

LIFE LINE

A Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

SHUBHAM KHARAT 3035213

SAKSHI MOKAL 3035256

DAKSHATA SURVE 3035419

SHEKHAR SARDESAI 3035345

Under the esteemed guidance of

Ms. Farzin Qureshi.



DEPARTMENT OF INFORMATION TECHNOLOGY
ELPHINSTONE COLLEGE OF ARTS SCIENCE AND COMMERCE

(Affiliated to University of Mumbai)

THIRD YEAR PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PNR No.: 2016016401070793.

Roll no: 2016129

Seat No.: 3035213

1. Name of the Student : SHUBHAM . KHARAT

2. Title of the Project : LIFE LINE

3. Name of the Guide : PROF. FARZIN. QURESHI

4. Is this your first submission? Yes

Signature of the Student

Signature of the Guide

Date: 22/4/19.....

Date: 22/4/19.....

Signature of the Coordinator

HOD B.Sc. (I.T)

Elphinstone College, Mumbai

Date:



MUMBAI, 400032

MAHARASHTRA

ELPHINSTONE COLLEGE OF ARTS SCIENCE AND COMMERCE

(Affiliated to University of Mumbai)

MUMBAI-MAHARASHTRA-400072

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled, "LIFE LINE", is bonafied work of **Shubham Kharat, Shekhar Sardesai, Sakshi Mokal, Dakshata Surve** bearing Seat No: 3035213, 3035345, 3035256, 3035419 submitted in partial fulfilment of the requirements for the honor of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

A handwritten signature in black ink, appearing to read "Amesha 22/4/19".
Internal Guide

A handwritten signature in black ink, appearing to read "N. Mehla 25/4/19".
External Examiner

A handwritten signature in blue ink, appearing to read "Ms. Sagdeo".
Coordinator

HOD B.Sc. (I.T)
Elphinstone College, Mumbai



ABSTRACT

"Blood bank organization system" is a web application that allows to access the whole information about blood bank management software, readily scalable and flexible to meet the complex need of blood bank who is the key facilitator for the health care sector; it also supports all the functionalities of blood bank. It is used for maintaining information about the campus. The project includes three main modules admin, donor, and acceptors. The admin module focuses on both the admin and acceptors. Unique user id and password is given to donor and acceptor, which identifies them uniquely. The member has to fill a login form to enter the login details.

Each member in the staff can change the password; query on a particular blood group and information on 'Why to donate blood' The whole project makes work easier for every person using the software in their required category.

ACKNOWLEDGEMENT

Victory is never achieved single handed so, it is our duty to acknowledge all those who have provided a helping hands in making this project success. We grab this opportunity to convey our enormous regards to wards all the notable people who have their valuable contribution in the hour of need.

We deeply thank our principal Prof. **Prof. MADHURI KAGALKAR** for giving us support throughout the course and made us capable of being worthy of recognition and extended query facility to us for making and computing this project smoothly.

We would like to express our genuine thanks to Prof. **Sagdev Mam (HOD)** (Head of IT Department) for his constant encouragement, which made this project a success.

We are highly obligated to Prof. **Farzin Qureshi** teaching staff member of the Information Technology department, who spread efforts in making the project a successful one with her guidance, appropriate advice and encouragement and of course the inspiration without which the project would be ineffective difficult task.

We honestly thank and express our deep gratitude to our teachers for their timely and important guidance. We also thank our Family members for their continued support in completing this project work and last but not the least; we wish to thank all my friends and well-wishers who are directly or indirectly connected with the success of our Project.

DECLARATION

We hereby declare that the project entitled, "Life Line" done at place where the project is done, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of our knowledge, no one has submitted to any other university. The project is done in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY) to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student

Shubham. P. Kharat.


Tables of Contents

CHAPTER 1: INTRODUCTION -----

1.1 Background -----
1.2 Objectives -----
1.3 Purpose, Scope, and Applicability -----
1.3.1 Purpose -----
1.3.2 Scope -----
1.3.3 Applicability -----
1.4 Achievements -----
1.5 Organization of Report -----

CHAPTER 2: SURVEY OF TECHNOLOGIES -----

CHAPTER 3: REQUIREMENTS AND ANALYSIS -----

3.1 Problem Definition -----
3.2 Requirements Specification -----
3.3 Planning and Scheduling -----
3.4 Software and Hardware Requirements -----
3.5 Preliminary Product Description -----
3.6 Conceptual Models -----

CHAPTER 4: SYSTEM DESIGN -----

4.1 Basic Modules -----
4.2 Data Design -----
4.2.1 Schema Design -----
4.2.2 Data Integrity and Constraints -----
4.3 Procedural Design -----
4.3.1 Logic Diagrams -----
4.3.2 Data Structures -----
4.3.3 Algorithms Design -----
4.4 User interface design -----
4.5 Security Issues -----
4.6 Test Cases Design -----

Chapter 5:Implementation and Testing

5.1 : Coding Efficiency
5.2: Testing Approach
5.2.1 :Unit Testing
5.2.2: Integrated Testing

5.3 Test cases

Capter 6: Results And Discussion

6.1 Test Reports

Chapter 7: Conclusion

LIST OF FIGURES

This section contains the list of figures which is appendice B of project.

1.1 Waterfall Model.....	1
1.2 Iterative Model.....	1
1.3 Zero Level DFD.....	1
1.4 Level 1 DFD	1
1.5 Level 2 DFD.....	1
1.6 Schema design	1
1.7 Flow Chart.....	1
1.8 ER Diagram 1.....	1
1.9 ER Diagram 2.....	1
2.0 ER of event.....	1
2.1 Use Case.....	1
2.2 Activity.....	1
2.3 Class Diagram	1
1.0 Backup.....	1

These figures are used to show the flow of the system and its working.

1.1 Waterfall Model.....

1.2 Iterative Model.....

1.3 Zero Level DFD.....

1.4 Level 1 DFD

1.5 Level 2 DFD.....

1.6 Schema design

1.7 Flow Chart.....

1.8 ER Diagram 1.....

1.9 ER Diagram 2.....

2.0 ER of event.....

2.1 Use Case.....

2.2 Activity.....

2.3 Class Diagram

1.0 Backup.....

INTRODUCTION

The residents of the world is rising with each approaching year and so are the diseases and health issues. With an increase in the people there is an increase in the need of blood.

The rising population of the world results in a lot of potential blood donors. But in spite of this not more than 10% of the total world population joins in blood donation.

With the growing population and the development in medical science the demand for blood has also increased. Due to the absence of communication between the blood donors and the blood recipients, most of the patients in need of blood do not get blood on time and hence lose their lives. organization is needed between the blood donors and hospitals and the blood banks. This inappropriate management of blood leads to wastage of the available blood inventory. Improper communication and organization between the blood banks and hospitals leads to wastage of the blood available. These glitches can be dealt with by industrializing the existing manual blood bank managing system. A high-end, effective, 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.

1.1 Background

In current system the blood bank management system exhibited at a lot of ineffectiveness and inefficiency that had fetched impact taken by organization.

The system which was physical that is based on paper card to collect blood donor data, keep record of blood donors and distribute results to blood donors, had weakness that needed IT based solutions. The system was categorized by delays and sometimes failure to access historical records; errors were viewed in entry and manual examination of results, secrecy and confidentiality of records lacked because unauthorized persons could easily access the records.

Disadvantages:

Without mixing the blood banks will lead to time consuming while searching of a particular group of blood. Without having accurate information, it is very difficult to supply the blood to the required people. Data sharing is not possible among blood banks, hospitals about the required blood group in the case of urgency.

PROPOSED SYSTEM:

The projected Blood Bank management system will help the people who are in need of a blood by giving them all details of blood group obtainability or about the donors with the same blood group. Our website work 24x7 so user can get info of blood donor any time. Blood donor can also get registered and save life of other individual. When blood is need in the process then people have very less time to get the blood available so if he gets the info like who can give him blood in time in his city is lifesaving.

Benefits:

Rapidity and accuracy there is no idleness of data.

It will be easily handle.

The proposed method is very easy to operate.

Reduce the Time spend on the paper work.

1.2 Objectives: The main objective of the project is to achieve the specifics of the blood bank blood group, donor, blood stock, blood cells, blood platelets, plasma etc.

The project is completely built at managerial end so the administrator is gauranteed the access of the system. The purpose of project is to build an application program to reduce the manual work for managing the blood bank, blood group, record, donor.

To generate a system that provides functions to support donors to view and manage their information easily. To maintain records of blood donors, blood donation information and blood stocks into database system. To inform donors about their blood result after their blood test.

It tracks all the details about the donor , blood stock ,blood cells etc.

1.3 Purpose and Scope

1.3.1 Purpose :

Blood Bank targets helping for social welfare. We have all the info; you will ever need.

Numerous people are here for you, to help you, prepared to donate blood for you anytime.

We have done all the job, rest is yours. search the blood group you need.

You can help us by recording on Blood Bank if you are willing to donate your blood when needed. As a pleased member of Blood Bank and a responsible human being, you can help someone in need. So one must donate blood to help the one who is in need.

1.3.2 Scope of the project:

Our plan aims to automate various processes of blood management system. It helps to collect flawless info of the users and manage the information perfectly. Due to computerization we can collect more amount of blood samples in very short period of time. The collection process will go on smoothly as it is online.

The Blood Bank aims serving for human welfare. We have all the information; you will ever need. Many people are here for you, to help you, willing to donate blood for you anytime. We have done all the job, rest is yours. search the blood group you need.

You can help us by registering on Online Blood Bank if you are willing to donate your blood when needed. As a proud member of Blood Bank and a responsible human being, you can help someone in need. So donate blood and save life.

1.3.3 Applicability

This organization is valid in blood banks and useful to hospitals and users too. The users will be able to view donors and lifesaving contact registered in the blood bank and this system will also help the user to search hospitals in desired location. Mainly this system is applicable for blood banks which are not yet online and those which are working manually.

As this system will reduce their efforts and side by side it will save nature too by boycotting papers, and by going digital.

1.4 Achievements

We gained a lot of knowledge about blood-banks and blood samples and about the procedure of the blood banks, Why the blood is paid? All this information we got while research and development about the project.

All this information will be conveyed to the users too via the project, so the people will be known to various aspects of blood and blood banks Rare blood groups like “Bombay blood” groups information is also conveyed.

The goal is to convey all the gathered information to the users and too “save one’s life” and the moto of the project is also the same.

1.5 Organization of Report

FEASIBILITY STUDY:

This idea is on Life Line Web App, it is highly feasible. It achieves all the feasibility desires to develop a good software. Feasibility study is done so that an impractical system is recognized early in description phase.

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the discoverer expect the future of the project and the helpfulness. A Feasibility study of a purposed system proposal is according to its workability, which is the effect on organization, ability to meet their user needs and effective use of resources. Thus when a new application is purposed it normally goes through a Feasibility study before it is accepted for the development.

TECHNOLOGY FEASIBILITY:

1. It is run on machine platform with least configuration.
2. A study of function, performance and limitations that may affect the ability to achieve an acceptable system.
3. Being a small scale project the working of the project is quietly simple.
4. The project is user friendly and so this new technology will save a lot of time of both the Data Entry Worker and the Management. The flow of the project is not complicated. system, the examination must on to suggest the type of equipment, required method developing the system, of running the system once it has been intended.

Technical issues raised during the investigation are :

1. Does the present technology is enough for the proposed one?
2. Can the system expand if developed?

The project should be developed such that the needed functions and performance are achieved within the limitations. The project is developed within latest technology. Through the knowledge may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may be still used. So there are minimal constraints involved with this project. The system has been developed using Asp.net the product is technically Feasible for improvement.

ECONOMICAL FEASIBILITY:

1. An valuation of development cost weighted contrary to the ultimate income or benefit resulting from the development system.
2. System build with least resource & low cost.
3. As it is not a large scale project it is not expensive.

The emerging system must be justified by cost and benefit. Standards to ensure that effort is focused on project, which will give best, return at the earliest. One of the factor, which affect the development of a new system, is the cost it would REQUIRE.

Chapter 2

Survey of Technologies

Front End Technology :

Technologies:

What does ASP.NET mean?

ASP.NET is a combined web development model integrated with .NET framework, intended to provide services to generate dynamic web applications and web services. It is constructed on the Common Language Runtime (CLR) of the .NET framework and includes those benefits like multi-language interoperability, type security, trash collection and inheritance.

It was created to abridge the growth of distributed applications in organized and object-oriented manner by separating the presentation and content then hence write clean code. ASP.NET uses the code-behind model to produce dynamic pages based on Model-View-Controller architecture.

ASP.NET works with the Internet Data Server (IDS) to deliver the content in answer to client requests. ASP.NET delivers access to all .NET classes, custom mechanisms and databases, similar to that of a desktop application while processing the requests.

Reason to use ASP.net in project:-

- Fewer coding is obligatory for application development in ASP.NET.
- Just in time collecting, early binding and storing services are available right out of the box.
- Large library of built-in classes can be reused.
- This declines the sum of coding required.

- ASP.NET supports common language runtime. Later, it can support several languages, such as C#, VB, etc.
- Presentation is fast. The ASP application is located in the server in compiled form. This reduces response time.
- Windows confirmation and form authentication are the safety features which are built right in.
- You can mix ASP.NET with ADO.NET. This updates database functionalities for websites.
- ASP.NET splits presentation and programming logic with the use of code-behind programs.

Back End Technologies:

MySQL:

MySQL is the highest widespread Open Foundation Relational SQL Database Management System. MySQL is one of the finest RDBMS being used for emerging various web-based software applications. MySQL runs on virtually all platforms, including Linux Unix and Windows. Though it can be used in a wide range of applications, MySQL is most often associated with web applications and online broadcasting.

How MySQL works:

MySQL is put up on a client server model. The MySQL server is the core of MySQL, which handles all of the database orders and commands.

Reason to choose MySQL:

MySQL is Secure: It uses secure data security layers to protect delicate data from outside access. Distinct rights can be set to provide multiple user levels, letting several people access to stored data. All keys used to access the MySQL database are encoded.

MySQL is Cheap: Provisionally the MySQL database software is free to download from the certified portal, and comes included pre-installed on most hosting servers it's not luxurious to use or maintain.

MySQL is Heavily Supported: You can run the software on a variety of platforms and operating systems including Windows, Linux, UNIX etc.

MySQL is Fast: When related to other database software like Sybase and Oracle, MySQL has fewer features to offer but this is good. It means that MySQL is much faster in terms of performance, and users still have access to all the necessary features they would through other database platforms.

MySQL Data Storage is Scalable: By default, MySQL can handle a total file size limit of about 4GB, though, this can be increased to meet your needs – by as much as 8TB of data.

LANGUAGE:

C# (C Sharp):

C# is a general object-oriented programming (OOP) language for networking and Web development. C# is specified as a common language infrastructure (CLI) language.

You can use C# to create Windows user applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, suitable user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

- Summarized method signatures called *delegates*, which enable type-safe event notifications.
- Properties, which serve as accessory for private member variables.
- Attributes, which provide declarative metadata about types at run time.

• Benefit of using C# :

- It's an Object Oriented language, which is a style of programming that carries a lot of its own advantages.
- More Creative
- Adaptability
- The most powerful programming language for the .NET Framework, with the help of Visual C++ and a redesigned common language runtime (CLR), a virtual machine component that executes all programs written for .NET
- Designed by Microsoft, so there's plenty of documentation and support

The above mention technologies and tools are used to develop the web based application, distributed application etc. During the survey of some interior application we studied the procedure and techniques, tools used by these application which exist in market today.

These technologies are easy to implement, user friendly, accessible from anywhere. Due to these feature they are used a lot in web application.

Chapter 3

Requirement Analysis:

Requirement Investigation, also called ‘requirement engineering’. Requirement Analysis is the procedure of determining user expectations for a new or modified product. In software engineering, such requirement are often called functional specifications. Requirement Analysis is an important aspect of project management.

Requirement investigation is well-defined as above but real conception of the requirement analysis is the software & hardware prerequisite which will check the availability of the specified hardware and software requirements. Which is first subpart under the constraint analysis.

Requirement Investigation is done after proper planning and documentation is done. So according to that list the requirements of the project are taken under the completion.

So one by one the requirement analysis are completed with the help of diagrams and many more things.

Requirement Analysis Subtopics are as follows:

1. Problem Definition.
2. Requirement Specification.
3. Planning & Scheduling.
4. Software Requirements
5. Hardware Requirements.
6. Preliminary Product Description.
7. Conceptual Models.

3.1 Problem definition:

The current process of blood bank management is manual , so it is tedious work and it requires a lot of man power and paper work.

There are rare blood groups like “Bombay Blood group” especially found in Mumbai and newly discovered due to manual process and less advancement in automation this information is not reached to the local people .Unavailability of blood in emergency cases .Less awareness among people about blood donation and blood banks .Deaths due to lack of blood during Major operations .

3.2 Requirement Specification

Admin

This is a super user who has all access to the blood system can prosper or remove profiles, update information, reserve database etc.

The following are the roles of Admin

1. Preserve donor details
2. Preserve acceptor details
3. Bring up-to-date donor details
4. Preserve acceptor details

Donors

Donor can be a free person or also can be the general public, blood bank, group etc. Each member in a donor is given a user id and password which recognizes him uniquely. This member is given a login form and following are the choices given to donors

1. Alteration password
2. Find a blood group
3. Change status
4. Maintain his History
5. Why donate blood
6. Logout

Acceptor

The following are the facilities will be provided to the acceptors.

1. Change password
2. Find and search for a blood
3. Who needs blood
4. Logout

3.3 Planning and Scheduling

Planning is the worst part of project management. As things doesn't go according to the plan. An Experienced Project Manager can calculate the proper estimate of time to complete the project. Planning is done by the Project manager and the project is divided into phases, Modules.

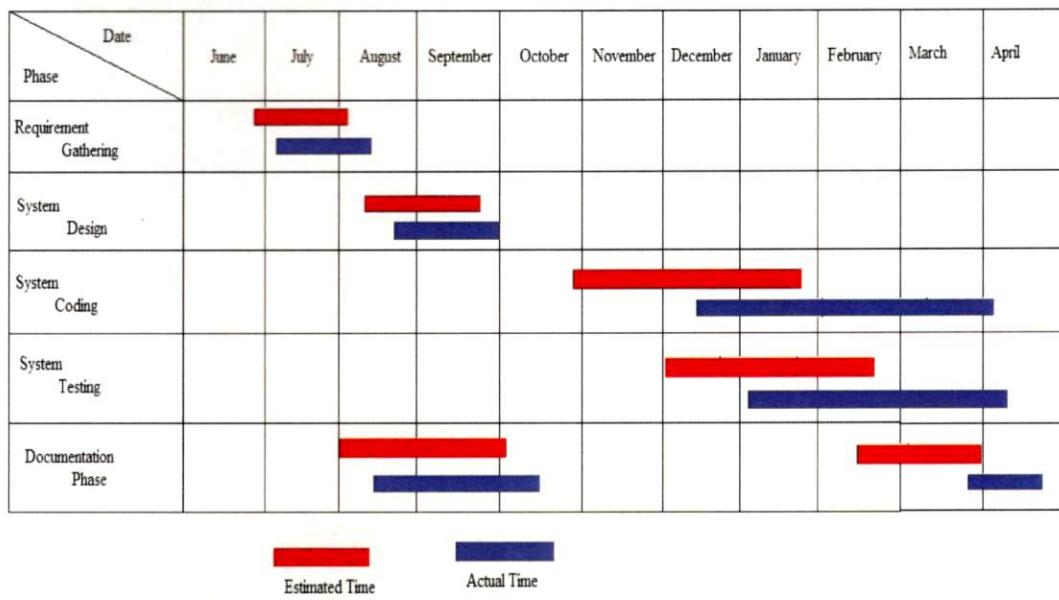
Phase 1- Introduction, SRS, System Analysis, System UML Design.

Phase 2- coding, Designing.

Phase 3- Testing.

The planning is Done according to the the phases Mentioned above:

August	- Project Selection
September	- Project overview , Information Gathering
October	- Documentation
November	- Final Phase 1 Documentation



3.4 System Requirements

Software/Hardware Requirement Specification:

Server Side:

- 512MB RAM or further.
- 500MB free space in Hard Disk.
- Windows XP or higher kind.
- Internet Connection

Minimum Client Requirement:

- 512MB RAM or more
- Windows XP or any other Operating System
- Internet Connection

Tools and Technology:

- **Tools:**
 - ASP.Net, MySQL, Bootstrap
 - Google Chrome or any Explorer.
 - Star UML for System design.
 - Paint, Adobe Photoshop

3.5 Preliminary product description

The Primary step in the system progress lifecycle is the introductory investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to estimate project requests. it is not a project study nor does it include the collection of details to describe the business system in all respect. Rather it is the collecting of information that helps committee members to evaluate the merits of the project request plus make an informed judgment about the feasibility of the proposed project.

Specialists working on the preliminary investigation should achieve the following objectives:

- * Explain and understand the project request
- * Control the size of the project.
- * Evaluate costs and benefits of alternative approaches.
- * Regulate the technical and operational feasibility of alternative approaches.
- * Testimony the findings to management, with recommendation outlining the acceptance or rejection of the proposal.

***Advantage to Organization**

The organization will apparently be able to gain benefits such as savings in working cost, reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill.

*** The Primary Cost**

The primary cost of setting up the system will include the cost of hardware software (OS (add - on software, utilities) & manual labor (setup & maintenance). The same has to bear by the organization.

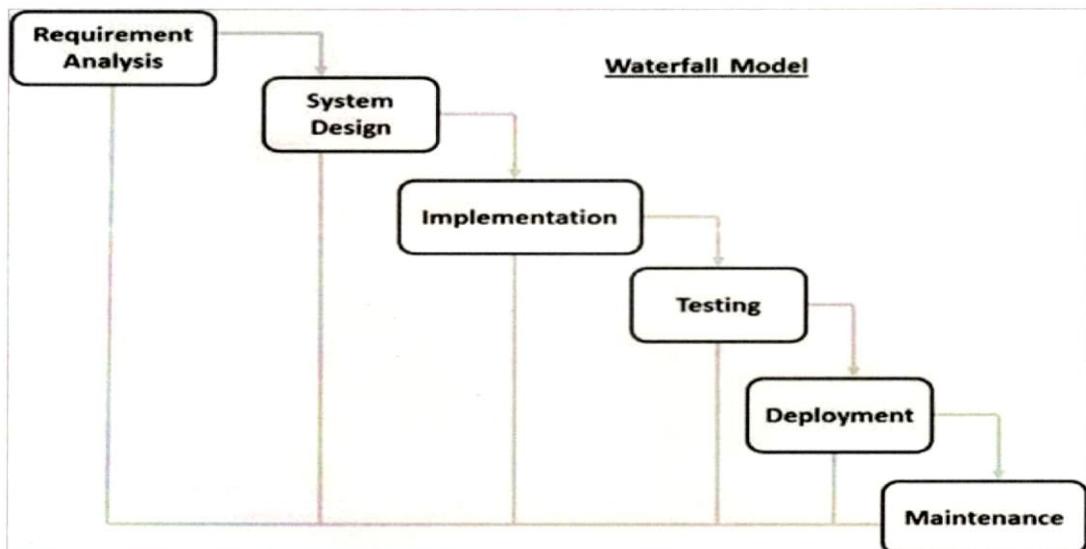
3.6 Conceptual Models.

Waterfall model.

The waterfall model is a sequential design process often used in development process in which progress is seen as flowing steadily downwards through the phases of :

- Requisite gathering: - Requirements are gathered in this phase. The overall functional requirements are collected as well as nonfunctional requirements are also considered.
- Design: -The design phase consists of evaluating whole design of the system by moving through each phase of the proposed system drawing diagrams, data models, etc.
- Implementation/Coding: - the actual development starts in this phase. The approved system is implemented in actual and resources are used according to the timeline.
- Integration/ Unit testing: - Each and every module is tested for its basic functionality and overall system is evaluated.
- Deployment: - The system is deployed to the user for practical use.
- Maintenance: - The implemented system is maintained in this phase with user communication.

Figure 1.1 WATERFALL MODEL



Iterative Model.

In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

The Iterative models working according to our project Android Game Application.

The Iterative Models are processed as follows:

- Requirements of our project.
- Building of the project.
- Design of Android Game Application.
- Testing the modules and finding the bugs, from the modules how much we are done.
- Implementation of our Android Game Application is still under the process.

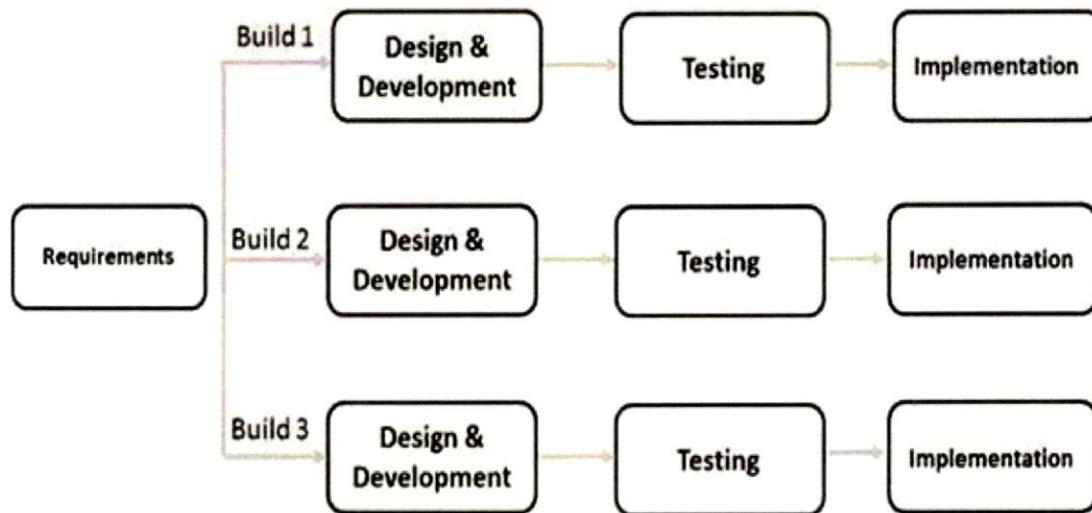


Figure 1.2 Iterative Model

Chapter 4

Basic Modules

4.1 MODULES OF BLOOD BANK MANAGEMENT SYSTEM:

Blood Bank management system is to deliver services for the people who are in need of blood by getting help from the donors who are interested in donating blood to the individuals. There are seven core modules in this organization.

Admin

Donors

Donor Registration

Modifying Donor Information

Acceptors

Donor Search

Life Saving Contacts

Admin:

Admin can succeed both donors & acceptors. He can add or remove any user from the organization. Each associate in a donor & acceptor is given a user id and password, which detects him exceptionally. As of admin module can change donor details, delete donor or change the password.

- Change Password
- Alter donor details
- remove donor details
- Logout

On every occasion a user wants to change his / her password he can select the change password option. The system displays the form, which asks him for his old password and new password. The organization then matches the old password with the current password in the database and if they match then the password is set to the new password in the record.

Donors

As of this module user can create their account, when user create his/her account the user get a user id and password, which recognizes him uniquely. From this module user can search donor for blood and can also refer his friend to become a donor. Donor can also get information like when he donated blood or when he will be able to donate blood.

Donor Registration:

In this section, people who are interested in donating blood get registered in my site and give his overall details related to him, i.e. he fills in a registration form by giving the total details such as name, address, city, sex, dob, blood group, telephone numbers, e-mail address, etc. He was also given two fields' username and password to fill such that he was a registered donor and he can enter the login form with his username and password and can modify his details if needed.

Modifying Donor Information:

The listed donor only is able to modify his details, no other person can alter his details as there was a login form which restricts others from entering the username and password providing high security for the details given by the donor. If the donor wants to modify his details, he was forced to give his username and password to enter in. After giving the username and password it checks for the donor whether he is an existing donor or not and if the username and key matches, he can then able to modify his total details. If the username and password do not exist then he gets a message as 'Improper ID and Key Entered, Try Again'.

Resulting links are available on donor and acceptor module.

- Why donate blood
- Who needs blood
- Find A Donor.
- Change password.
- Logout

Acceptors

This building block helps user to find blood group. When user click on find a blood group system ask him to enter blood group he want to search. Later entering the blood group, system search for the availability of the blood group and give him the list of the donors who has the same blood group. Every time a user wants to change password he can select the update password option. Then system ask the user to enter old username and password then system check the identifications and change the password.

- Find A Donor.
- Refer A Friend.
- Change password.
- Find a Blood group.
- Logout

Donor Search:

The individuals who are in need of blood can search in our site for getting the details of donors having the same blood group and within the same city. They can straight click on the link search a donor and can select a city name as well as the blood group which he needs. He then gets the details of the donors who exist within the city and the same blood group that he has selected. If no match was are found for the city and group selected by him he gets a message '*SORRY DONORS ARE NOT AVAILABE WITH THE SUBSEQUENT BLOOD GROUP AND AREA*'.

Life Saving Contacts:

If at all the public in search of a donor doesn't get any match for their area and group then they will be provided a service i.e. he will be given a Contact Person details for their nearby cities who have the details of many other donors with him. The individuals in search can call him and can get the details of the donors and can be delivered facilities in this method. But this life saving contact persons can be accessible only for a limited number of cities but not for all. These contact persons are the sanctioned persons of my blood bank.

4.2 DataDesign

zero level dfd :

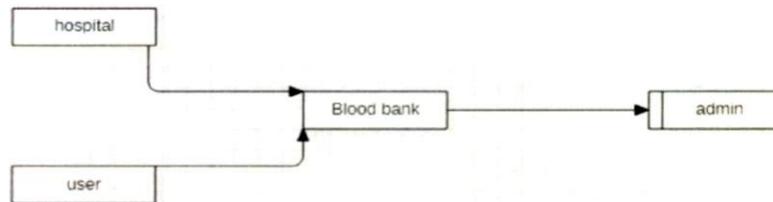


Figure 1.3 Zero Level DFD

DFD LEVEL 1

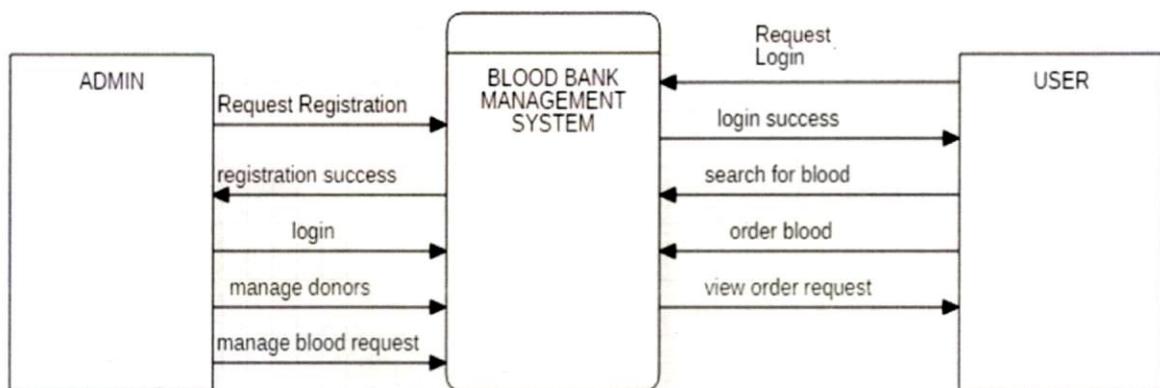
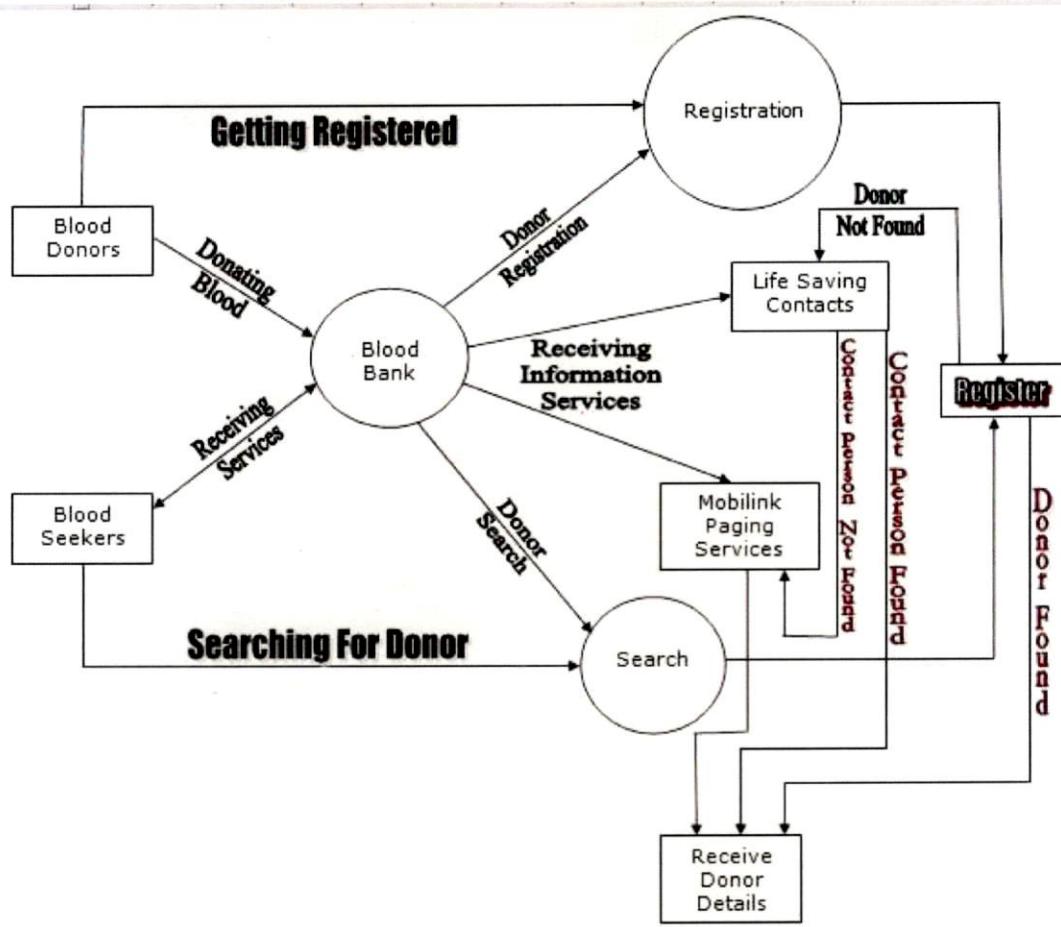
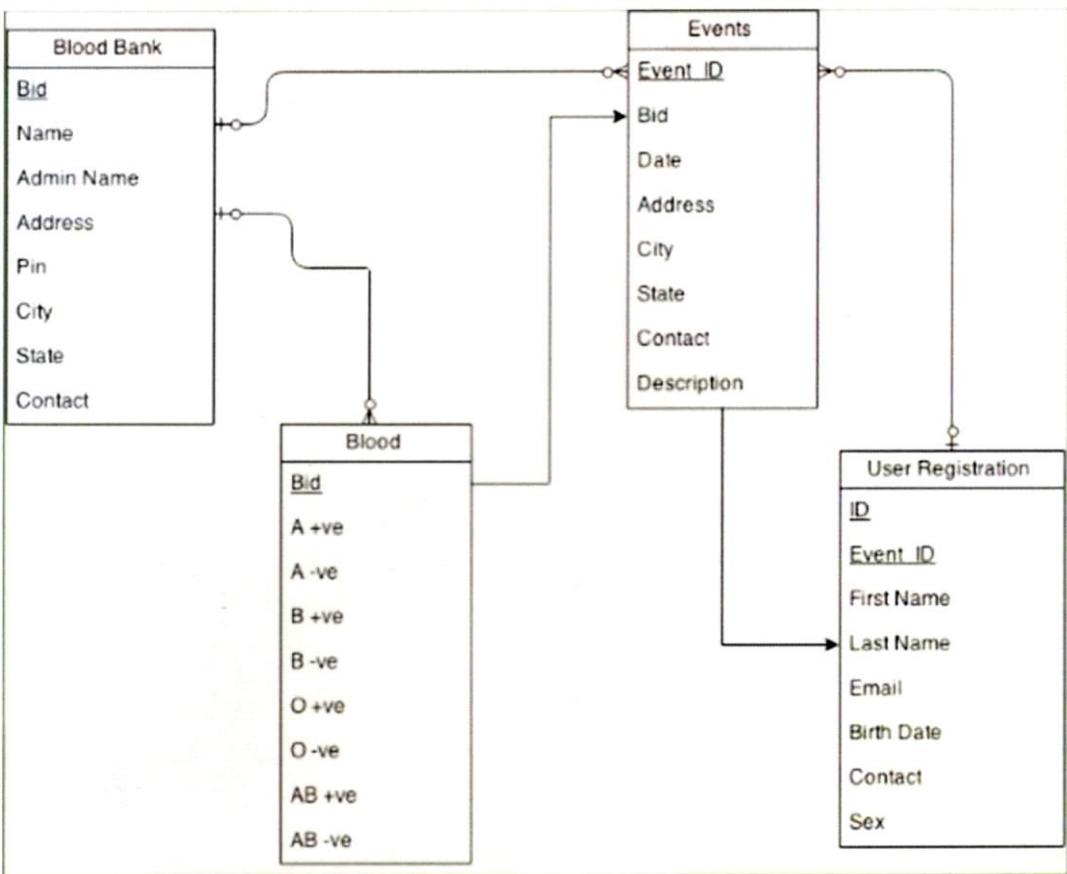


Figure 1.4 Level One DFD



Schema Design:



1.6 Figure Schema Data Base Diagram

4.2.2 Data Integrity and Constraints

Admin:

Column name	Data type	Length
Admin id	Int	10
Admin Name	Varchar	20

Hospital:

Column name	Data Type	Length
Hospital Id	Int	10
Hospital Name	Varchar	50

Donor:

Column name	Data type	Length
Donor id	Int	10
Donor Name	Varchar	20
Donor address	Varchar	100
Donor phone no	Int	10
Donor blood group	Varchar	10

User:

Column name	Data type	Length
User id	Int	10
User Name	Varchar	20
User address	Varchar	100
User phone no	Int	10
User blood group	Varchar	10

Blood camp event:

Column name	Data type	Length
P -id	int	10
P-name	Varchar	50
P-address	Varchar	100
P-phone no	int	10
P-Blood group	Varchar	10

Blood:

Column name	Data type	Length
blood id	Int	10
blood group	Varchar	20
Blood type	Varchar	10
Blood stock	Int	100

Blood Bank:

Column name	Data Type	Length
Blood Bank id	Int	10
Blood Bank Name	Varchar	50
Blood id	Int	10
Blood stock	Int	100
Blood Bank Contact	Int	10

Admin table consist:

Admin id: int (10) PRIMARY KEY.

Admin Name: varchar (20).

Hospital Table consist:

Hospital id: int (20).

Hospital Name: varchar (30).

Hospital Stock: int (20).

Donor Consist:
Donor id: int (20).
Donor name: varchar (50).
Donor address: varchar (50).

User consist:
Donor id: int (20).
Donor name: varchar (50).
Donor address: varchar (50).

Blood camp event consist:
Blood camp id: int (20).
Blood camp name: varchar (50).
Blood camp venue: varchar (50).
Blood camp date: int (20)

Blood consist:
Blood id: int (20).
Blood type: varchar (50).
Blood Stock: int (100).

Blood Bank:
Blood bank id: int (20).
Blood Bank name: varchar (50).
Blood Bank address: varchar (50).
Blood Bank Stock: int (100).

4.3 Procedural Design

4.3.1 Logic diagrams:

Flow Chart of System:

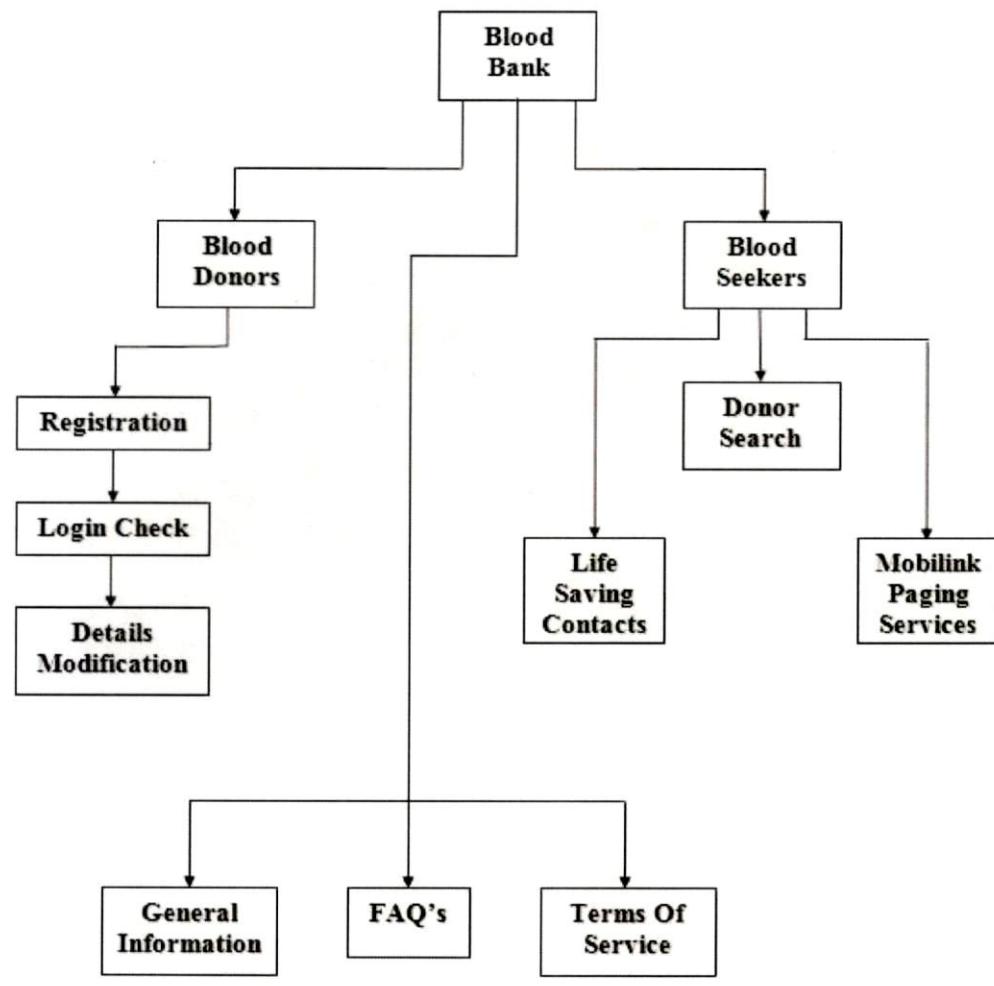


Figure 1.7 flow chart

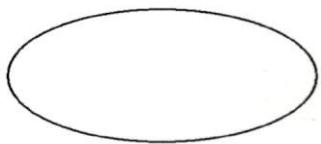
Entity Relationship Diagram

SYMBOL

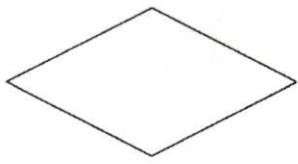


PURPOSE

Represents Entity sets

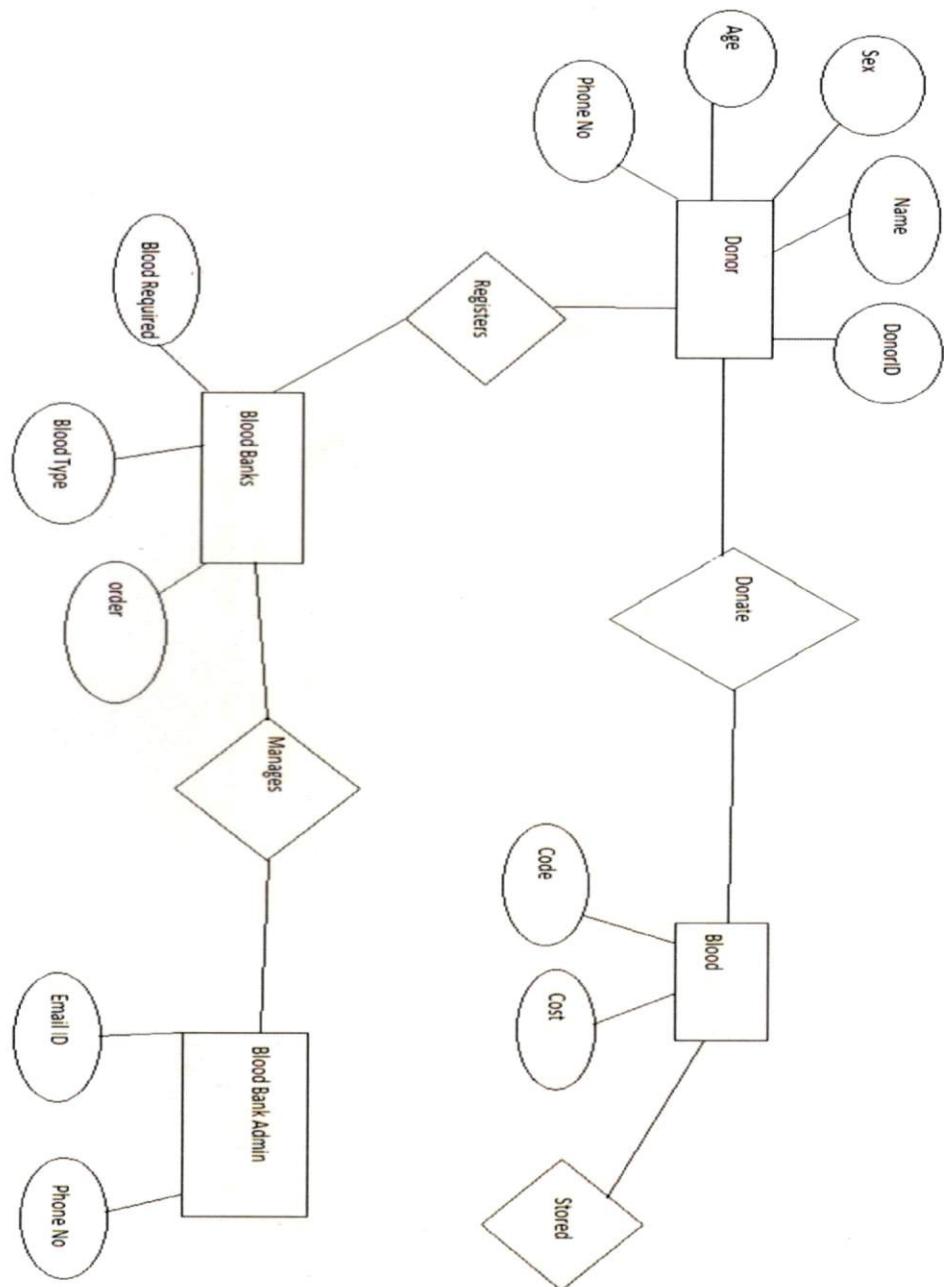


Represent attributes.

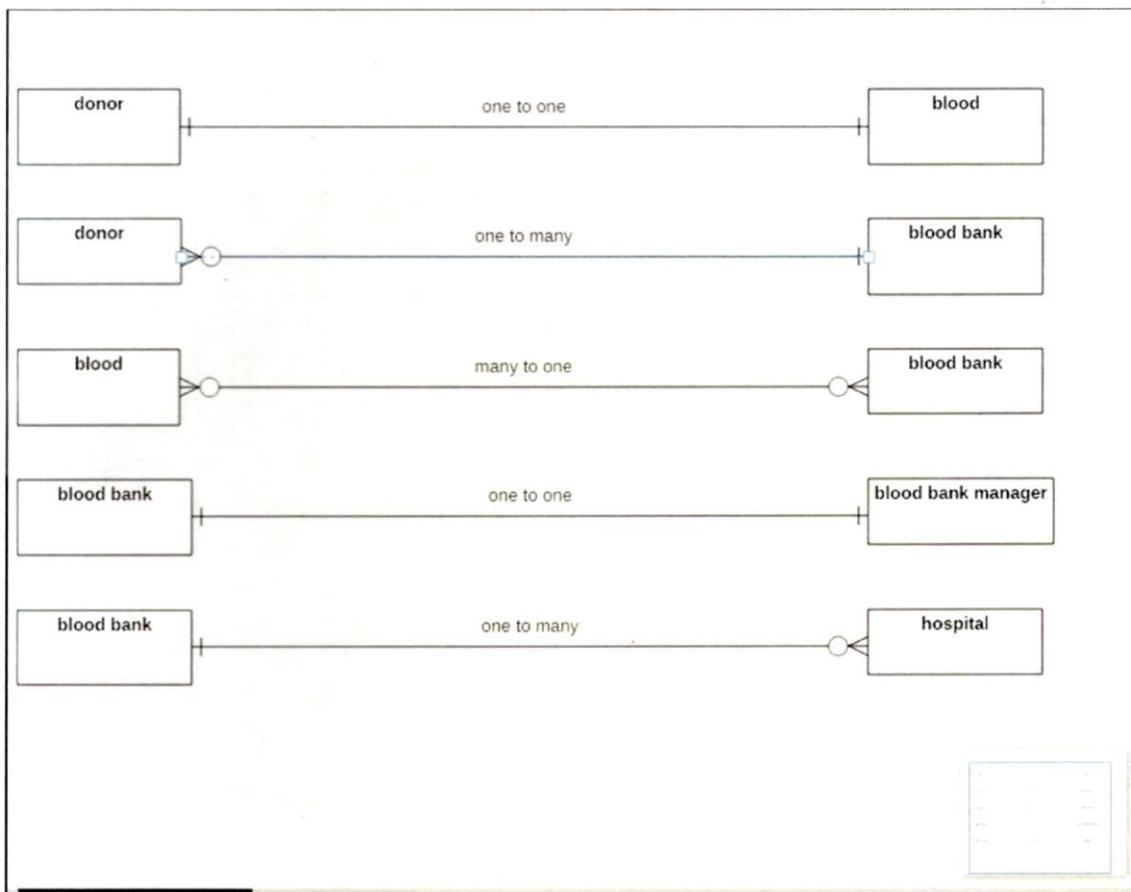


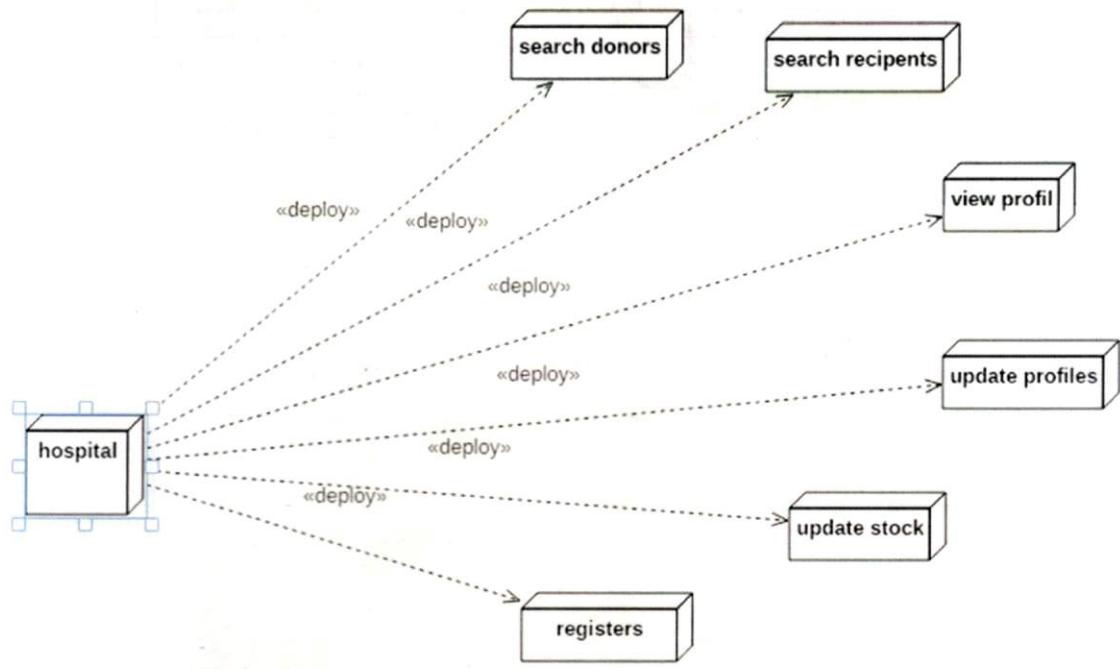
Represent Relationship Sets.

E-R Diagram Fig no 1.8

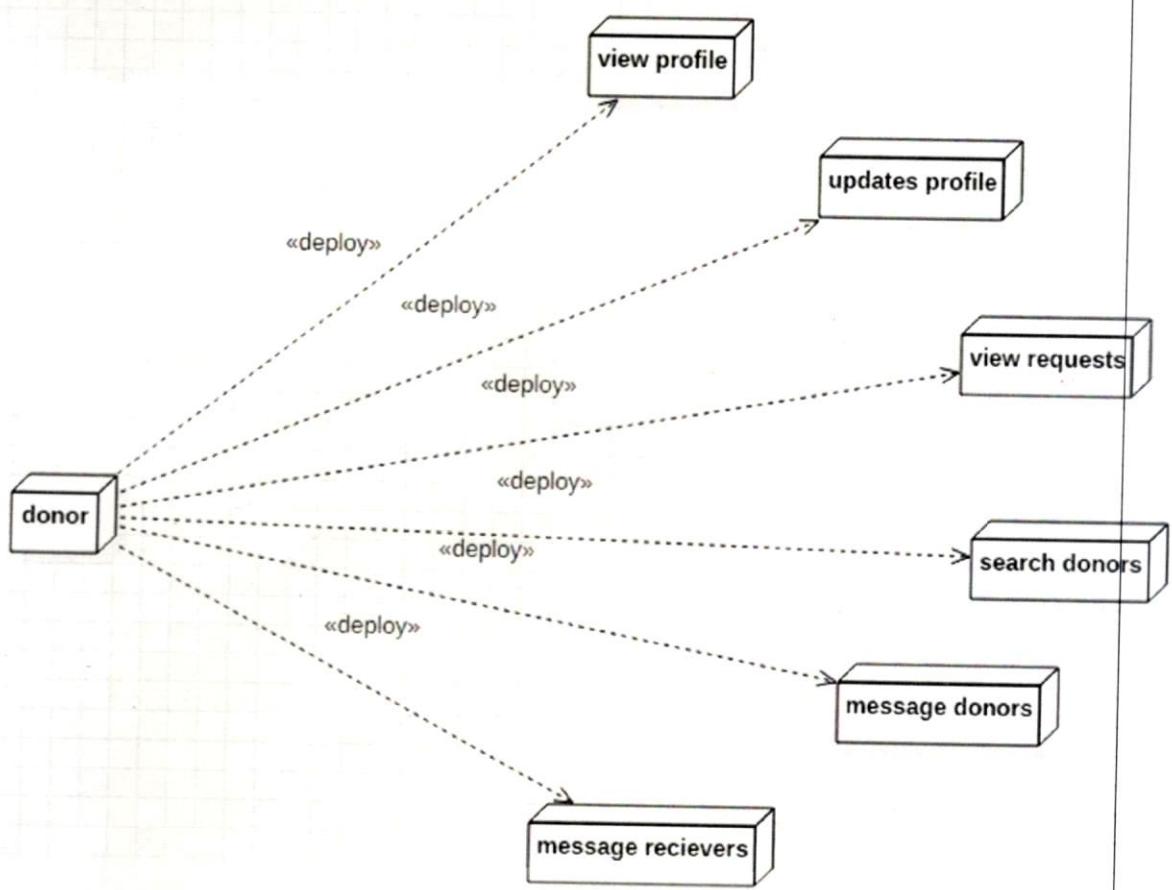


Entity Relationship Fig 1.9

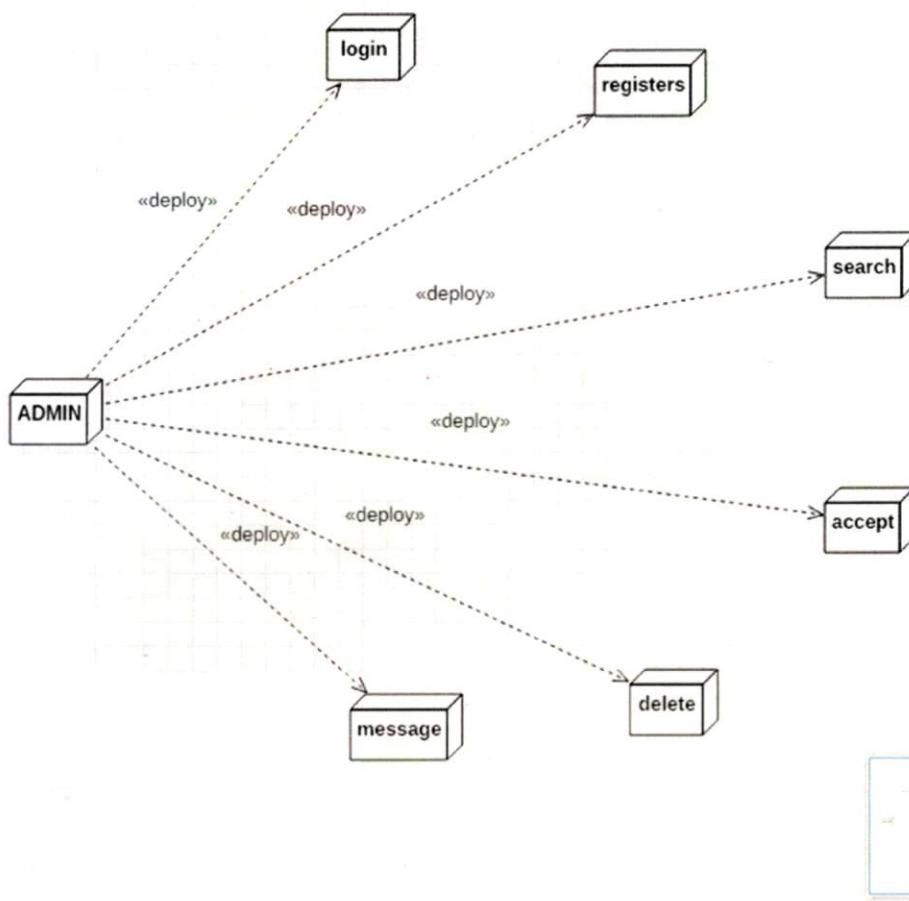




Hospitals Deployment Diagram figure 2.0

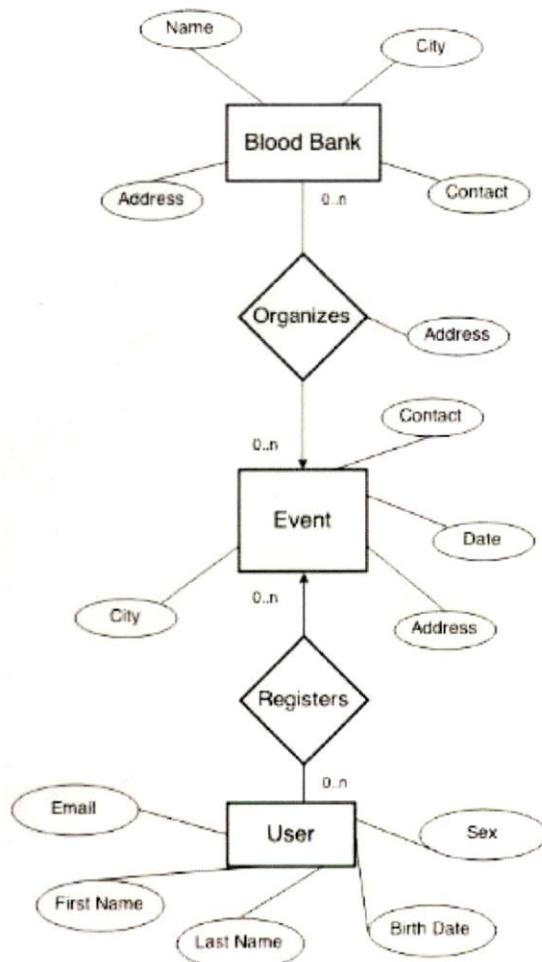


Donors Deployment Diagram figure 2.1



Admins Deployment Diagram Figure 2.2

Camping Events Diagram:



ER of Event Figure 2.3

Use case Diagram:

- **Use cases:**

A use case describes a sequence of action that provides something of measurable value to an action actor and is drawn as horizontal ellipses.

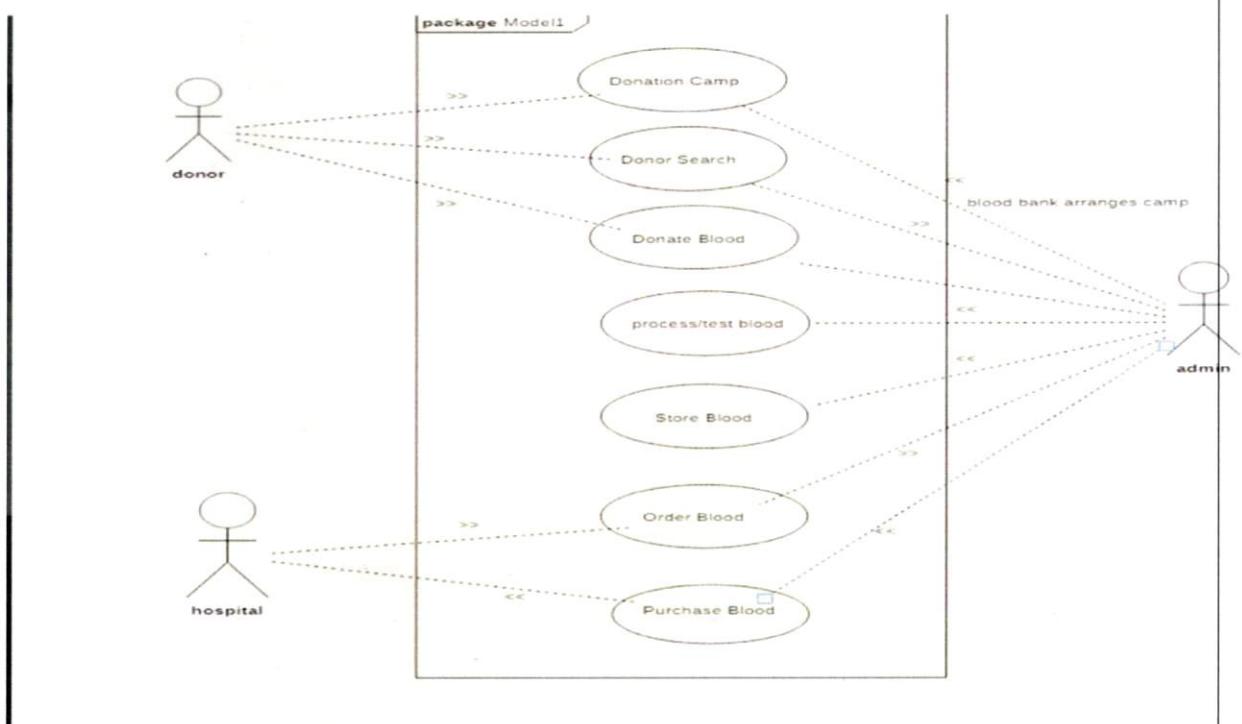
Notation:-



- **Actor:**

An actor is a person or organization external to the system that plays a role in more interaction with your system. Actor is drawn as stick figures.

Notation:-



USE CASE DIAGRAM Figure no 2.4

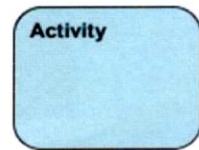
ACTIVITY DIAGRAM

Activity chart is another essential behavior diagram in uml diagram to describe dynamic features of the system. Activity diagram is basically an advanced version of flow chart that modeling the flow from one activity to another activity.

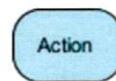
Notation Description

Activity
Is used to represent a set of actions

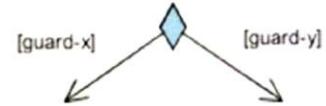
UML Notation



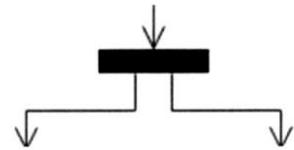
Action
A task to be performed



Decision Node
Represent a test condition



Fork Node
Split behavior into a set of parallel or concurrent flows of activities (or actions)



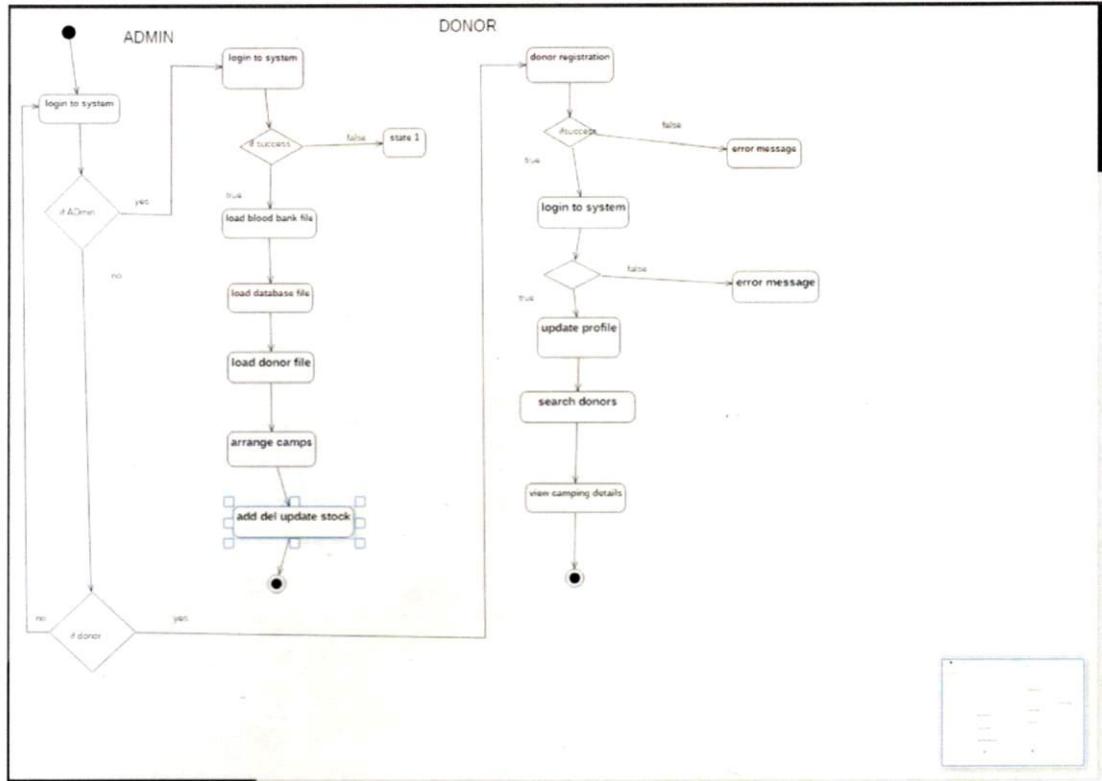


Figure 2.5 Activity Diagram

CLASS DIAGRAM

A class diagram contains of a collection of classes and boundaries reflecting important entities of the system, and the relationships between these classes and interfaces. Classes in a class diagram are unified in a classified fashion. Class Diagrams describe the static structure of a system, or how it is prepared rather than how it behaves.

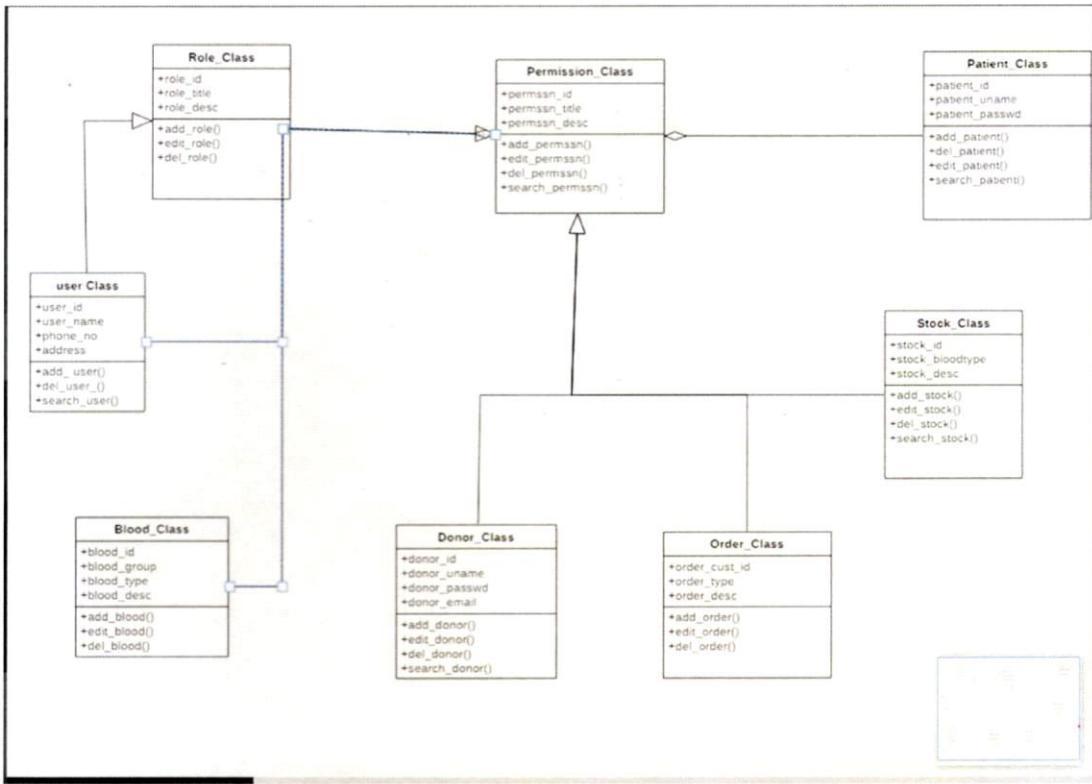


Figure 2.6 Class Diagram

4.5 SECURITY ISSUES

Security is a vital issue for current IT system. System manager and safety manager are accountable for safeguard's organization data and computing infrastructure.

Password Encryption: Whatsoever security actions taken, in any case if anyone with vicious or malicious mind gets access to the database he can temper the serious data. To limit it we've used encrypting algorithm facilitated to us by Microsoft ASP.Net.

Two procedures are there:

SHA1- It takes the key and encrypt it in a string of 40 characters

MD5- It takes the key and encrypt it in a string of 32characters

The loveliness of these Algorithm are that whatever be the length of password the length of converted string remains same.

Though, a question arises here that if this thing is that good then why didn't we used it for keeping each and every data????

Authorization:

Super Admin is certified to give permission to admin for the registration of the students.

Authentication:

Only listed members can access their account. Every registered member has different and unique User ID.

Information Integrity:

Lone Super admin has right in the change of the information. There is no possibility by the end user or illegal user of hacking the data.

Detection:

Super Admin will have a backup of the project. Also there is a compound and alphanumeric password for admin and student.

4.6 TEST CASE DESIGN

Event	Trigger	Source	Activity	Response	Destination
User/Register	Registration Details	users	Registration of users	Login page	users
User verification	verification	admin	Verification of users	Verification process	admin
User/login	Email id Password	User / admin	Login	User page	Users / admin
View Events	Events Details	users	Events Details	Events Page	users
View Notices	Notice Details	users	Notice Details	Notice Page	users
Gives Complains	Complains Details	users	Complains Details	Complain Page	users
Print Sticker	Sticker	users	Sticker	Parking Sticker Page	users
Calculate Bill	Bill Details	users	Bill Details	Bills Page	users
Print Receipt	Receipt	users	Receipt	Receipt pdf	users

Chapter 5

Coding Details and Code Efficiency :

Home page : This is my home page



Home page coding:

```
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
<div class="pogoSlider" id="js-main-slider">
    <div class="pogoSlider-slide" data-transition="verticalSlide" data-duration="2000" style="background-image:url(images/blood1.jpg);">
        <div class="pogoSlider-slide-element">
            <div class="container">
                <a href="#about" class="btn btn-circle scroll">
                    <i class="fa fa-hand-o-down" aria-hidden="true"></i>
                </a>
            </div>
        </div>
    </div>

    <div class="pogoSlider-slide" data-transition="verticalSlide" data-duration="2000" style="background-image:url(images/blood6.jpg);">
        <div class="pogoSlider-slide-element">
            <div class="container">
                <a href="#about" class="btn btn-circle scroll">
                    <i class="fa fa-hand-o-down" aria-hidden="true"></i>
                </a>
            </div>
        </div>
    </div>
</div>
```

```

<div class="pogoSlider-slide " data-transition="verticalSlide" data-duration="2000"
style="background-image:url(images/blood5.jpg);">
    <div class="pogoSlider-slide-element">
        <div class="container">
            <a href="#about" class="btn btn-circle scroll">
                <i class="fa fa-hand-o-down" aria-hidden="true"></i>
            </a>
        </div>
    </div>
</div>

<div class="pogoSlider-slide " data-transition="verticalSlide" data-duration="2000"
style="background-image:url(images/blood4.jpg);>

<div class="pogoSlider-slide-element">
    <div class="container">
        <a href="#about" class="btn btn-circle scroll">
            <i class="fa fa-hand-o-down" aria-hidden="true"></i>
        </a>
    </div>
</div>
</div><!-- //pogoSlider -->
</asp:Content>

```

Signup for donor :

The screenshot shows a web browser window with the URL localhost:51090/bloodbank/RegisterAsDonor.aspx. The page title is "Register As Donor". The form consists of several input fields:

Name :	<input type="text"/>
Address :	<input type="text"/>
Emailid :	<input type="text"/>
Phone No. :	<input type="text"/>
Blood Group:	Select <input type="button" value="▼"/>
Date of Birth :	<input type="text"/>
Upload Photo :	<input type="file"/> Choose File No file chosen
<input type="button" value="Register As Donor"/>	

At the bottom of the browser window, there are three tabs: "samruddhifinalne.pdf", "docm remain final.docx", and "samruddhifinalnew.pdf".

Coding of Signup.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/bloodbank.master"
AutoEventWireup="true" CodeFile="registerasdonor.aspx.cs" Inherits="registerasdonor"
%>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
<br><br><br><br><br><br><br><br>

<center>
<h3 class="title clr" style="background-color:green;color:white">Register As
Donor</h3>
<br>      <br><br><br>

          <form action="#" method="post">
          <center>
<table width="70%" style="background-color:white;color:green" border="2">
<tr>
          <td><b>Name :</b></td> <td> <asp:TextBox ID="txtName"
runat="server"></asp:TextBox>
          <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ErrorMessage="Please Enter your First name"
ControlToValidate="txtName"></asp:RequiredFieldValidator><br/>
          <asp:RegularExpressionValidator ID="RegularExpressionValidator2"
runat="server" ErrorMessage="Name contains Only Characters"
ControlToValidate="txtName" ValidationExpression="[a-zA-
Z'.\s]{1,50}"></asp:RegularExpressionValidator>      <br/>
          <td><b>Address :</b></td><td> <asp:TextBox ID="txtAddress"
runat="server"
          TextMode="Multiline"></asp:TextBox>
          <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ErrorMessage="Please Enter your address"
ControlToValidate="txtAddress"></asp:RequiredFieldValidator><br/>
          <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ErrorMessage="Address contains Only Characters and Numeric"
ControlToValidate="txtAddress" ValidationExpression="[a-zA-
Z'.\s]{1,50}"></asp:RegularExpressionValidator>      <br/>
          </td>
          <td><b>Emailid : </b></td><td> <asp:TextBox ID="txtEmailid"
runat="server"></asp:TextBox>
```

```

        <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server" ErrorMessage="Please Enter valid emailid"
ControlToValidate="txtEmailid"></asp:RequiredFieldValidator><br/>
        <asp:RegularExpressionValidator ID="RegularExpressionValidator7"
runat="server" ErrorMessage="Invalid EmailID" ControlToValidate="txtEmailid"
ValidationExpression="\w+([-.\'])\w+@\w+([-.\'])\w+\*\.\w+([-.
.]\w+)*"></asp:RegularExpressionValidator>      <br/>

<tr>
<td><b>Phone No :</b></td><td>
<asp:TextBox ID="txtPhoneno" runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server" ErrorMessage="Please Enter Valid Phoneno"
ControlToValidate="txtPhoneno"></asp:RequiredFieldValidator><br/>
        <asp:RegularExpressionValidator ID="RegularExpressionValidator6"
runat="server" ErrorMessage="Invalid Mobile No" ControlToValidate="txtPhoneno"
ValidationExpression="^([7-9]{1})([0-9]{9})$"></asp:RegularExpressionValidator><br/>
        <td><b>Blood Group:</b> </td><td>
        <asp:DropDownList ID="ddlbloodgroup" runat="server"
Height="35px" Width="125px">
        <asp:ListItem>Select</asp:ListItem>
        <asp:ListItem>A+</asp:ListItem>
        <asp:ListItem>A-</asp:ListItem>
        <asp:ListItem>B+</asp:ListItem>
        <asp:ListItem>AB+</asp:ListItem>
        <asp:ListItem>AB-</asp:ListItem>
        <asp:ListItem>O+</asp:ListItem>
        <asp:ListItem>O-</asp:ListItem>
        <asp:ListItem>B-</asp:ListItem>
    </asp:DropDownList>

<asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server"
ErrorMessage="Please Select an Option"
ControlToValidate="ddlbloodgroup"
ForeColor="Red"></asp:RequiredFieldValidator><br /><br />

        <td><b>Date of Birth :</b></td>
        <td>
        <asp:TextBox ID="txtdateofbirth" runat="server"></asp:TextBox>

```

```
        <asp:RequiredFieldValidator ID="RequiredFieldValidator5"
runat="server" ErrorMessage="Please Enter Donation Amount"
ControlToValidate="txtdateofbirth"></asp:RequiredFieldValidator><br/>
        <asp:RegularExpressionValidator ID="RegularExpressionValidator5"
runat="server" ErrorMessage="Please enter valid date of birth"
ControlToValidate="txtdateofbirth" ValidationExpression="^[A-Za-z0-9-
]+$"></asp:RegularExpressionValidator>      <br/>
    </td>

</tr>
<tr>
    <td><p class="text" style="color:red"><b> Upload Photo
:</b></td><td><asp:FileUpload style="color:red" ID="donorimage" runat="server" />
</td><br/><br/>
    </tr>
    <tr>
        <td></td><td>
            <div class="bttn">
                <div class="agile-submit"> <br><br> <asp:Button ID="btnAddDonor"
runat="server" style="background-color:green;color:white" Text="Register As Donor"
onclick="btnAddDonor_Click" />      </p>
            </div>
            </div>
        </td>
    </tr>
    </table>
</center>
</form>
</center>

</asp:Content>
```

Admin home: This is Admin Home page

The screenshot shows a web browser window with four tabs open, all titled "Blood Bank". The active tab displays the "Admin Home" page for "Life Line". The menu includes links for Admin Home, Schedule Camp, Add Team Member, View / Update / Delete Volunteers, View / Update / Delete Donors, View / Update / Delete Team Member, View / Update / Delete Camp Details, View / Update / Delete Requests for Blood, and Logout. Below the menu, a green banner says "Welcome to Admin Home". At the bottom, there is a black footer bar with social media icons for Facebook, Twitter, Google+, and LinkedIn, followed by the text "©2018 Blood Bank. All rights reserved | Designed by Shubham".

Add update delete camps: Admin panel

The screenshot shows a web browser window with four tabs open, all titled "Blood Bank". The active tab displays the "View / Update / Delete Camps" page. A table lists two blood donation camps:

idcampname	campstatus	camptype	location	doctorinchargephoneno	email	imageurl	campdate
Edit Delete3 blood donation	Upcoming Camp	Free Blood Donation Camp	mumbai	panchal	7877077446shub@gmail.com		14-03-2019
Edit Delete1 blood donation & clothes	Upcoming Camp	Free Blood Donation Camp	Tagore nagarikhori east Mumbai -83	shubham	7877077446shuo@gmail.com		10-04-2019

Coding of view update delete camps:

```
<center>

<br><br>
<h3 class="title clr" style="background-color:green;color:white"> View / Update
/Delete Camps</h3>
</center>

<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
AllowSorting="True"

onselectedindexchanged="GridView1_SelectedIndexChanged"
AutoGenerateColumns="False"
OnRowCancelingEdit="gvImage_RowCancelingEdit" DataKeyNames="id"
CellPadding="3" OnPageIndexChanging="GridView1_PageIndexChanging"
OnRowEditing="gvImage_RowEditing"
OnRowUpdating="gvImage_RowUpdating" OnRowDeleting="gvImage_RowDeleting"
BackColor="White" style="color:White"
BorderColor="White" BorderStyle="Ridge" BorderWidth="2px" CellSpacing="1"
GridLines="None">

<Columns>
    <asp:CommandField ShowDeleteButton="True" ShowEditButton="True" />
    <asp:BoundField DataField="id" HeaderText="id" InsertVisible="False"
ReadOnly="True" SortExpression="id" />
    <asp:TemplateField HeaderText="campname" SortExpression="campname">
        <EditItemTemplate>
            <asp:TextBox ID="txtCampname" runat="server" Text='<%#
Bind("campname") %>'></asp:TextBox>
        </EditItemTemplate>
        <ItemTemplate>
            <asp:Label ID="Label1" runat="server" Text='<%# Bind("campname") %>'></asp:Label>
        </ItemTemplate>
    </asp:TemplateField>
    <asp:TemplateField HeaderText="campstatus" SortExpression="campstatus">
        <EditItemTemplate>
            <asp:TextBox ID="txtCampstatus" runat="server" Text='<%#
Bind("campstatus") %>'></asp:TextBox>
        </EditItemTemplate>
    </asp:TemplateField>

```

```
<ItemTemplate>
    <asp:Label ID="Label2" runat="server" Text='<%# Bind("campstatus") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="camptype" SortExpression="camptype">
    <EditItemTemplate>
        <asp:TextBox ID="txtCamptype" runat="server" Text='<%# Bind("camptype") %>'></asp:TextBox>
    </EditItemTemplate>
    <ItemTemplate> <asp:Label ID="Label3" runat="server" Text='<%# Bind("camptype") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="location" SortExpression="location">
    <EditItemTemplate>
        <asp:TextBox ID="txtLocation" runat="server" Text='<%# Bind("location") %>'></asp:TextBox>
    </EditItemTemplate>
    <ItemTemplate>
<asp:Label ID="Label4" runat="server" Text='<%# Bind("location") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="doctorincharge"
SortExpression="doctorincharge">
    <EditItemTemplate>
        <asp:TextBox ID="txtDoctorincharge" runat="server" Text='<%# Bind("doctorincharge") %>'></asp:TextBox>
    </EditItemTemplate>
    <ItemTemplate>
<asp:Label ID="Label5" runat="server" Text='<%# Bind("doctorincharge") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
<asp:TemplateField HeaderText="phoneno" SortExpression="phoneno">
    <EditItemTemplate> <asp:TextBox ID="txtPhoneno" runat="server" Text='<%# Bind("phoneno") %>'></asp:TextBox>
    </EditItemTemplate>
    <ItemTemplate>
<asp:Label ID="Label6" runat="server" Text='<%# Bind("phoneno") %>'></asp:Label>
    </ItemTemplate>
</asp:TemplateField>
```

```

        <asp:TemplateField HeaderText="email" SortExpression="email">
            <EditItemTemplate>
                <asp:TextBox ID="txtEmail" runat="server" Text='<%# Bind("email") %>'></asp:TextBox>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="Label7" runat="server" Text='<%# Bind("email") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="imageurl" SortExpression="imageurl">
            <EditItemTemplate>
                <asp:image ID="Image1" runat="server" imageurl='<%# Bind("imageurl") %>' border-radius="30%" Width="150" Height="150"></asp:image>
                <asp:FileUpload ID="FileUpload1" runat="server" />
            </EditItemTemplate>
            <ItemTemplate>
                <asp:image ID="Image2" runat="server" imageurl='<%# Bind("imageurl") %>' border-radius="30%" Width="150" Height="150"></asp:image>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="campdate" SortExpression="campdate">
            <EditItemTemplate>
                <asp:TextBox ID="txtcampdate" runat="server" Text='<%# Bind("campdate") %>'></asp:TextBox>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="Label8" runat="server" Text='<%# Bind("campdate") %>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
    <FooterStyle BackColor="#F7DFB5" ForeColor="#8C4510" />
    <HeaderStyle BackColor="green" Font-Bold="True" ForeColor="White" />
    <PagerStyle ForeColor="green" HorizontalAlign="Center" />
    <RowStyle BackColor="white" ForeColor="green" />
    <SelectedRowStyle BackColor="white" Font-Bold="True" ForeColor="green" />
    <SortedAscendingCellStyle BackColor="#FFF1D4" />
    <SortedAscendingHeaderStyle BackColor="#B95C30" />
    <SortedDescendingCellStyle BackColor="#F1E5CE" />
    <SortedDescendingHeaderStyle BackColor="#93451F" />
</asp:GridView>
</center>

```

View Update Delete Donor: Admin panel

ID	Name	Address	Phone No	Emailid	Bloodgroup	Date of Birth	Image		
2	shub	vikhroli	7944044776	shub@gmail.com	A+	19		Edit	Delete
3	shuba	vik	7944044778	shubh@gmail.com	AB+	20		Edit	Delete
5	dakshata	dader	7944044776	dishu@gmail.com	B+	20		Edit	Delete

Coding for view update delete donor:

```
<center>

<br><br>
<h3 class="title clr" style="background-color:green;color:white">
    View / Update /Delete Donors </h3>

</center>
<br/><br />

<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
    AllowSorting="True"

    onselectedindexchanged="GridView1_SelectedIndexChanged"
    AutoGenerateColumns="False"
    OnRowCancelingEdit="gvImage_RowCancelingEdit" DataKeyNames="id"
    CellPadding="3" OnPageIndexChanging="GridView1_PageIndexChanging"
    OnRowEditing="gvImage_RowEditing"
    OnRowUpdating="gvImage_RowUpdating" OnRowDeleting="gvImage_RowDeleting"
    BackColor="white" style="color:green">
```

```
        BorderColor="green" BorderStyle="Ridge" BorderWidth="2px" CellSpacing="1"
        GridLines="Both" font-size="medium">
    <Columns>

        <asp:TemplateField HeaderText="Id" HeaderStyle-Width="150px">

            <EditItemTemplate>
                <asp:TextBox ID="txtid" runat="server" Text='<%# Eval("id") %>' />
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="lblid" runat="server" Text='<%# Eval("id") %>'>
            </asp:Label>
            </ItemTemplate>
            <HeaderStyle Width="150px"></HeaderStyle>
        </asp:TemplateField>

        <asp:TemplateField HeaderText="Name" HeaderStyle-Width="150px">
            <EditItemTemplate>
                <asp:TextBox ID="txtname" runat="server" Text='<%# Eval("name") %>' />
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="lblname" runat="server" Text='<%# Eval("name") %>'>
            </asp:Label>
            </ItemTemplate>
            <HeaderStyle Width="150px"></HeaderStyle>
        </asp:TemplateField>

        <asp:TemplateField HeaderText="Address" HeaderStyle-Width="150px">
            <EditItemTemplate>
                <asp:TextBox ID="txtaddress" runat="server" Text='<%# Eval("address") %>' />
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="lbladdress" runat="server" Text='<%# Eval("address") %>'>
            </asp:Label>
            </ItemTemplate>
            <HeaderStyle Width="150px"></HeaderStyle>
        </asp:TemplateField>

        <asp:TemplateField HeaderText="Phone No" HeaderStyle-Width="150px">
```

```
<EditItemTemplate>
    <asp:TextBox ID="txtphoneno" runat="server" Text='<%# Eval("phoneno") %>' />
</EditItemTemplate>
<ItemTemplate>
    <asp:Label ID="lblmobno" runat="server" Text='<%# Eval("phoneno") %>'></asp:Label>
</ItemTemplate>
<HeaderStyle Width="150px"></HeaderStyle>
</asp:TemplateField>

<asp:TemplateField HeaderText="Emailid" HeaderStyle-Width="150px">
    <EditItemTemplate>
        <asp:TextBox ID="txtemailid" runat="server" Text='<%# Eval("emailid") %>' />
    </EditItemTemplate>
    <ItemTemplate>
        <asp:Label ID="lblemail" runat="server" Text='<%# Eval("emailid") %>'></asp:Label>
    </ItemTemplate>
<HeaderStyle Width="150px"></HeaderStyle>
</asp:TemplateField>

<asp:TemplateField HeaderText="Bloodgroup" HeaderStyle-Width="150px">
    <EditItemTemplate>
        <asp:TextBox ID="txtbloodgroup" runat="server" Text='<%# Eval("bloodgroup") %>' />
    </EditItemTemplate>
    <ItemTemplate>
        <asp:Label ID="lblpassword" runat="server" Text='<%# Eval("bloodgroup") %>'></asp:Label>
    </ItemTemplate>
<HeaderStyle Width="150px"></HeaderStyle>
</asp:TemplateField>

<asp:TemplateField HeaderText="Date of Birth" HeaderStyle-Width="150px">
    <EditItemTemplate>
        <asp:TextBox ID="txtdateofbirth" runat="server" Text='<%# Eval("dateofbirth") %>' />
    </EditItemTemplate>
    <ItemTemplate>
```

```
<asp:Label ID="lbldateofbirth" runat="server" Text='<%# Eval("dateofbirth") %>'>
</asp:Label>
</ItemTemplate>
<HeaderStyle Width="150px"></HeaderStyle>
</asp:TemplateField>

<asp:TemplateField HeaderText="Image" HeaderStyle-Width="150px">
<EditItemTemplate>
<asp:TextBox ID="txtimgurl" runat="server" Text='<%# Eval("imgurl") %>' Width="100" Height="100"/>
</EditItemTemplate>
<ItemTemplate>
<asp:image ID="Image2" runat="server" imageurl='<%# Eval("imgurl") %>' Width="100" Height="100"/>
</ItemTemplate>

<HeaderStyle Width="150px"></HeaderStyle>
</asp:TemplateField>
<asp:TemplateField HeaderStyle-Width="150px">
<ItemTemplate>
<asp:LinkButton ID="LkB1" runat="server" CommandName="Edit">Edit</asp:LinkButton>
<asp:LinkButton ID="LkB11" runat="server" CommandName="Delete">Delete</asp:LinkButton>
</ItemTemplate>
<EditItemTemplate>
<asp:LinkButton ID="LkB2" runat="server" CommandName="Update">Update</asp:LinkButton>
<asp:LinkButton ID="LkB3" runat="server" CommandName="Cancel">Cancel</asp:LinkButton>
</EditItemTemplate>
<HeaderStyle Width="150px"></HeaderStyle>
</asp:TemplateField>

<asp:CommandField ShowSelectButton="True" />
</Columns>

<FooterStyle BackColor="green" ForeColor="white" />
<HeaderStyle BackColor="white" Font-Bold="True" ForeColor="green" />
```

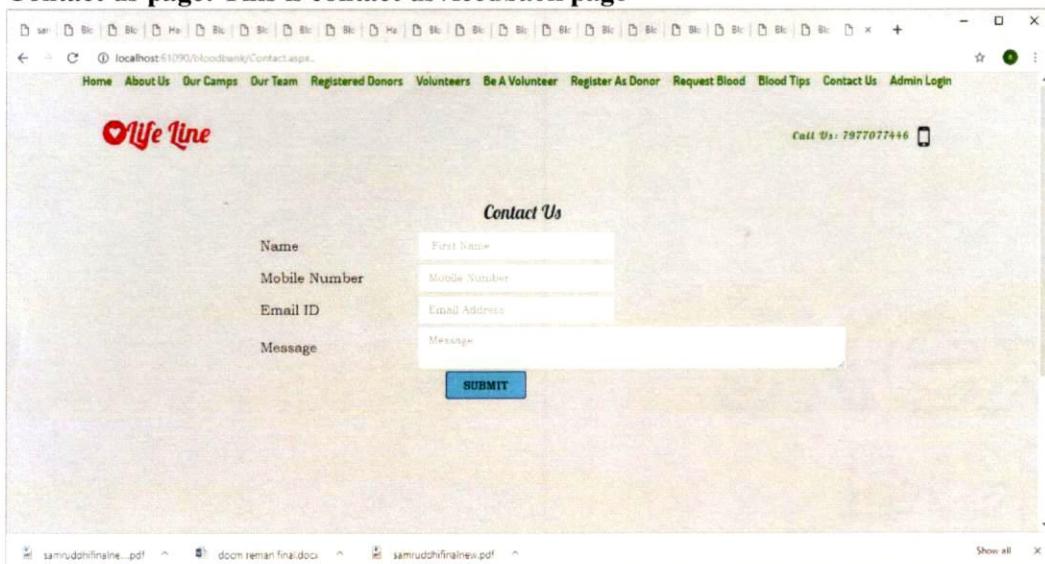
```

<PagerStyle ForeColor="green" HorizontalAlign="Right" BackColor="white" />
<RowStyle BackColor="white" ForeColor="green" />
<SelectedRowStyle BackColor="white" Font-Bold="True" ForeColor="green" />
<SortedAscendingCellStyle BackColor="#f3821b" />
<SortedAscendingHeaderStyle BackColor="#f3821b" />
<SortedDescendingCellStyle BackColor="#f3821b" />
<SortedDescendingHeaderStyle BackColor="#f3821b" />

</asp:GridView>

```

Contact us page: This is contact us /feedback page



Coding of contact us page:

```

<h3>Contact Us</h3>
<table style="font-family: 'Bookman Old Style'; font-size:medium; width:1000px; margin-left:200px">

    <tr style="height:10px">
        <td></td>
        <td></td>
    </tr>

```

```
<tr style="height:40px">

    <td style="padding-left:60px;font-size: 18px; ">
        Name
    </td>
    <td>
        <div class="col-lg-6">
            <div class="input-group">
                <asp:TextBox ID="txtfname" Width="255px"
Height="35px" runat="server" class="form-control" placeholder="First
Name"></asp:TextBox>
                <asp:RequiredFieldValidator
ID="RequiredFieldValidator1" runat="server" Style="top: 10px;
left: 200px; position: absolute; height: 22px; width: 128px"
ErrorMessage="First Name: Field Required" ValidationGroup="g1"
ControlToValidate="txtfname"
ForeColor="Red">*</asp:RequiredFieldValidator>

                <asp:RegularExpressionValidator ID="regName"
runat="server" Style="top: 14px;
left: 216px; position: absolute; height: 22px; width: 177px"
ControlToValidate="txtfname" ValidationGroup="g1" ValidationExpression="^[_a-zA-
Z'.\s]{1,30}" Text="" ErrorMessage="First Name: Text Only" ForeColor="Red" />

            </div>
        </div>
    </td>
</tr>
<tr style="height:40px">
    <td style="padding-left:60px;font-size: 18px; ">
        Mobile Number
    </td>
    <td>
        <div class="col-lg-6">
            <div class="input-group">
                <asp:TextBox ID="txtmob"
onkeypress="javascript:return isNumber(event)" Width="255px" Height="35px"

```

```

runat="server" class="form-control" placeholder="Mobile Number"
MaxLength="10"></asp:TextBox>
            <asp:RequiredFieldValidator
ID="RequiredFieldValidator5" runat="server" Style="top: 10px;
left: 200px; position: absolute; height: 22px; width: 128px"
ErrorMessage="Mobile Number: Field Required" ValidationGroup="g1"
ControlToValidate="txtmob" ForeColor="Red">*</asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator
ID="RegularExpressionValidator1" runat="server"
ControlToValidate="txtmob" ErrorMessage="Mobile No. 10digit Required"
ValidationGroup="g1" style="top: 14px;
left: 216px; position: absolute; height: 22px; width: 177px" ForeColor="Red"
ValidationExpression="[0-9]{10}">*</asp:RegularExpressionValidator>

        </div>
        </div>
    </td>
</tr>

        <tr style="height:40px">
    <td style="padding-left:60px;font-size: 18px; ">
        Email ID
    </td>
    <td>
        <div class="col-lg-6">
            <div class="input-group">
                <asp:TextBox ID="txtemail" Width="255px" Height="35px" runat="server" class="form-control" placeholder="Email Address"></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator6" runat="server" Style="top:
10px;
left: 200px; position: absolute; height: 22px; width: 128px" ErrorMessage="Email:
Field Required" ValidationGroup="g1"
ControlToValidate="txtemail" ForeColor="Red">*</asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator ID="RegularExpressionValidator3" runat="server" Style="top: 14px;
left: 216px; position: absolute; height: 22px; width: 177px"
ErrorMessage="Email: Format Invalid" ControlToValidate="txtEmail"
ValidationGroup="g1" ForeColor="Red">
            </div>
        </div>
    </td>
</tr>

```

```

        ValidationExpression="\w+([-.\.]\w+)*@\w+([.-]\w+)*\.\w+([.-]\w+)*">*</asp:RegularExpressionValidator>
    </div>
    </div>
</td>
</tr>

<tr style="height:40px">
<td style="padding-left:60px;font-size: 18px; ">
Message
</td>
<td>
<div class="col-lg-6">
<div class="input-group">
<asp:TextBox ID="txtSub" TextMode="MultiLine"
Width="555px" runat="server" class="form-control" placeholder="Message
"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" Style="top: 10px;
left: 200px; position: absolute; height: 22px; width: 255px"
ErrorMessage="Message: Field Required" ValidationGroup="g1"
ControlToValidate="txtSub" ForeColor="Red"></asp:RequiredFieldValidator>

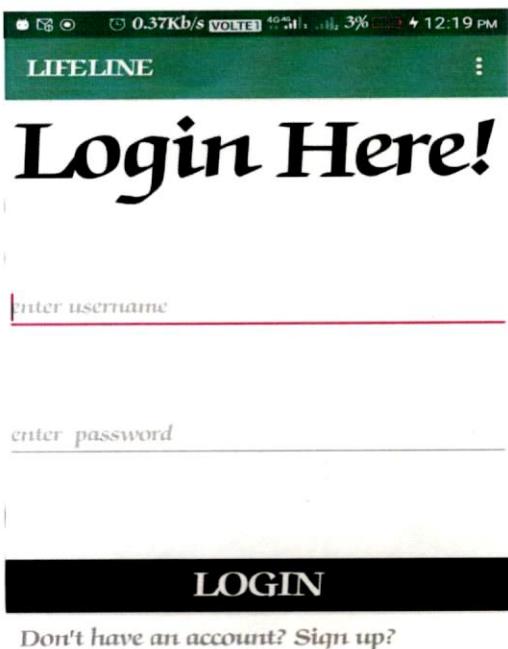
<tr style="height:40px">
<td colspan="2" style="vertical-align:central;padding-left:300px">
<asp:Button ID="Button1" class="btn btn-default dropdown-toggle"
ValidationGroup="g1" Height="35px" runat="server" Text="Submit" BackColor="#21a9e1"
Font-Bold="True" BorderColor="Blue" OnClick="Button1_Click" ></asp:Button>
</td>

<tr style="height:30px">
<td></td>
<td>
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
ShowMessageBox="True" ShowSummary="False" ValidationGroup="g1" Height="38px" />
</td>
</tr>

</table>

```

Android :login page



Android login page coding:

```
package com.example.lifelineproject;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class mainpage extends AppCompatActivity {

    Button b1,b2,b3,b4,b5;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_mainpage);

b1=(Button)findViewById(R.id.button1);

b2=(Button)findViewById(R.id.button2);

b3=(Button)findViewById(R.id.button3);

b4=(Button)findViewById(R.id.button4);

b5=(Button)findViewById(R.id.button5);

b1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i=new Intent(getApplicationContext(),info.class);

startActivity(i);

b2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i=new Intent(getApplicationContext(),pre.class);

startActivity(i);

b3.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i=new Intent(getApplicationContext(),donor.class);

startActivity(i);

b4.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent i=new Intent(getApplicationContext(),acceptor.class);
```

```
        startActivity(i);

    }

});

b5.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        Intent i=new Intent(getApplicationContext(),hospital.class);

        startActivity(i);
    }
});
```

mainpage.xml

```
<?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:orientation="vertical"

    tools:context=".mainpage">

    <Button

        android:id="@+id/button1"

        android:textSize="30sp"

        android:background="#000"

        android:textColor="#F8F4F4"

        android:layout_marginTop="20dp"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_weight="1"
    >
```

```
        android:text="BLOOD INFORMATION" />
```

```
<Button  
    android:id="@+id/button2"  
    android:textSize="30sp"  
    android:background="#000"  
    android:textColor="#F8F4F4"  
  
    android:layout_weight="1"  
    android:layout_marginTop="20dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Precautions" />
```

```
<Button  
    android:id="@+id/button3"  
    android:textSize="30sp"  
    android:background="#000"  
    android:textColor="#F8F4F4"  
  
    android:layout_marginTop="20dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="Donor" />
```

```
<Button  
    android:id="@+id/button4"  
    android:textSize="30sp"  
    android:background="#000"  
    android:textColor="#F8F4F4"  
  
    android:layout_marginTop="20dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:text="Acceptor" />  
  
<Button  
    android:id="@+id/button5"  
    android:textSize="30sp"  
    android:background="#000"  
    android:textColor="#F8F4F4"  
    android:layout_marginTop="20dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:layout_marginBottom="20dp"  
    android:text="Hospitals" /> </LinearLayout>
```

Android : Home Page



Admin home page coding:

```
package com.example.lifelineproject;  
  
import android.content.Intent;  
  
import android.support.annotation.NonNull;  
  
import android.support.v7.app.AppCompatActivity;  
  
import android.os.Bundle;  
  
import android.view.Menu;  
  
import android.view.MenuItem;  
  
import android.view.View;  
  
import android.widget.Button;  
  
import android.widget.EditText;  
  
import android.widget.TextView;  
  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;
```

```
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

public class home extends AppCompatActivity {

    EditText ed1,ed2;
    Button b1,b2;
    TextView txt;
    FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);

        ed1=(EditText)findViewById(R.id.edt1);
        ed2=(EditText)findViewById(R.id.edt2);
        b1=findViewById(R.id.btlogin);
        b2=findViewById(R.id.bnln);

        txt=findViewById(R.id.txtlogin);
        mAuth=FirebaseAuth.getInstance();
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {logInActivity();}
        });
    }
}
```

```
b2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent i=new Intent(getApplicationContext(),mainpage.class);  
        startActivity(i);  
  
        txt.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                ToSignUpActivity();  
            }  
            private void ToSignUpActivity() {  
                Intent i=new Intent(getApplicationContext(),SignUp.class);  
                startActivity(i);  
            }  
            private void logInActivity() {  
                String name=ed1.getText().toString();  
                String pass=ed2.getText().toString();  
  
                mAuth.signInWithEmailAndPassword(name,pass).addOnCompleteListener(new  
                OnCompleteListener<AuthResult>() {  
                    @Override  
                    public void onComplete(@NonNull Task<AuthResult> task) {  
                        if(task.isSuccessful())  
                        {  
                            Toast.makeText(getApplicationContext(),"Log In  
                            SuccessFully..!!",Toast.LENGTH_SHORT).show();  
                            Intent i=new Intent(getApplicationContext(),mainpage.class);  
                            startActivity(i);  
                        }  
                    }  
                });  
            }  
        });  
    }  
});
```

```
        else
            Toast.makeText(getApplicationContext(),"Login
Unsuccessfull..!!",Toast.LENGTH_SHORT).show();

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.opt,menu);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()){
            case R.id.item1:
                Intent i=new Intent(getApplicationContext(),aboutus.class);
                startActivity(i);
                break;
            case R.id.item2:
                Intent i1=new Intent(getApplicationContext(),gallery.class);
                startActivity(i1);
                break;
            case R.id.item3:
                Intent ib=new Intent(getApplicationContext(),bmi.class);
                startActivity(ib);
                break;
            case R.id.item4:
                Intent iu=new Intent(getApplicationContext(),updateprofile.class);
                startActivity(iu);
                break;
        }
    }
}
```

```
case R.id.item5:  
    Intent u=new Intent(getApplicationContext(),adminmain.class);  
    startActivity(u);  
    break;  
  
default:  
}
```

Home.xml

```
<?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:orientation="vertical"  
    android:layout_height="match_parent"  
    tools:context=".home">  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Login Here!"  
    android:layout_gravity="center"  
    android:textSize="70sp"  
    android:textColor="#000"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText  
    android:id="@+id/edt1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter username"  
    android:layout_marginTop="70dp"  
    android:inputType="textPersonName"  
    android:text="" />
```

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/btnn"/>
```

```
<EditText  
    android:id="@+id/edt2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter password"  
    android:layout_marginTop="70dp"  
    android:inputType="textPersonName"  
    android:text="" />
```

```
<Button
```

```
        android:id="@+id/btlogin"
        android:textSize="30sp"
        android:textColor="#F0EAEA"
        android:background="#000"
        android:layout_marginTop="90dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="Don't have an account? Sign up?"
    android:layout_margin="10sp"
    android:textSize="20sp"
    android:id="@+id/txtlogin"/> </LinearLayout>
```

Android registration page:

The screenshot shows a mobile application interface for user registration. At the top, there is a status bar with signal strength, battery level (58%), and time (5:28 PM). Below the status bar is a dark green header bar with the word "LIFELINE" in white capital letters. The main content area has a white background. It contains several input fields and controls. At the top, there is a red underline under the placeholder "enter first name". Below it is a red underline under the placeholder "enter last name". Further down are red underlines under the placeholders "enter contact number", "enter email id", "enter date of birth", "enter weight", and "enter your age". At the bottom of the form, there are two radio buttons: one labeled "Male" and another labeled "Female". A large black button at the very bottom is labeled "SUBMIT" in white capital letters.

enter first name

enter last name

enter contact number

enter email id

enter date of birth

enter weight

enter your age

enter your city

Male Female

SUBMIT

Android registration page coding:

```
package com.example.lifelineproject;  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import com.google.firebaseio.database.DatabaseReference;  
import com.google.firebaseio.database.FirebaseDatabase;  
public class donorreg extends AppCompatActivity {  
    EditText e1,e2,e3,e4,e5,e6,e7,e8;  
    Button b1;  
    DatabaseReference reference;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_donorreg);  
  
        b1 = (Button) findViewById(R.id.btnsubmit);  
        e1 = (EditText) findViewById(R.id.edt1);  
        e2 = (EditText) findViewById(R.id.edt2);  
        e3 = (EditText) findViewById(R.id.edt3);  
        e4 = (EditText) findViewById(R.id.edt4);  
        e5 = (EditText) findViewById(R.id.edt5);
```

```
e6 = (EditText) findViewById(R.id.edt6);
e7 = (EditText) findViewById(R.id.edt7);
e8 = (EditText) findViewById(R.id.edt8);

reference = FirebaseDatabase.getInstance().getReference("Apos");
b1.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        AddData();
    }
private void AddData() {
    String fname=e1.getText().toString().trim();
    String lname=e2.getText().toString().trim();
    String no=e3.getText().toString().trim();
    String email=e4.getText().toString().trim();
    String dob=e5.getText().toString().trim();
    String weight=e6.getText().toString().trim();
    String age=e7.getText().toString().trim();
    String city=e8.getText().toString().trim();
    String id=reference.push().getKey();

    SaveData saveData=new SaveData(fname,lname,no,email,dob,weight,age,city);
    reference.child(id).setValue(saveData);

    Toast.makeText(this, "successful", Toast.LENGTH_SHORT).show();
}
```

donorreg.xml

```
<?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"
    tools:context=".donorreg">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Register Here!"
            android:layout_gravity="center"
            android:textSize="50sp"
            android:textColor="#000"
            android:layout_marginTop="20dp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
    
```

```
<EditText  
    android:id="@+id/edt1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter first name"  
    android:layout_marginTop="10dp"  
    android:inputType="textPersonName"  
    android:text="" />
```

```
<EditText  
    android:id="@+id/edt2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="enter last name"  
    android:ems="10"  
    android:layout_marginTop="10dp"  
    android:inputType="textPersonName"  
    android:text="" />
```

```
<EditText  
    android:id="@+id/edt3"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter contact number"  
    android:layout_marginTop="10dp"/>
```

```
<EditText  
  
    android:id="@+id/edt4"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter email id"  
    android:layout_marginTop="10dp"  
    android:inputType="textPersonName"  
    android:text="" />
```

```
<EditText  
  
    android:id="@+id/edt5"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter date of birth"  
    android:layout_marginTop="10dp"  
    android:inputType="textPersonName"  
    android:text="" />
```

```
<EditText  
  
    android:id="@+id/edt6"  
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter weight"  
    android:layout_marginTop="10dp"  
    android:inputType="textPersonName"  
    android:text="" />
```

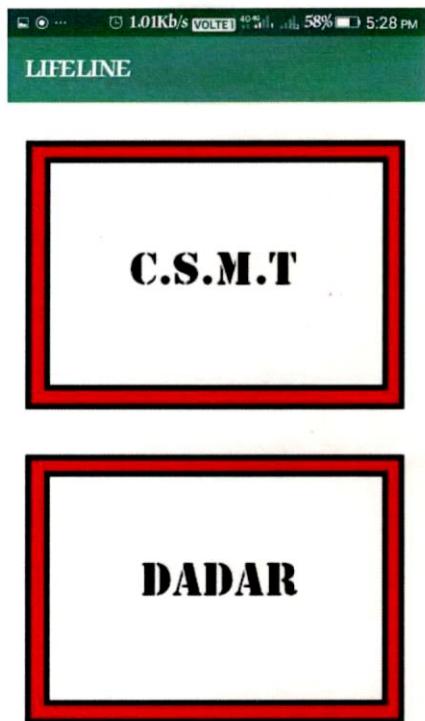
```
<EditText  
    android:id="@+id/edt7"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:inputType="textPersonName"  
    android:hint="enter your age"  
    android:layout_marginTop="10dp"/>
```

```
<EditText  
    android:id="@+id/edt8"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:ems="10"  
    android:hint="enter your city"  
    android:layout_marginTop="10dp"/>
```

```
<RadioGroup  
    android:layout_marginTop="12dp"  
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Male"
        android:textSize="15sp"
        android:layout_marginRight="50dp"/>
    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="50dp"
        android:text="Female"
        android:textSize="15sp"/>
</RadioGroup>
<Button
    android:id="@+id/btnsubmit"
    android:textSize="35sp"
    android:background="#000"
    android:textColor="#F3EFEF"
    android:layout_marginTop="30dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="submit" />
</LinearLayout>
</ScrollView>
```

Android event page :



Android event page coding;

```
package com.example.lifelineproject;  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ImageButton;  
import android.widget.ImageView;  
  
public class events extends AppCompatActivity {  
  
    ImageView i1,i2,i3,i4,i5,i6,i7,i8;
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_events);  
    i1=(ImageButton)findViewById(R.id.imageView1);  
    i2=(ImageButton)findViewById(R.id.imageView2);  
    i3=(ImageButton)findViewById(R.id.imageView3);  
    i4=(ImageButton)findViewById(R.id.imageView4);  
    i5=(ImageButton)findViewById(R.id.imageView5);  
  
    i1.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            Intent i=new Intent(getApplicationContext(),eventandheri.class);  
            startActivity(i);  
  
        i2.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Intent i=new Intent(getApplicationContext(),events cst.class);  
                startActivity(i);  
  
            i3.setOnClickListener(new View.OnClickListener() {  
                @Override  
                public void onClick(View v) {  
                    Intent i=new Intent(getApplicationContext(),eventdadar.class);  
                    startActivity(i);  
                }  
            }  
        }  
    }  
}
```

```
i4.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent i=new Intent(getApplicationContext(),eventpanvel.class);  
        startActivity(i);  
    }  
}  
i5.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent i=new Intent(getApplicationContext(),eventthane.class);  
        startActivity(i);  
    }  
}  
  
events.xml  
  
<?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"  
    tools:context=".events">  
  
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:orientation="vertical"  
    >  
  
<ImageButton  
    android:layout_marginTop="20dp"  
    android:id="@+id/imageView1"  
    android:layout_width="wrap_content"
```

```
        android:layout_height="250dp"
        android:layout_weight="1"
        app:srcCompat="@drawable/an" />

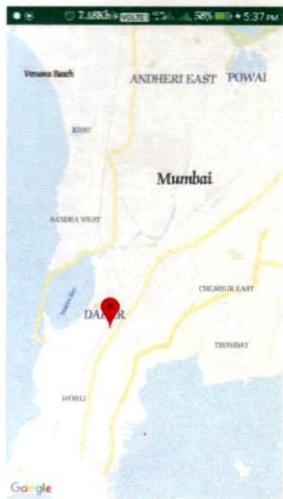
<ImageButton
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="250dp"
    android:layout_marginTop="20dp"
    android:layout_weight="1"
    app:srcCompat="@drawable/cs" />

<ImageButton
    android:id="@+id/imageView3"
    android:layout_width="wrap_content"
    android:layout_height="250dp"
    android:layout_weight="1"
    app:srcCompat="@drawable/da"
    android:layout_marginTop="20dp"
    />

<ImageButton
    android:id="@+id/imageView4"
    android:layout_width="wrap_content"
    android:layout_height="250dp"
    android:layout_weight="1"
    app:srcCompat="@drawable/pa"
    android:layout_marginTop="20dp" />
```

```
<ImageButton  
    android:id="@+id/imageView5"  
    android:layout_width="wrap_content"  
    android:layout_height="250dp"  
    android:layout_weight="1"  
    app:srcCompat="@drawable/th"  
    android:layout_marginTop="20dp"/>  
</LinearLayout></ScrollView>
```

Android Map :



Android map coding :

```
map.java  
  
package com.example.lifelineproject;  
  
import android.support.v4.app.FragmentActivity;  
  
import android.os.Bundle;  
  
import android.widget.Toast;  
  
import com.google.android.gms.maps.CameraUpdateFactory;  
  
import com.google.android.gms.maps.GoogleMap;  
  
import com.google.android.gms.maps.OnMapReadyCallback;
```

```
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

public class map extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    static final LatLng DADAR=new LatLng(19.0180,72.8448);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager()
            .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));*/
        Marker DADAR=googleMap.addMarker(new
        MarkerOptions().position(map.DADAR).title("hospitals"));
        googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(map.DADAR,15));
    }
}
```

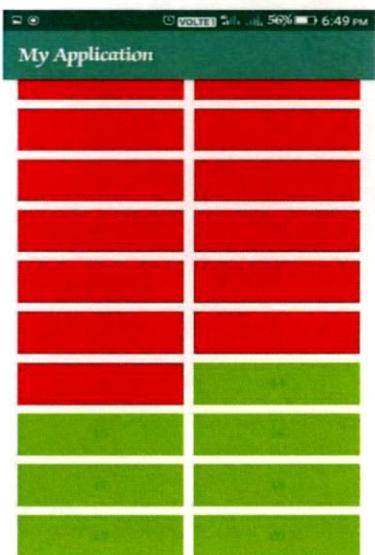
```
googleMap.animateCamera(CameraUpdateFactory.zoomTo(10),200,null);

if(googleMap==null){

    Toast.makeText(getApplicationContext(),"sorry unable to create maps",
    Toast.LENGTH_SHORT).show();

}}
```

Android bed availability:



Android bed availability coding:

```
package com.example.myapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.ToggleButton;

public class mybed extends AppCompatActivity implements View.OnClickListener {

    LinearLayout linearLayoutBed1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);

setContentView(R.layout.activity_mybed_new);

linearLayoutBed1 = (LinearLayout) findViewById(R.id.linearLayoutBed1);

populateBedUI(20, 13, false);

private void populateBedUI(int totalCount1, int occupiedCount1, boolean
enableToggleButton) {

    linearLayoutBed1.removeAllViews();

    LinearLayout.LayoutParams layoutParamsHorizontal = new
    LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT
        , LinearLayout.LayoutParams.WRAP_CONTENT);

    LinearLayout linearLayoutHorizontal = new LinearLayout(this);

    linearLayoutHorizontal.setOrientation(LinearLayout.HORIZONTAL);

    linearLayoutHorizontal.setLayoutParams(layoutParamsHorizontal);

    for (int i = 0; i < totalCount1; i++) {

        if (i % 2 == 0) {

            linearLayoutHorizontal = new LinearLayout(this);

            linearLayoutHorizontal.setOrientation(LinearLayout.HORIZONTAL);

            linearLayoutHorizontal.setLayoutParams(layoutParamsHorizontal);

        }

        ToggleButton toggleButton = new ToggleButton(this);

        toggleButton.setId((i + 1));

        toggleButton.setTextOff("'" + (i + 1));

        toggleButton.setTextOn("'" + (i + 1));

        toggleButton.setBackgroundResource(R.drawable.toggle_button_selector);

        if (i < occupiedCount1) {

            toggleButton.setChecked(true);

        } else {

            toggleButton.setChecked(false);

        }

    }

}
```

```
    LinearLayout.LayoutParams layoutParams = new
LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT
    , LinearLayout.LayoutParams.WRAP_CONTENT);

    layoutParams.leftMargin = 10;
    layoutParams.rightMargin = 10;
    layoutParams.topMargin = 10;
    layoutParams.bottomMargin = 10;
    layoutParams.weight = 1;

    toggleButton.setLayoutParams(layoutParams);

    if (enableToggleButton) {
        //TODO to enable button click
        toggleButton.setOnClickListener(this);
        toggleButton.setEnabled(true);
    } else {
        //TODO to disable button click
        toggleButton.setEnabled(false);
    }

    linearLayoutHorizontal.addView(toggleButton);
    linearLayoutBed1.removeView(linearLayoutHorizontal);
    linearLayoutBed1.addView(linearLayoutHorizontal);

@Override
public void onClick(View v) {}
```

Chapter 6

Testing and Results

5.1 Testing Approach

Testing is a life-cycle activity which starts at the time of proposal and end when acceptance testing is completed and product is finally delivered to customer. Testing is done at every stage of life cycle from proposal testing till acceptance testing. After proposal there is Requirement testing ,Design testing,Unit testing , Module testing, Integration testing, System testing and last is acceptance testing. In our project we used Unit testing ,Integration testing ,black box testing and white box testing.

- Unit testing:**

Unit is the smallest part of software which includes codes,classes, and methods and can be tested individually for correctness. Individual components and units are tested to ensure that they work as individual as defined in design.In our project we test all module individually and ensure they work correctly.

- Integration testing:**

Integration testing involves integration of units to make module. It may start at module level,where different units and components come together to form a module and go up to system level. Integration testing also tests the functionality of software. Integration testing mainly focus on output. We integrate all units and then test module .

USABILITY TESTING:**Admin login:**

Sr no.	Variable name	Valid case	Invalid case	Sample input	Remark (valid or invalid)
1	Admin user id	Special characters 0-9, a-z	Not special characters	admin@gmail.com Admin@.com	Valid Invalid

Donor Signup:

Sr no.	Variable name	Valid case	Invalid case	Sample input	Remark (valid or invalid)
1	Name	a-z, A-Z, 0-9, Special characters	-	member Member1 member@28	Valid Valid Valid
2	Age	0-9	Special characters, A-Z, a-z	20 2B	Valid Invalid
3	Address	a-z, A-Z, 0-9, Special characters	-	Abc Ab1 ab@1	Valid Valid valid
4	Mobile	0-9	a-z, No characters	9867432194 abcd@1	Valid Invalid
5	emailid	a-z, only '@'	No spl char	shubh@gmail.com shub#@gmail.com	Valid Invalid

PERFORMANCE TESTING: ADMIN

Case number	Case description	Expected result	Transaction	User think time (per second)
1	Admin enters in the account with username and password.	Admin page is displayed.	Admin-login	2
2	Admin fill's the camps detail and add the camp	Camp event added successfully	Add camps	5
3	Add Volunteer	volunteer Added successfully by Admin	Add Volunteer	2
4	View feedback	Feedback view successfully.	Admin will view	2
5	View blood request	successfully	view Blood req	2

PERFORMANCE TESTING USER:

Case number	Case Description	Expected result	Transaction	User think time per sec
1	View camps on click	Camps page displayed	user panel	2
2	View registered donors	Registered donors displayed	User panel	2
3	View volunteer	View volunteers	User panel	2
4	Be a volunteer	Signup form volunteer	User panel	2
5	Be a donor	Signup form donor	User panel	2
6	Request blood	Enquiry form	User panel	2
7	Blood tips	View blood tips	User panel	2

TEST REPORTS:

Sr	Test for module	Input	Expected Result	Remark
1	Admin login	Admin id Password	1-check and validate User id and password	Success

			stored in database. 2-if any of them is incorrect, “invalid user id and password” this validation is displayed.	
2	Add member	Member Name Age Address Mobile Email ID Parent's Email ID Select pic	1-validate all fields and store. 2- if any one of them is incorrect/null Proper validation should be displayed	Success
3	Add blood request	Name Age Blood group Address Phone number Doctor name	1-validate all fields store 2- if any one of them is incorrect/null Proper validation should be displayed	Success

CHAPTER 7

7.1 CONCLUSION

- We are developing a blood bank android application to reduce the human efforts and to increase crystal clear transparency between society and blood bank.
- It will allow the members of the society to access information about a blood donation etc, its importance in life.
- It also helps the society to connect to world and people in need .
- Thus this application provides a virtual tour of life saving events occurring around them.

7.1.1 SIGNIFICANCE OF THE SYSTEM

1. The most important things about this tool are that they are user-friendly.
2. camps or events notification can be conveyed to all members .
3. users can enquire and send feedback to admin through contact page
4. Privacy of data.
5. Fast and efficient.
6. Ease to maintain data.
7. Stay connected and build bonds.

7.2 LIMITATION OF THE SYSTEM

1. Only admin can view the all feedback ,only admin can add camps.
2. We think that not a single project is ever considered as complete forever because our mind is always thinking new and our necessities also are growing.
3. User requirements keep changing as the system is being used.

We think that not a single project is ever considered as complete forever because our mind is always thinking new and our necessities also are growing.

4. User requirements keep changing as the system is being used.
5. Based on the future security issues, security improvement can be done using emerging technologies.

7.3 FUTURE SCOPE OF THE SYSTEM

1. We think that not a single project is ever considered as complete forever because our mind is always thinking new and our necessities also are growing.
2. In near future, their might be some kind of changes happens like online payment transaction.
3. The project can be enhanced further by developing a full fledge automated system.

BIBLIOGRAPHY

- www.youtube.com
- www.tutorialspoint.com
- www.abhiandroid.com
- www.w3schols.com
- www.stackoverflow.com
- www.Github.com
- www.javaTpoint.com
- www.Google.com