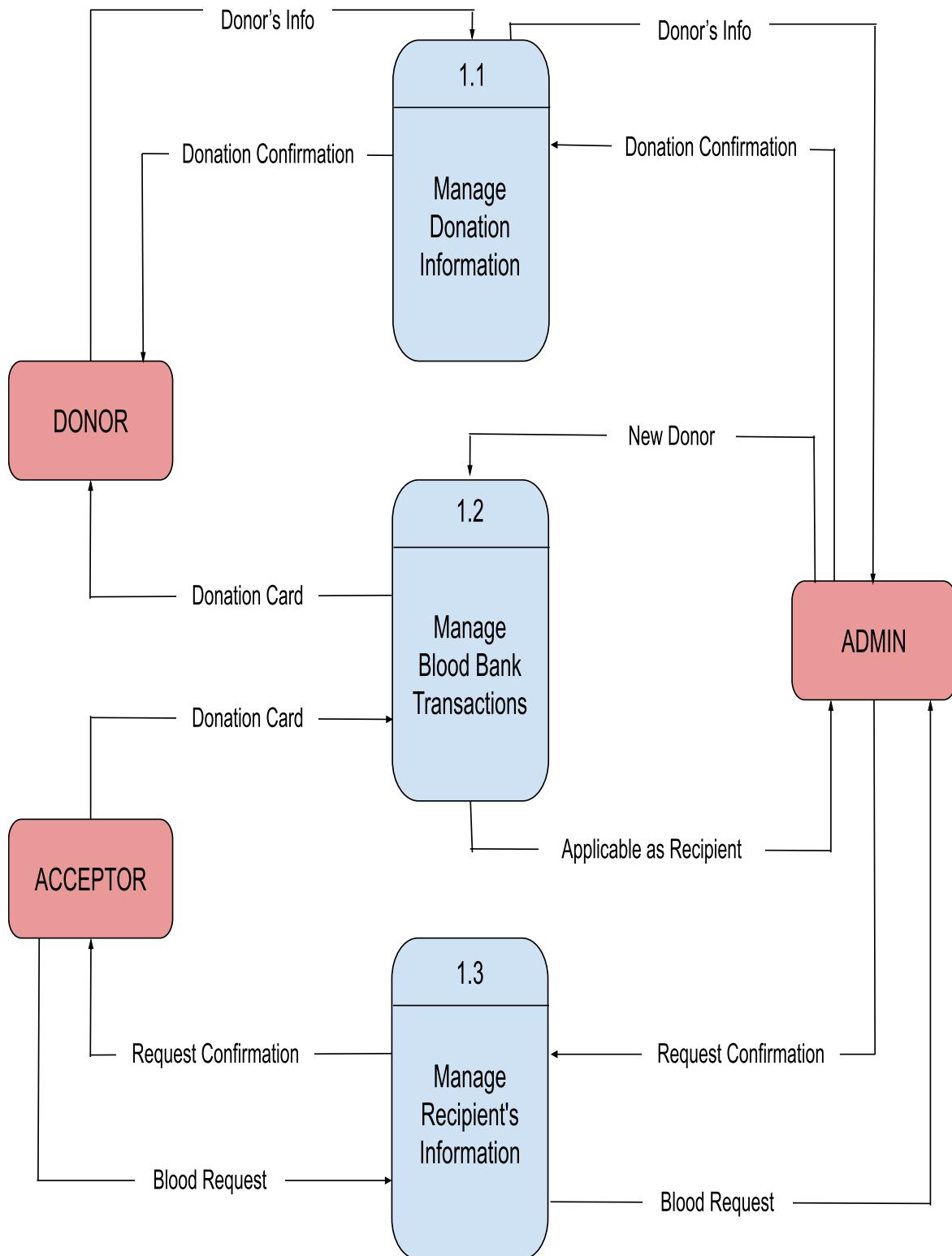


Figure: 2.1.b DFD Level 1

2.1.1.3 SECOND LEVEL DATA FLOW DIAGRAM (2nd Level DFD)

DFD Level 2 then goes one step deeper into parts of Level 1 of Blood Bank. It may require more functionalities of Blood Bank to reach the necessary level of detail about the Blood Bank functioning. First Level DFD (1st Level) of Blood Bank Management System shows how the system is divided into subsystems (processes). The 2nd Level DFD contains more details of Acceptor, Order, Donor, Blood Group, Blood.

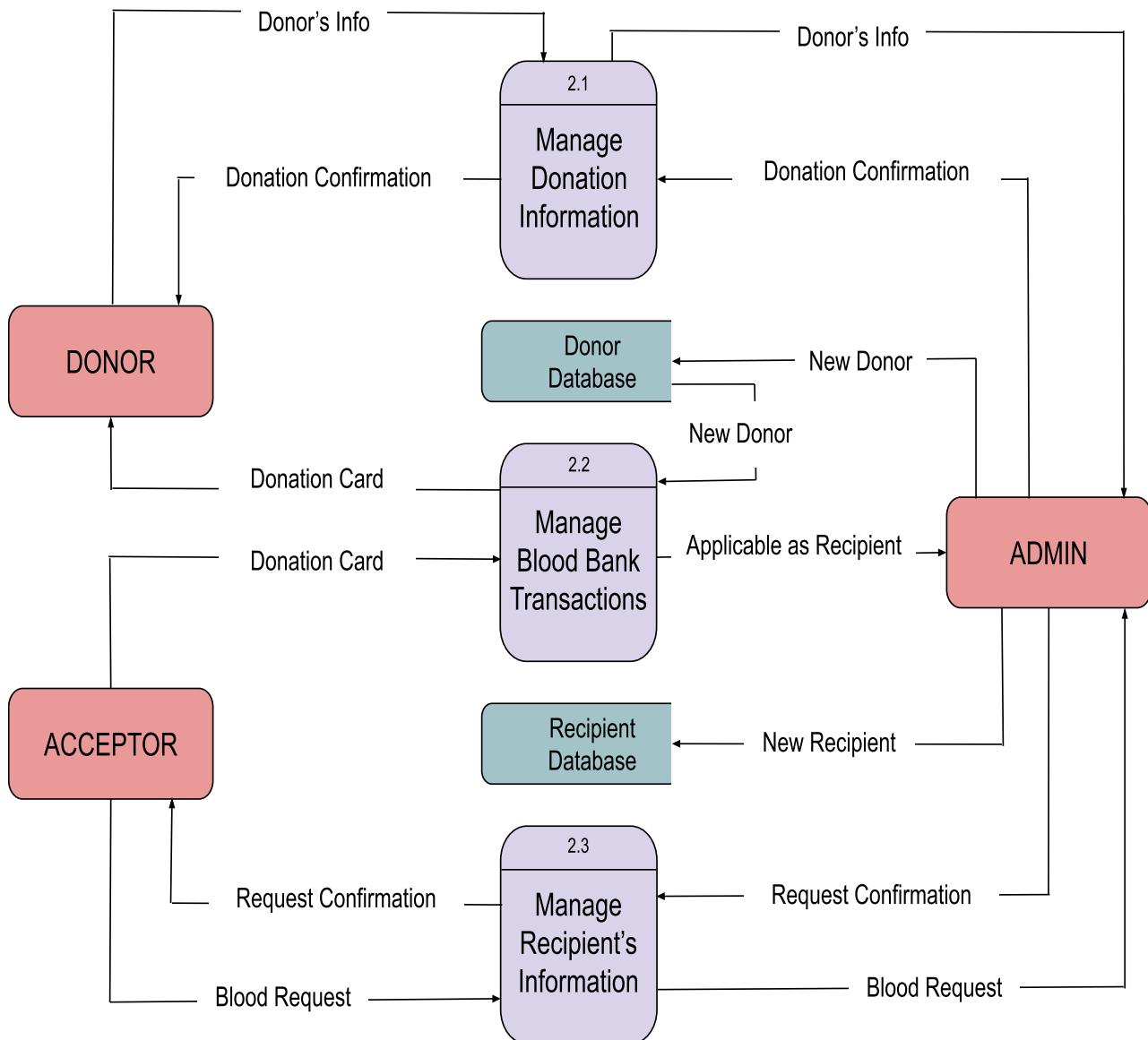


Figure: 2.1.c DFD Level 2

Low level functionalities of Blood Bank Management System

- Admin logins to the system and manages all the functionalities of the Blood Bank Management System.
- Admin can add, edit, delete and view the records of Blood, Donor, Acceptor.
- Admin can manage all the details of Blood Group and requests.
- Admin can also generate reports of Blood, Blood Group, Donor, Order.
- Admin can search the details of Blood Group, Donor, Order.
- Admin can apply different levels of filters on reports of Blood, Blood Stock, Donor.
- Admin can track the detailed information of Blood Group, Donor.

2.2 DATA MODEL

Data models are visual representations of an enterprise's data elements and the connections between them. By helping to define and structure data in the context of relevant business processes, models support the development of effective information systems.

2.2.1 ER DIAGRAM

This ER (Entity Relationship) Diagram represents the model of Blood Bank Management System Entity. The entity-relationship diagram of Blood Bank Management System shows all the visual instruments of database tables and the relations between Blood Group, Donor, Blood, Order etc. It used structure data and to define the relationships between structured data groups of Blood Bank Management System functionalities. The main entities of the Blood Bank Management System are Blood, Blood Group, Blood cells, Donor, Stock and Order.

Blood Bank Management System entities and their attributes:

- Blood bank entity: bloodbank_id, bloodbank_name, mobile_no., address, city, state, country, pincode.
- Donor Entity: donor_id, name, blood, gender, dob, email_id, mobile_no, address, city, state, country, pincode.
- Acceptor Entity: request_id, name, mobile_no, request.
- Admin Entity: login_id, username, password.

Description of Blood Bank Management System Database:

- The details of Blood are store into the Blood tables respective with all tables
- Each entity contains primary keys and unique keys.
- There are one-to-one and one-to-many relationships available.

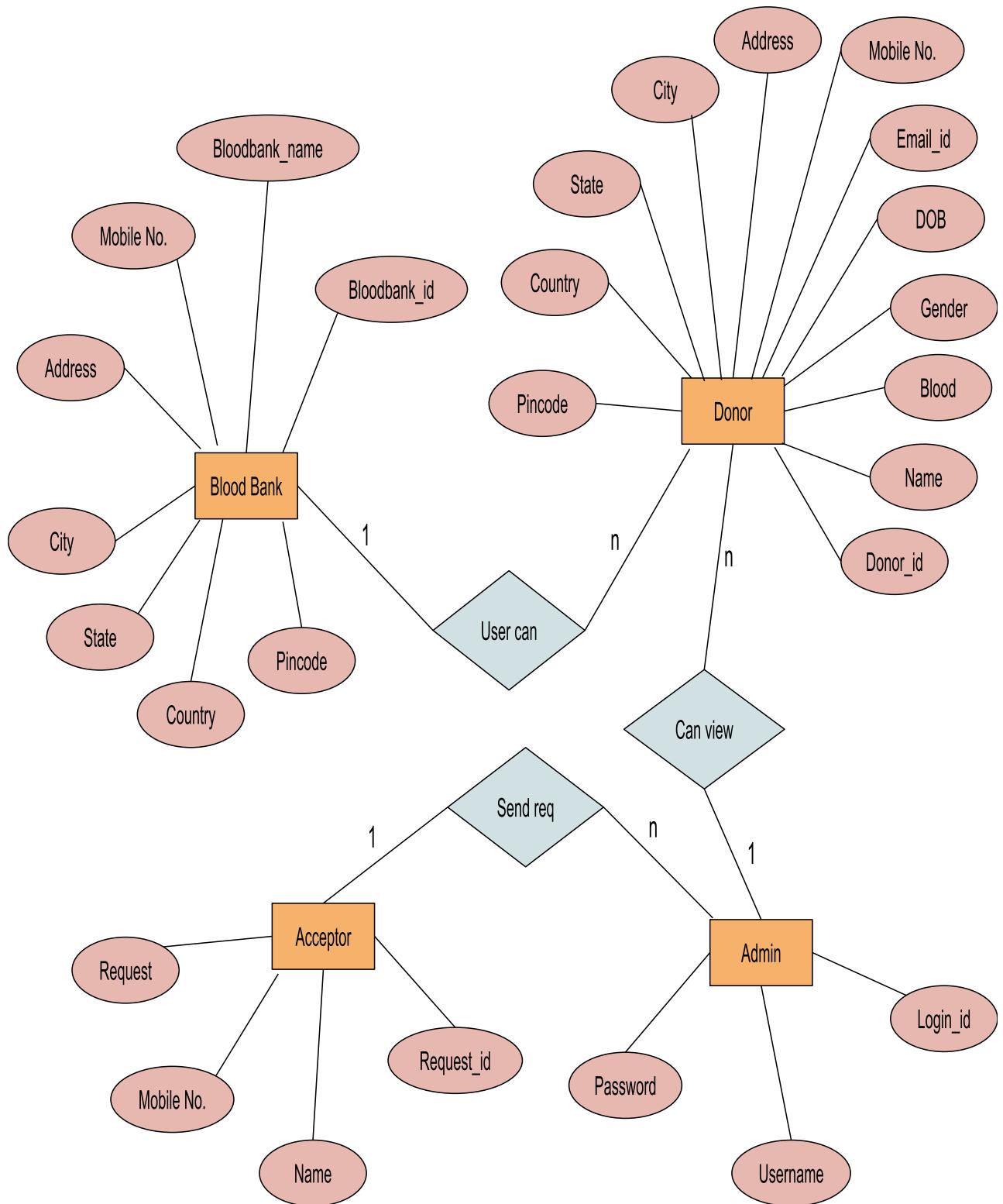


Fig: 2.2.a ER Diagram

2.3 PROCESS FLOW MODEL

A process model is a formal way of representing how a business operates. A process flow diagram is one method of representing a process model.

2.3.1 ACTIVITY DIAGRAM

This is the Activity UML diagram of Blood Bank Management System which shows the flows between the activity of Stock, Blood, Acceptor, Donor. The main activity involved in this UML Activity Diagram of Blood Bank Management System are as follows.

- Blood Activity
- Acceptor Activity
- Donor Activity

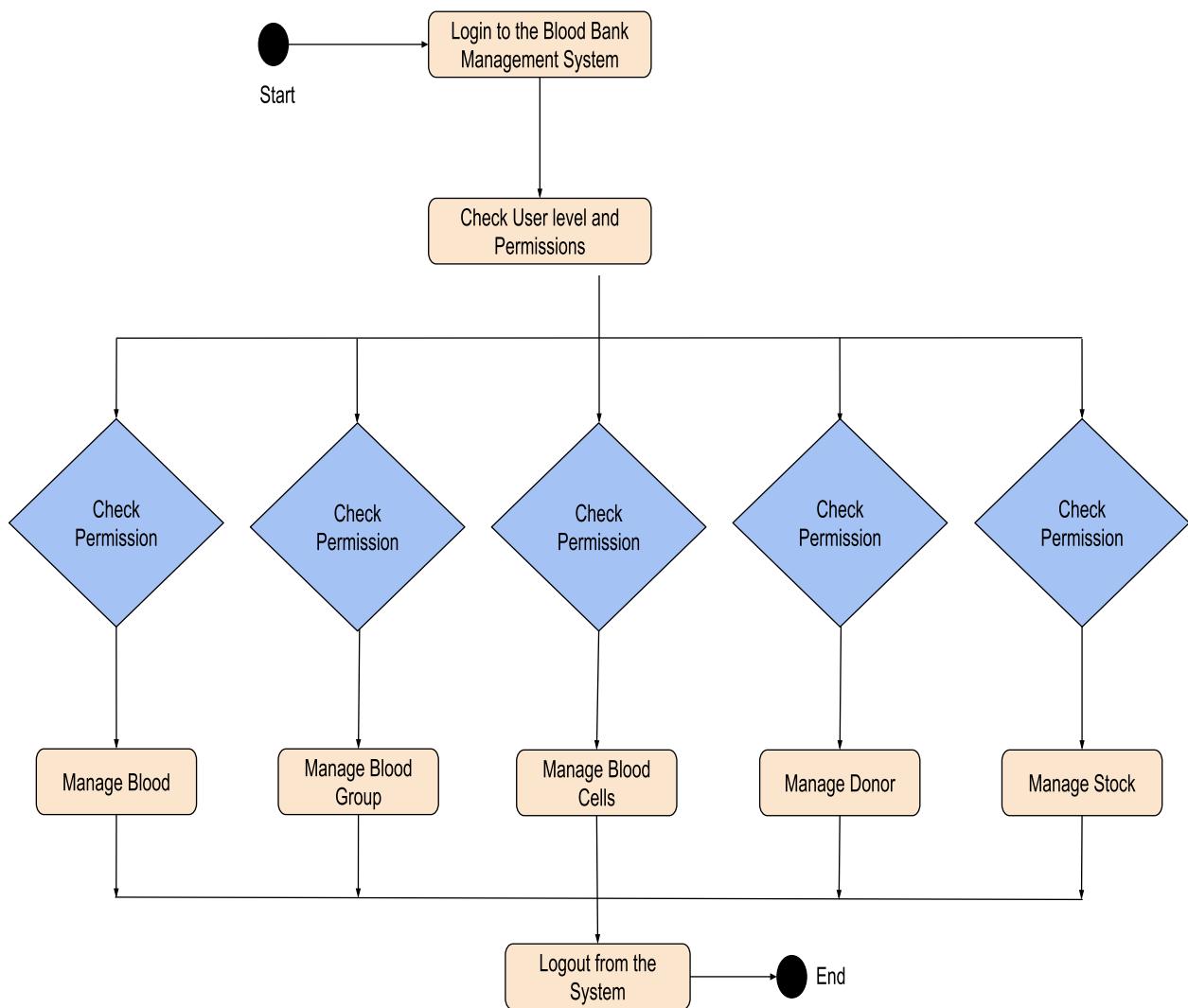


Fig: 2.3.a Activity Diagram

Features Of the Activity UML Diagram of Blood Bank Management System

- Admin User can search Stock, view description of a selected Stock, add Stock, update Stock and delete Stock.
- It shows the activity flow of editing, adding and updating of Blood
- User will be able to search and generate report of Blood cells, Patient, Donor
- All objects such as (Stock, Blood, Donor) are interlinked
- It shows the full description and flow of Stock, Patient, Donor, Blood cells, Blood.

2.4 BEHAVIORAL DIAGRAM

Behavioral modeling is an approach used by companies to better understand and predict consumer actions. Behavioral modeling uses available consumer and business spending data to estimate future behavior in specific circumstances.

2.4.1 SEQUENCE DIAGRAM

A sequence diagram shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. Sequence diagrams are typically associated with use case realizations in the 4+1 architectural view model of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

This is the UML sequence diagram of Blood Bank Management System which shows the interaction between the objects of Blood, Donor, Order, Patient, Blood Group. The instance of class objects involved in this UML Sequence Diagram of Blood Bank Management System are as follows:

- Blood management
- Donor Management
- Order Management
- Blood Group Management

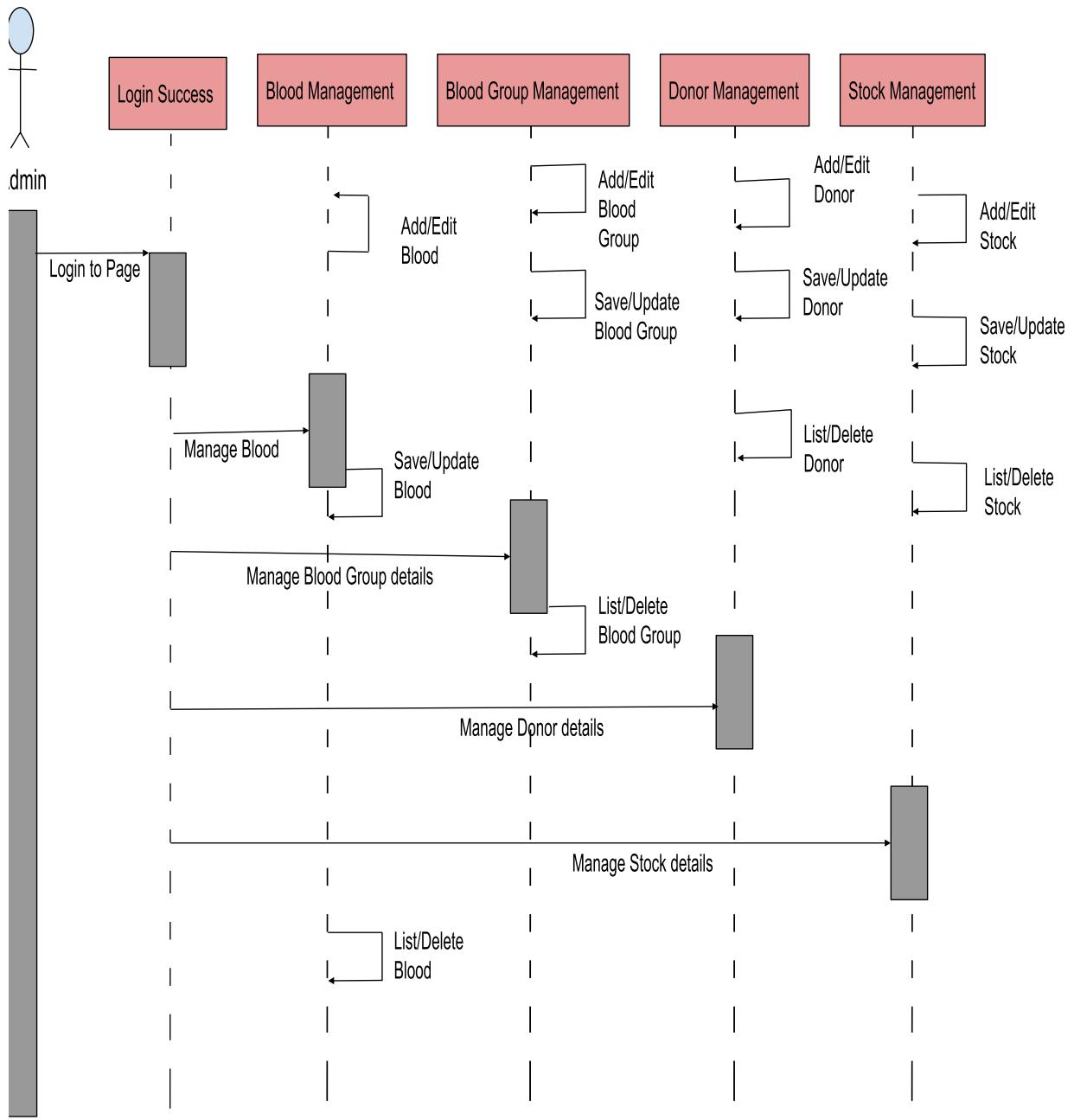


Fig: 2.4.a Sequence Diagram

2.4.2 CLASS DIAGRAM

Blood Bank Management System Class Diagram describes the structure of a Blood Bank Management System classes, their attributes, operations (or methods), and the relationships among objects. The main classes of the Blood Bank Management System are Blood, Blood Group, Blood cells, Donor, Stock, Order.

Classes of Blood Bank Management System Class Diagram

- Blood Class: Manage all the operations of Blood
- Blood Group Class Manage all the operations of Blood Group
- Blood cells Class: Manage all the operations of Blood cells
- Donor Class Manage all the operations of Donor
- Stock Class Manage all the operations of Stock
- Order Class Manage all the operations of Order

Classes and their attributes of Blood Bank Management System Class Diagram:

- Hospital Attributes: hospital_id, request_date, member_type, bloodgroup, quantity, request_location, request_contact, member_name.
- Acceptor Attributes: acceptor_id, name, age, city, bloodgroup, phone_no, gender, registration.
- Donor Attributes: donor_id, donor_name, contact_no, bloodgroup, contact_no., location, donor_city.
- Blood Bank Attributes: bloodbank_id, name, address, phone, website, email, location, city, blood_stock.
- Request Attributes: request_id, request_date, member_type, bloodgroup, quantity, request_location, request_contact, member_name.
- Chat Attributes: chat_id, sender_id, receiver_id, message, date/time.
- Blood Stock Attributes: id, bloobank_name, bag_no, source, bloodgroup, blood_type, quantity, hemolysis, comment

Classes and their methods of Blood Bank Management System Class Diagram:

- Hospital_Method: requestblood(), purchaseblood()
- Acceptor Methods: registration(), search(), make requests(), online chat()
- Blood bank Methods: request blood(), purchase blood(), maintain blood()
- Request Methods: visible to everyone(), represents blood()
- Donor Methods: registration(), search request(), withdraw details(), online chat()
- Chat Methods: donor chat(), accepter chat()
- Blood stock Methods: blood stock maintainability(), view to users()

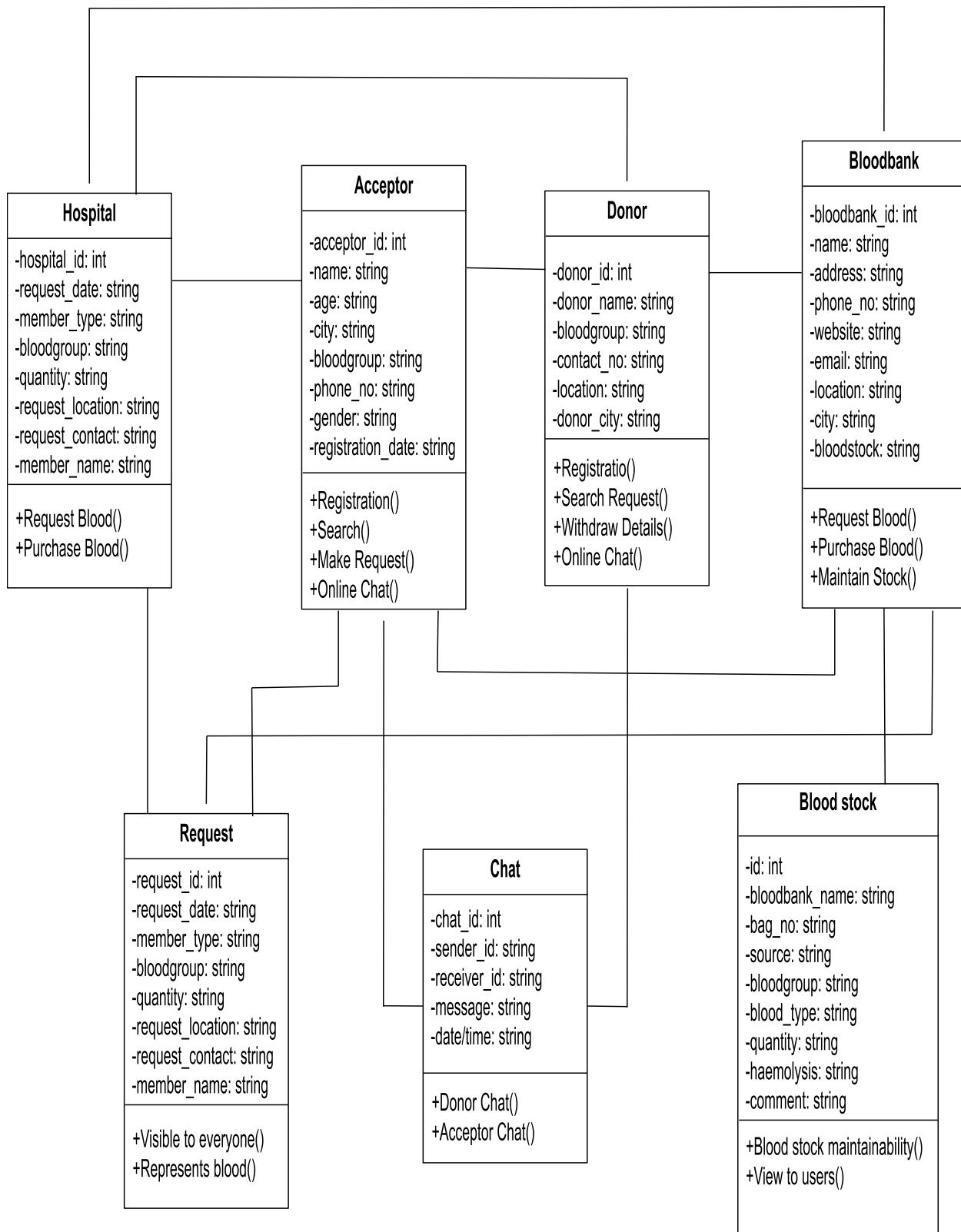


Fig: 2.4.b Class Diagram

2.5 SYSTEM DESIGN

Behavioral modeling is an approach used by companies to better understand and predict consumer actions. Behavioral modeling uses available consumer and business spending data to estimate future behavior in specific circumstances.

2.5.1 FEASIBILITY STUDY:

The feasibility study is performed to determine whether the proposed system is viable considering the Technical, Operational and Economical factors. After going through a feasibility study, we can have a clear-cut view of the system's benefits and drawbacks.

2.5.1.1 TECHNICAL FEASIBILITY:

The proposed system is developed using Visual Studio, Xamarin as front-end tool and C-Sharp as the back end. The proposed system needs an Android Device (Min Version 4.0) to serve the requests submitted by the users. All the required hardware and software are readily available in the market. Hence the system is technically feasible.

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Does the proposed equipment have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the system.

2.5.1.2 OPERATIONAL FEASIBILITY:

The proposed system is operationally feasible because of the following reasons.

- The customer is benefited more as most of his time is saved.
- The cost of the proposed system is almost negligible when compared to the benefits gained.

Proposed projects are beneficial only if they can be turned into an information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.5.1.3 ECONOMICAL FEASIBILITY:

As the necessary hardware and software are available in the market at a low cost, the initial investment is the only cost incurred and does not need any further enhancements. Hence it is economically feasible. The system is feasible in all respects and hence it encourages taking up the system design. A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any additional hardware or software. Since the interface for this system is developed using the existing resources and technologies. There is nominal expenditure and economical feasibility for certain.

CHAPTER-3

MODULE IMPLEMENTATION & SYSTEM INTEGRATION

This section describes the basic system design goals, functionality and architecture. It includes a high level description of the approach used to develop the system design. It also includes high-level descriptions of the systems hardware, software, database, and security components. Depending on the complexity of the system this section may also include component and/or contextual diagrams of the system and system components.

3.1 MODULE IMPLEMENTATION

This is the flow chart of the functional model of the project fundamentals as each phase is required to be passed through before going to the next as the output of one phase could be the input of the next phase now we will go through with each phase one by one.

i. Splash Screen: The very first phase of this app is a splash screen which will remain open for 2 sec before starting anything in the app.



Figure: 3.1.a Splash Screen

After splash screen users will be redirected to the Login or Signup page.

ii. User Details: It is the phase in which the user is responsible to enter the data in the corresponding form in an appropriate manner:-

- a) Name
- b) Phone No.
- c) Email
- d) Password
- e) Blood Group
- f) Location Access

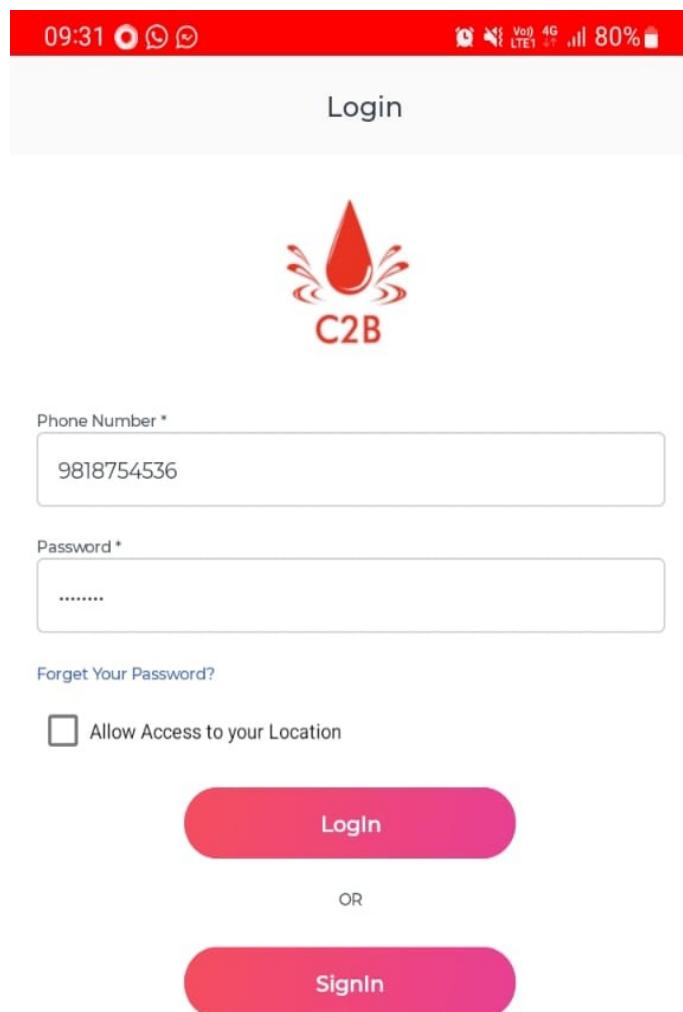


Figure: 3.1.b User Login Screen

After entering the details in the form, users have to submit the details and OTP for registration process after this the page will redirect to the next step of the process.



Name *

Phone Number *

Email *

Password *

O T P

Enter One time Password

BloodGroup

Choose Blood Group

4 5 3 6

Allow Access to your Location

SignUp

OR

Login

SUBMIT

Figure: 3.1.c User Signup Screen

Figure: 3.1.d OTP Screen

iii. Main Screen: On Main Screen User can see a Home, Menu and About Us on bottom of page and a Map with location of nearby hospitals, a donor button and a receiver button.

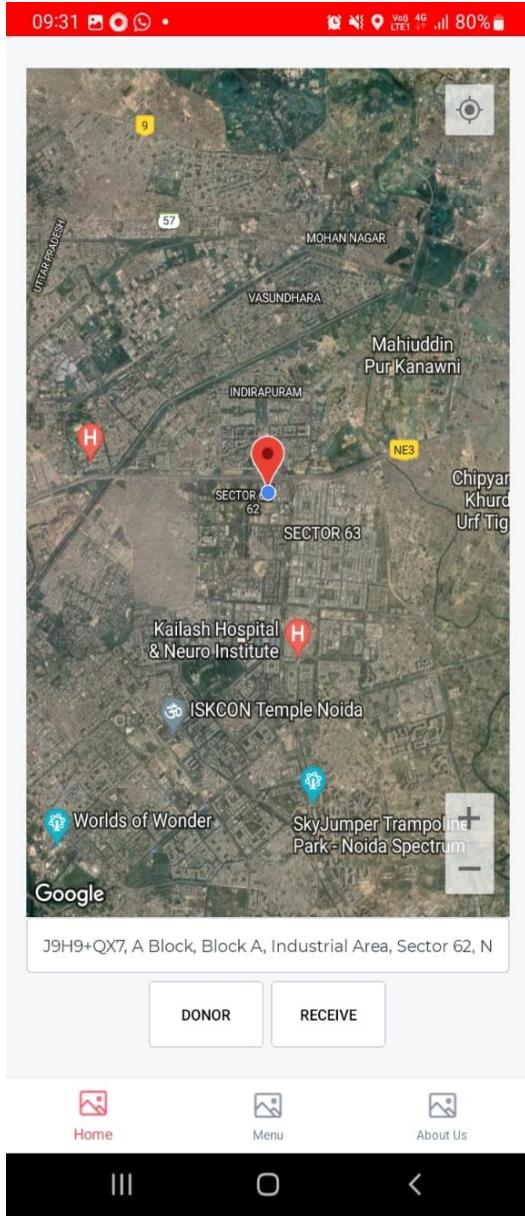


Figure: 3.1.e Main Screen

By clicking on these buttons users will be led to various pages.

iv. Menu: If the user clicks on the Menu from the main screen various options will be shown such as Profile, Insurance, Donate History, Ambulance Service, Privacy Policy, Help, Refer Friend, RateUS each having their own functionality.

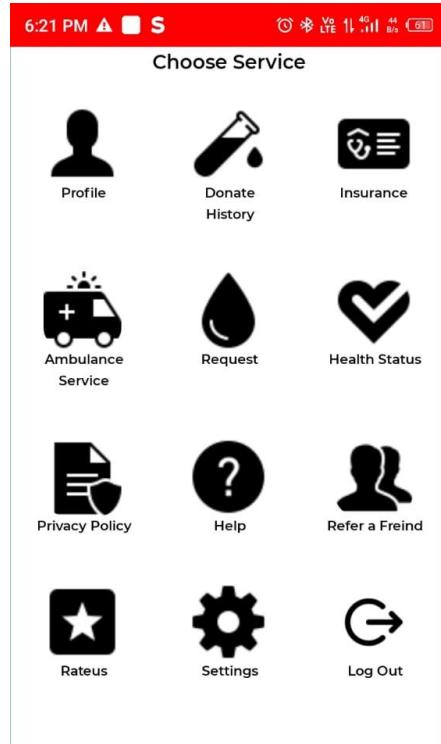


Figure: 3.1.f Menu Screen

a) **Profile:** Here users can check their basic information.

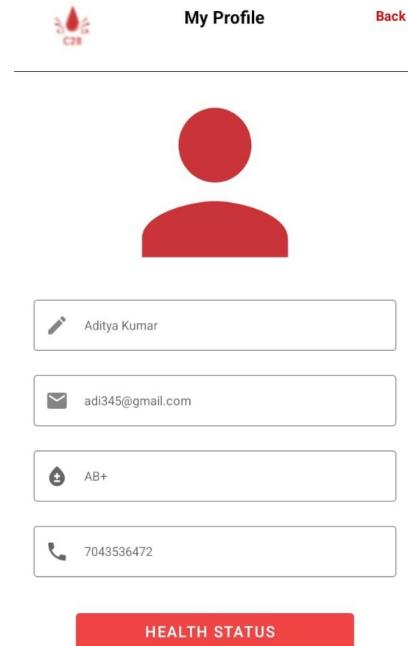


Figure: 3.1.g Profile Screen

b) Insurance: Users can apply for insurance from here

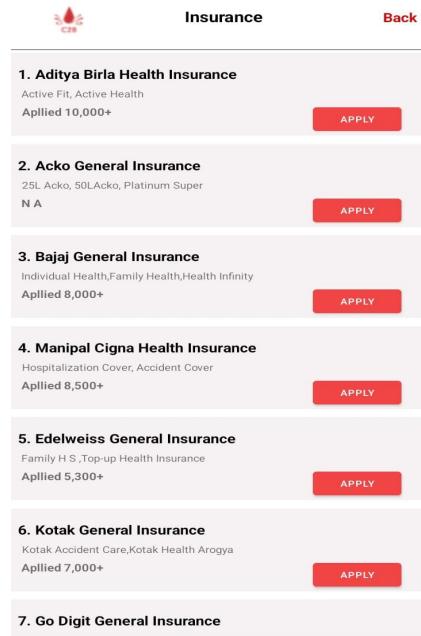


Figure: 3.1.h Insurance Screen

c) Donate History: The donations made by users are available here.

d) Ambulance Service: Ambulance Services are available in case of emergencies.

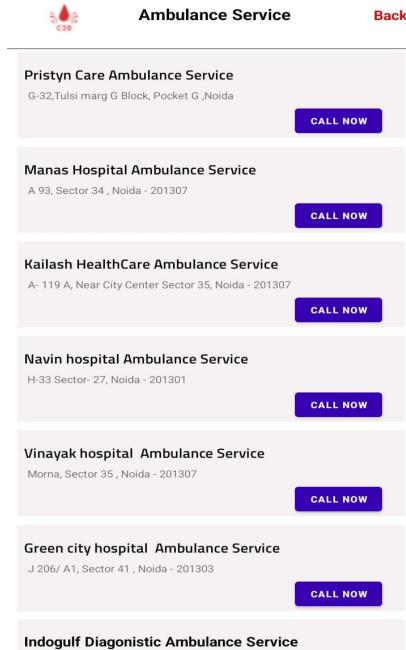


Figure: 3.1.j Ambulance Service Screen

- e) **Privacy Policy:** Users can check our privacy policies here.

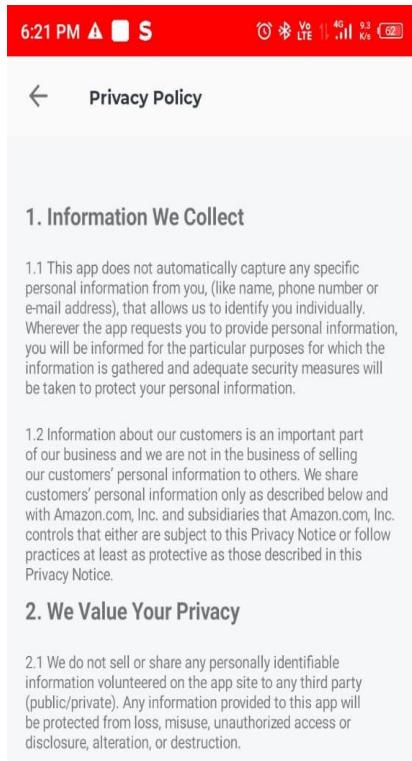


Figure: 3.1.k Privacy Policy Screen

- f) **Help:** Users can seek help here if they want any.
g) **Refer Friend:** Friends can also be referred so they can use the app for their betterment.
h) **Rate Us:** Users can rate their app experience through this.

v. **Donor:**

If the user wants to donate the blood, he can choose this button for further processes. User will be redirected to a **FAQ** page where he will answer some questions, based on the answers by the user it will decide whether the user is eligible to donate the blood or not. If a user is eligible to donate the blood he will be redirected to the donation page where he can donate to any one of the given options. After choosing one option, the donor will be directed to the information page where he fills the donor's information. (If the user himself is the donor, auto filling of information will be done, while if the donor is not the user or someone else from users id he needs to fill the information.) Now Donors can Donate the Blood.

Donor Screen (Left):

- Q1. Facing Common health Problems (cold,flu,sore throat,cold sore, stomach infections)?
 Yes
 No
- Q2. Had any detail procedures done such as filling , cleaning or restoration(the day before restoration)?
 Yes
 No
- Q3. Are you diabetic and taking insulin injections to manage diabetes. (If you have your blood glucose level under control with diet or oral medication then you can donate blood)?
 Yes
 No
- Q4. History of genital ulcers , multiple sex partners , or drug addiction?
 Yes
 No
- Q5. Have hepatitis B, hepatitis C, tuberculosis,leprosy , human deficiency virus (HIV), heart disease epilepsy, bleeding disorders, thalassemia , sickle cells anemia and cancer?
 Yes
 No

Alert Message (Middle):

Sorry you are not a eligible to donate the blood

OK

Employee Services Form (Right):

- BloodGroup
 Choose Blood Group
- Name *
 Enter Name
- Phone Number *
 Enter Phone No
- Submit

Figure: 3.1.1 Donor Screen

vi. Receiver:

If the user wants to receive the blood, he can choose this button. Users will be redirected to the available blood near them, they can select any one of the options and then proceed further. After proceeding, the information page for Receiver is redirected. (If the receiver is the user himself, autofill done or if the receiver is someone else from the user's id he needs to fill the information). Now Receiver can receive the blood.

The screenshot shows a smartphone interface with a red header bar containing icons for time (6:30 PM), signal strength, battery level (60%), and other connectivity status. Below the header is a white header bar with a left arrow icon and the text "Receiver List". The main content is a table with the following data:

DonorName	PhoneNumber	BloodGroup	Address
Ajit Singh	9931320639	A+	Sector-18 Noida
Shivendra Nath	9313803940	A+	Sector-9 Noida
Satyam Tiwari	9433803930	A+	Sector-62 Noida
Shasank Tiwari	9138034340	A+	Sector-15 Noida
Shivraj	9213804940	A+	Sector-16 Noida
Jyoti Sharma	9813703940	A+	Sector-65 Noida
Ragahv Jain	9813873950	A+	Sector-63 Noida
Rishav Choudary	9713883940	A+	Sector-59 Noida
Amit Kumar	7013803940	A+	Sector-51 Noida
Kapil Madan	8413803940	A+	Sector-45 Noida

Figure: 3.1.m Receiver Screen

3.2 SYSTEM INTEGRATION

System has to provide services of features that tackle many problems as indicated briefly:

- Solve the problem concerning the lack of blood quantities by making the blood available to improve the healthcare services.
- Find the right donors at the right time.
- Provide the full management system for patients who need blood.
- Live monitoring for blood types, current quantities, and recommendations from hospitals.
- People won't pay money to get the blood so they will save their money.
- There will be more blood bags in all hospitals (i.e., supporter live-events).
- Solve the problem of delaying, which happens when the patients need the blood.
- Improves the confidence between the government and our system through enhancing the medical sector of the Ministry of Health.
- Organize the distribution process of ambulances on all regions to avoid the problems like the ambulances were founded in regions don't have any blood donor or in regions suffer from lacking the blood type that has a greatest need, this solution save the time and efforts so the process become more organized.

CHAPTER-4

TESTING AND EVALUATION

4.1 TESTING

In this chapter we will run the test cases and execute the backend and go through with the work process to run and execute the code.

POSTMAN TESTING

Postman is an API development tool which helps to build, test and modify APIs. Almost any functionality that could be needed by any developer is encapsulated in this tool. It is used by over 5 million developers every month to make their API development easy and simple. It has the ability to make various types of HTTP requests(GET, POST, PUT, PATCH), saving environments for later use, converting the API to code for various languages(like Javascript, Python).

Tools

The Postman platform includes a comprehensive set of tools that help accelerate the API lifecycle from design, testing, documentation, and mocking to the sharing and discoverability of your APIs.

API client

The Postman API client is the foundational tool of Postman, and it enables us to easily explore, debug, and test our APIs while also enabling us to define complex API requests for HTTP, REST, SOAP, GraphQL, and WebSockets. The API client automatically detects the language of the response, links, and format text inside the body to make inspection easy. The client also includes built-in support for authentication protocols like OAuth 1.2/2.0, AWS Signature, Hawk, and many more. Through the API client, we can organize requests into Postman Collections to help us organize our requests for reuse so we don't waste time building everything from scratch.

Our collections can also contain JavaScript code to tie requests together or automate common workflows, and we can use scripting to visualize our API responses as charts and graphs.

API design

We can design our API specifications in Postman using OpenAPI, RAML, GraphQL, or SOAP formats. Postman's schema editor makes it easy to work with specification files of any size, and it validates specifications with a built-in linting engine. We can also generate Postman Collections for multiple stages of our API lifecycle—for mocks, documentation, tests, monitors, and more—all from the specification file, all in sync.

API documentation

Documentation is a core part of our API workflow. Postman supports markdown-enabled and machine-readable documentation through the Postman Collection format, and we can generate docs through our OpenAPI files as well. Our docs will automatically include detail on our requests, with sample code in various client languages. We can share the docs with our team or with the world through workspaces or publish them in a dedicated portal.

API testing

Build and run tests directly in Postman or as part of our CI/CD pipeline through Newman (a Collection Runner that enables us to run and test a Postman Collection directly from the command line). Postman can be used to write functional tests, integration tests, regression tests, and more. Postman's Node.js-based runtime contains support for common patterns and libraries that we can use to build tests quickly.

Mock servers

Mock servers enable us to see exactly how our API will run—even before it's in production. Create mock servers in Postman to simulate API endpoints when we don't want to (or can't) send API requests to a real API. We can also simulate network latency in our mock server by specifying custom delays for responses. Mock servers are hosted on Postman's cloud, which

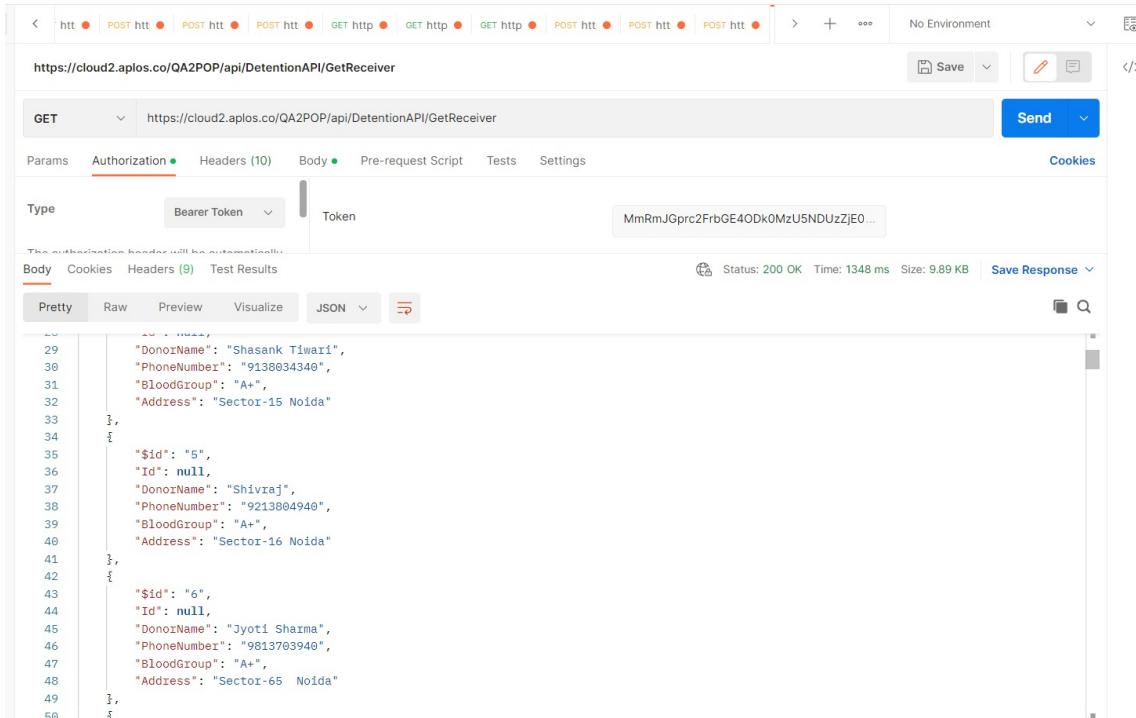
means they're available wherever we need them—whether in our local, testing, or staging environments.

Monitors

Postman monitors help us stay up to date on the health and performance of our APIs. Monitors can be run across multiple geographic regions and integrated with third-party alerting systems and dashboards like Datadog, New Relic, Slack, and more. Postman monitors are hosted on Postman's cloud, which means we can set them up quickly.

API detection

Capture requests and cookies from our browser into Postman to speed up our debugging workflow through the Postman Interceptor or the Postman proxy. Interceptor can catch requests and responses. Postman's proxy runs inside the Postman application and can be used with HTTP or HTTPS websites.



The screenshot shows the Postman interface with a GET request to <https://cloud2.aplos.co/QA2POP/api/DetentionAPI/GetReceiver>. The response body is a JSON array of three donor records:

```
29     "DonorName": "Shasank Tiwari",
30     "PhoneNumber": "9138034340",
31     "BloodGroup": "A+",
32     "Address": "Sector-15 Noida"
33   },
34   {
35     "$id": "5",
36     "Id": null,
37     "DonorName": "Shivraj",
38     "PhoneNumber": "9213804940",
39     "BloodGroup": "A+",
40     "Address": "Sector-16 Noida"
41   },
42   {
43     "$id": "6",
44     "Id": null,
45     "DonorName": "Jyoti Sharma",
46     "PhoneNumber": "9813703940",
47     "BloodGroup": "A+",
48     "Address": "Sector-65 Noida"
49   }
```

Figure: 4.1.a API Testing Image 1

The screenshot shows the Postman application interface. At the top, the URL is https://cloud2.aplos.co/QA2POP/api/DetentionAPI/Getambulance. Below the URL, the method is set to GET. The 'Authorization' tab is selected, showing a Bearer Token with the value MmRmJGprc2FrbGE4ODk0MzU5NDUzZjE0... . The 'Body' tab is also selected, displaying the JSON response:

```

1 [
2   {
3     "$id": "1",
4     "Id": null,
5     "CompanyName": "Pristyn Care Ambulance Service",
6     "PhoneNumber": "9713883940",
7     "Address": "G-32, Tulsim Marg G Block, Pocket G, Noida",
8     "Price": ""
9   },
10  {
11    "$id": "2",
12    "Id": null,
13    "CompanyName": "Apollo Hospital Ambulance Service",
14    "PhoneNumber": "7013803940",
15    "Address": "E 2, Apollo Hospitals Rd, Block E, Sector 26, Noida, Uttar Pradesh 201301, Noida",
16    "Price": ""
17  },
18  {
19    "$id": "3",
20    "Id": null,
21    "CompanyName": "Dipakshi Hospital Ambulance Service",
22    "PhoneNumber": "8413803940",

```

The status bar at the bottom indicates Status: 200 OK, Time: 1086 ms, Size: 5.66 KB, and a Save Response button.

Figure: 4.1.b API Testing Image 2

The screenshot shows the Postman application interface. At the top, the URL is https://cloud2.aplos.co/QA2POP/api/DetentionAPI/GetUserInfo?user=Aman Kumar Rai&password=12345. Below the URL, the method is set to GET. The 'Authorization' tab is selected, showing a Bearer Token with the value MmRmJGprc2FrbGE4ODk0MzU5NDUzZjE0... . The 'Body' tab is also selected, displaying the JSON response:

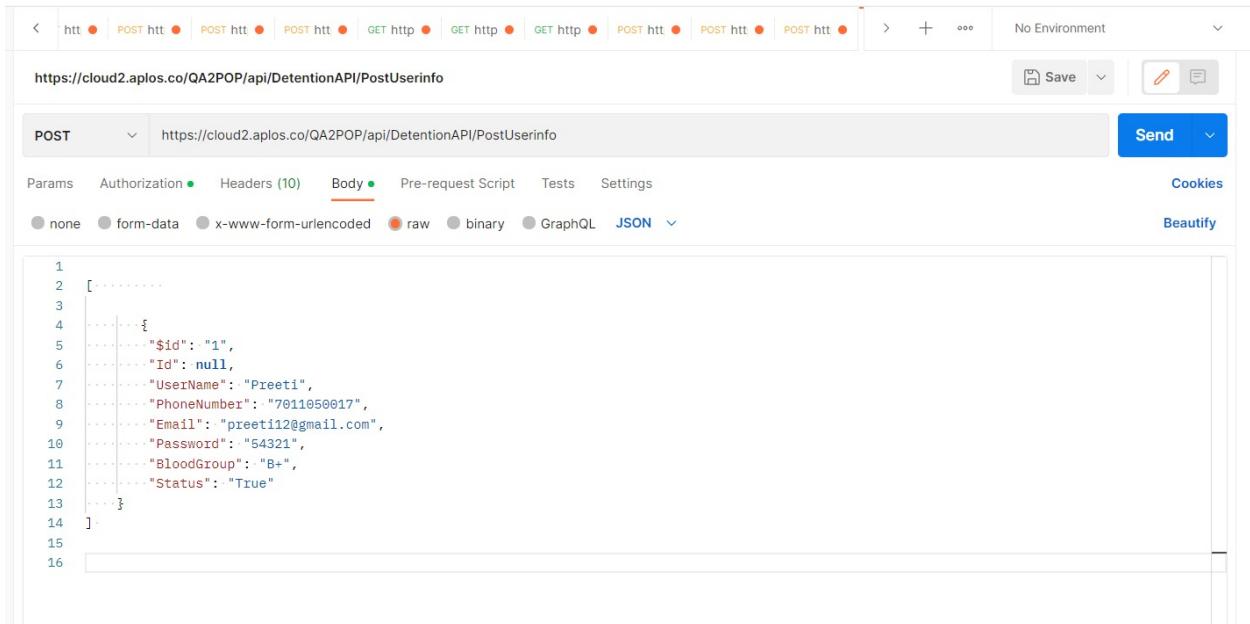
```

1 [
2   {
3     "$id": "1",
4     "Id": null,
5     "UserName": "Aman Kumar Rai",
6     "PhoneNumber": "7642527546",
7     "Email": "1710kraman@gmail.com",
8     "Password": "12345",
9     "BloodGroup": "B+",
10    "Status": "True"
11  }

```

The status bar at the bottom indicates Status: 200 OK, Time: 558 ms, Size: 422 B, and a Save Response button.

Figure: 4.1.c API Testing Image 3



```
1
2 [ ...
3
4   ...
5     ...
6       ...
7         ...
8           ...
9             ...
10            ...
11              ...
12                ...
13                  ...
14 ]
15
16 ]
```

Figure: 4.1.d API Testing Image 4

MANUAL TESTING

Manual Testing is the Software Testing Technique that is carried out using the functions and features of an application. In manual testing, a tester carries out tests on the software by following a set of predefined test cases. In this testing, testers make test cases for the codes and test the software and give the final report about that software. Manual testing is time-consuming testing because it is done by humans and there is a chance of human errors. Every new application must be manually tested before its testing can be automated. Manual testing requires more effort rather than automation testing but is necessary to check automation feasibility. There is no requirement of knowledge of any testing tool in manual testing.

Tools

Selenium

Selenium is a manual testing tool that is used for testing the web application. It provides a playback tool for authoring functional tests without needing to learn test scripting language. It

also provides domain-specific language to write tests in various programming languages such as C#, groovy, PHP, Perl, java, python, ruby, and Scala. It is an open-source tool that supports operating systems like Windows, Linux, MacOS. Selenium IDE, Selenium Client API, Selenium remote control, selenium webDriver, and selenium Grid are the components of selenium.

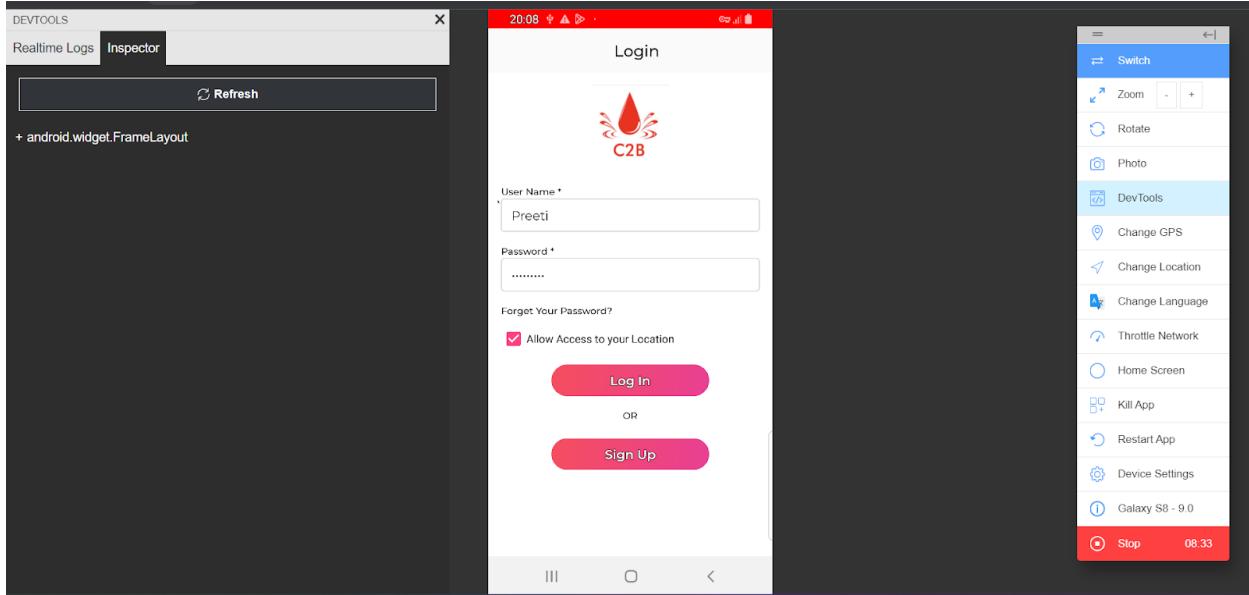


Figure: 4.1.e Manual Testing Setup

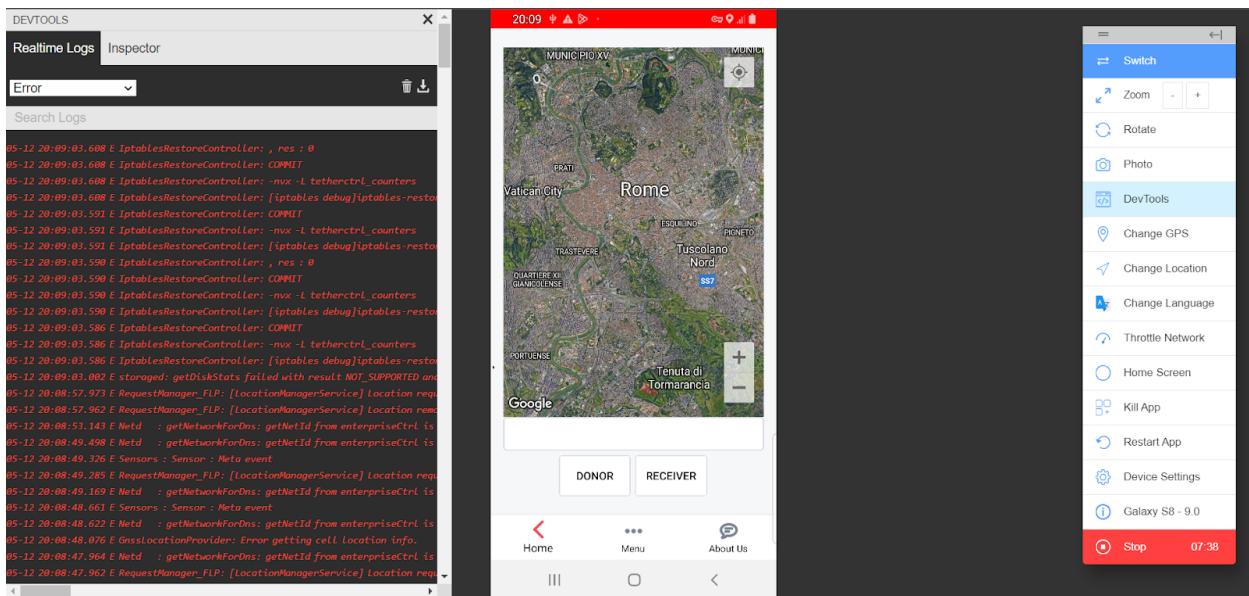


Figure: 4.1.f Errors while Testing

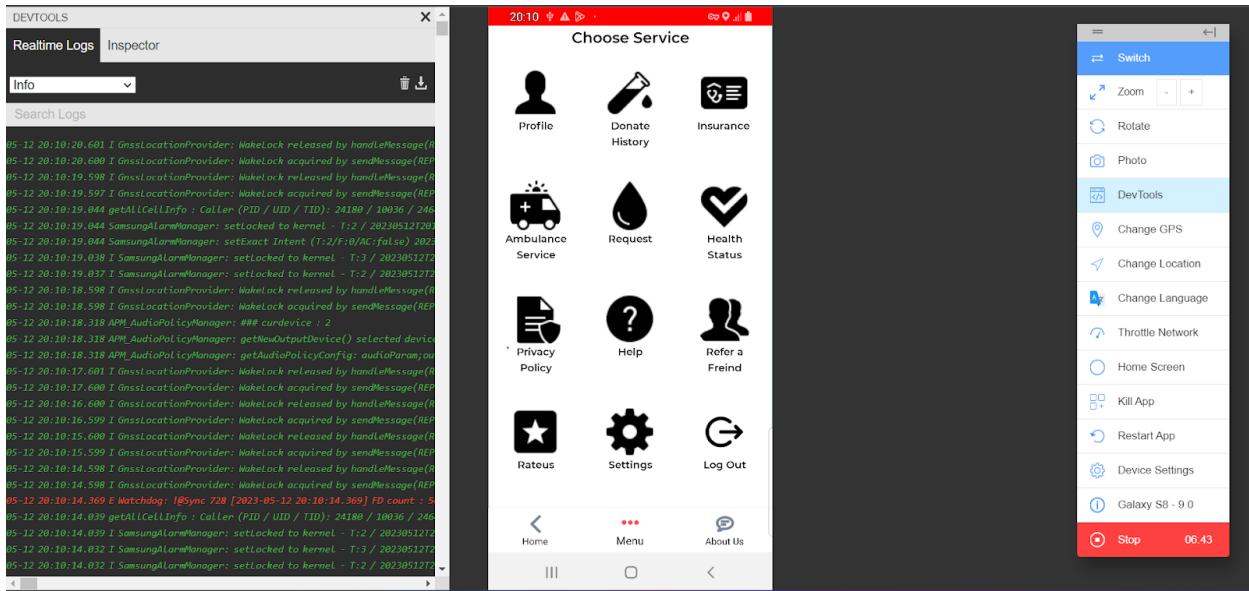


Figure: 4.1.g Manual Testing 1

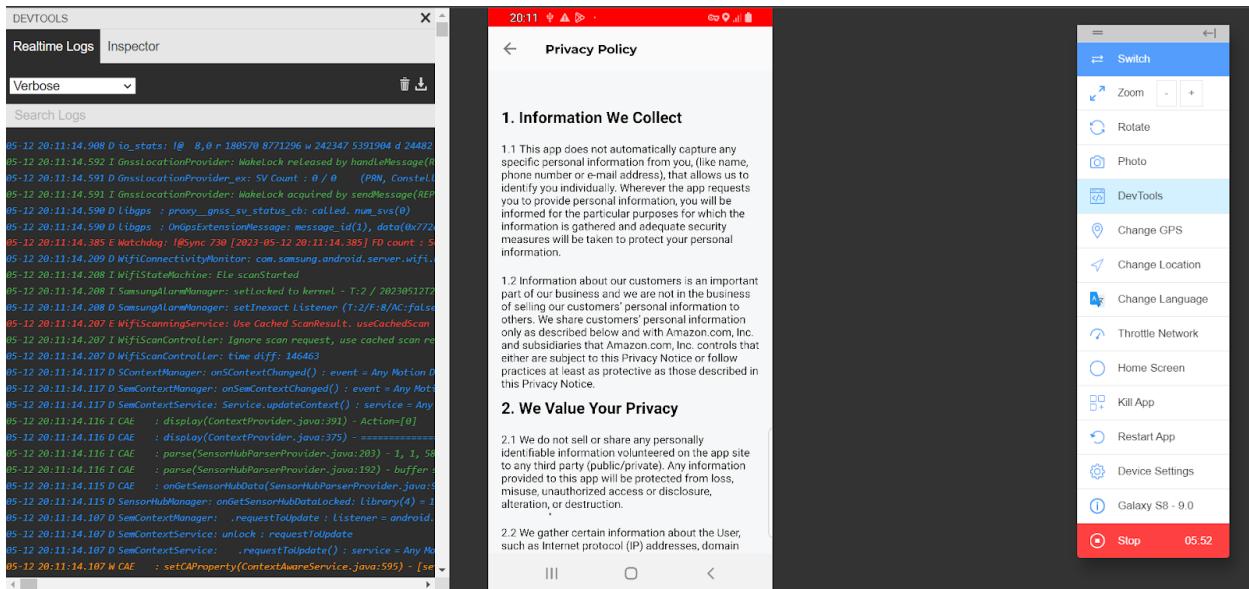


Figure: 4.1.h Manual Testing 2

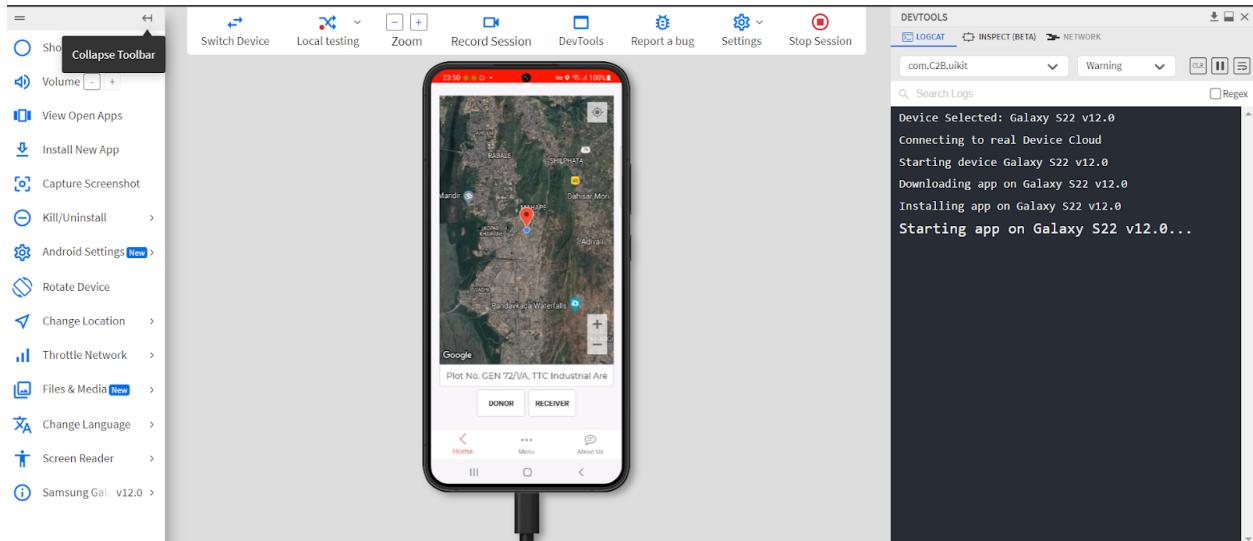


Figure: 4.1.i Manual Testing 3

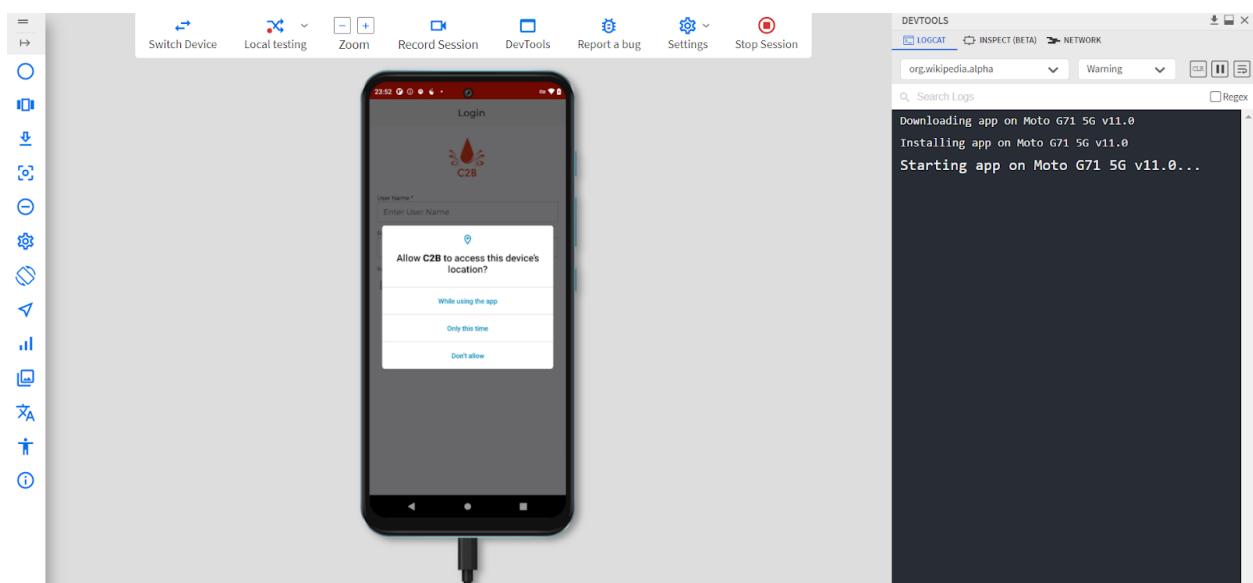


Figure: 4.1.j Manual Testing 4

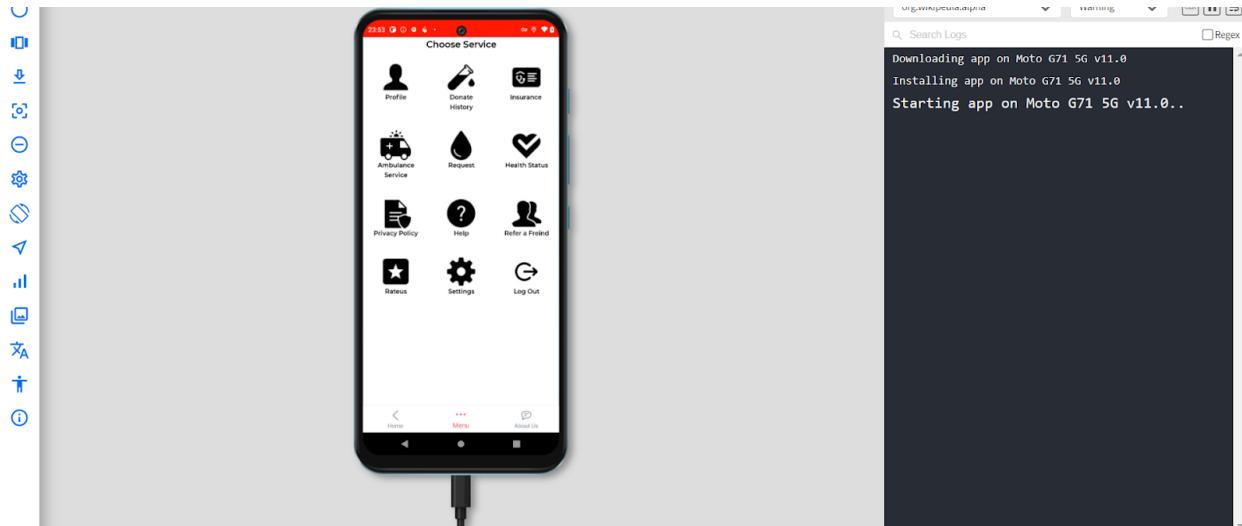


Figure: 4.1.k Manual Testing 5

CHAPTER-5

TASK TIMELINE & SCHEDULE OF ACTIVITIES

5.1 ANALYSIS

1. Study of research papers - (27 June 2022 - 20 July 2022) In this stage we researched different research papers that were available related to the Topic of our Project. We got an idea of what all work has already been done on the selected topic and what we need to do further to make it useful.
2. Perform System Analysis - (21 July 2022 - 1 August 2022) In this step we analyzed the system and got to know about the problems, opportunities and discoveries.
3. Study of Technologies to be used - (2 August 2022 - 15 September 2022) At this stage we were ready to begin with the actual study of the technologies that were to be used in our project. We learnt about Xamarin, C-Sharp, SQL Technologies and their various uses and benefits.
4. Assign Task - (16 October 2022 - 20 October 2022) After learning the technologies and understanding which member will be able to do which type of work, we assigned different parts of the project to all the team members. All the members worked on their part and would also help the other team members if needed.

5.2 DESIGN

1. Perform System Design - (21 October 2022 - 10 November 2022) At this stage, we defined the architecture, product design, modules, interfaces, and data for our project to satisfy the specified requirements.

5.3 BUILD

1. Develop System - (11 November 2022 - 22 February 2023) This was one of the most crucial parts of our project where we built our project and actually started combining the different pieces of the project together to make a final project.

5.4 TESTING

1. Perform Unit Testing - (23 February 2023 - 27 February 2023) In unit testing, the individual parts of source code were tested to determine whether they fit to use or not.
2. Perform Integration Testing - (28 February 2023 - 10 March 2023) At this stage, individual modules were combined and tested as a group. It is conducted to evaluate the compliance of a system or component with specific functional requirements.

5.5 MAINTENANCE

1. Deploy System - (11 March 2023 - 31 March 2023) After all the above steps, finally the product was moved from a temporary or development state to a permanent or desired state.
2. Perform Bug Fixes and Maintenance - (01 April 2023 - 20 April 2023) In this state we used different methods to perform bug fixes and maintenance.

CHAPTER 6

PROJECT MANAGEMENT

6.1 Major Risk and Contingency Plans

Risk management should be included in the planning to implement a quality management system in blood banks, given that it makes it possible to identify, characterize, and quantify the potential problems associated with the processes developed to classify them afterwards according to their importance and to elaborate an action plan for each of them. The aim of this work was to weight the use of this tool in process planning management, being able to extend its use from the blood bank to the remainder of the area of expertise. A matrix of Failure Mode and Effects Analysis (FMEA) was produced from the impact or severity of the failure, the probability of occurrence, and the detection capacity. The information obtained from the analysis allowed the calculation of the Risk Priority Number (RPN) whose value indicates if the process is under control or if some measures must be incorporated to keep the process under control. The results obtained have demonstrated that knowing the Risk Priority Number of each activity in a planning stage favours the design of a preventive and corrective action plan in advance.

6.2 Principal Learning Outcomes

In this report, a blood bank management app has been developed for the ones who are in need of blood.

We achieved an accuracy of on our dataset.

CONCLUSION AND FUTURE SCOPE

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming but also about all handling procedures related with blood banks. It also provides knowledge about the latest technologies used in developing android applications that will be in great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

This project has given us an ample opportunity to design, code, test and implement an application. This has helped in putting into practice various Software Engineering principles and Database Management concepts like maintaining integrity and consistency of data. Further, this has helped us to learn more about XAMARIN, C-SHARP, SQL, .NET.

We thank our guide for her invaluable contribution in guiding us throughout the project. We also thank our parents and friends who have supported and motivated us to complete this project successfully.

This paper explained the proposed Blood bank system, which linked the blood bank with the donors by sending messages to the donor who registered in the blood bank as a constant donor.

To inform them of a shortfall in one of the blood groups both by his platoon. The application used by the blood bank employee through smartphones. It is characterized by ease of use in organizing blood donation.

REFERENCES

1. “Blood Bank Management System” by Dr. Ajit Pal Singh.
https://www.researchgate.net/publication/358983452_blood-bank-management-system_July_2021_7471122569_6713106
2. “Blood Donation Management System” by Devanjan K. Srivastava, Utkarsh Tanwar, M.G.Krishna Rao, Priya Manohar, Balraj Singh.
<https://www.ijcrt.org/papers/IJCRT2105420.pdf>
3. “A Blood Donor Management System Web Application” by Sankar S, Tharunkumar S, Udhayakumar S, Vijay S.
<http://jjircce.com/admin/main/storage/app/pdf/tFUQEfqf2oFaLCbdI RPLJXFgaM9zE2BiDzxtNT5P.pdf>
4. “Online Blood Bank Management System” by Mohd Saif Abbas, Sheenu Rizvi.
https://www.researchgate.net/publication/362935450_Online_Blood_Bank_Management_System
5. “Cloud based online blood bank management system” by Aditya S.Iyer, Dr. C Menaka, Adnan Faisal, Ammar Hussain, Chethan S.D
<https://www.jetir.org/papers/JETIRFM06017.pdf>