

A  
Project Report titled  
“BLOOD BANK MANAGEMENT SYSTEM”

Submitted to  
Department of Information Technology  
of



Bunts Sangha's  
S. M. Shetty College of Science, Commerce and Management Studies, Powai

For Partial Fulfillment for Degree of  
Bachelor of Science (Information Technology)

2021 – 2022

In the Subject Head  
Project (VI Semester)

Submitted by

SHARMA MOHIT RAMPRAVESH

Bunts Sangha's  
S. M. Shetty College of Science, Commerce and Management Studies, Powai

**Bunts Sangha's**  
**S. M. Shetty College of Science, Commerce and Management Studies, Powai**

**CERTIFICATE**

This is to certify that **SHARMA MOHIT RAMPRAVESH** (Exam Seat No: 2017616 ), a final year student of **Bachelor of Information Technology** (B.Sc.IT) from University of Mumbai has successfully completed the project entitled "**BLOOD BANK MANAGEMENT SYSTEM**" as a part of academic in the subject headProject (VI Semester) which is approved for degree of Bachelor of Information Technology (B.Sc.IT) a undergraduate course of **Mumbai University** during academic year **2021-22**.

**Project Guide**

**Examiner**

**Coordinator**



**BUNTS SANGHA'S  
S.M. SHETTY COLLEGE OF SCIENCE, COMMERCE  
&  
MANAGEMENT  
STUDIES POWAI, MUMBAI-76**

PERMANENTLY AFFILIATED TO THE UNIVERSITY OF MUMBAI  
ACCREDITED BY N.A.A.C. – ‘A’ GRADE  
IMC RBNQ Certificate of Merit 2019  
ISO 21001:2018 Certified

---

**PLAGIARISM CERTIFICATE**

This is to certify that **Mr. Mohit Sharma (Roll No.67)**, third year (Semester VI) student of **Bachelor of Information Technology** from University of Mumbai has submitted the project book content for plagiarism check. It was found that his content is good, as per format and the duplication is under UGC Recommended rules given in UGC (Promotion of Academic Integrity and Prevention of Plagiarism in Higher Educational Institutions) Regulations, 2018 [D.O.No.F.1-18/2010 (CPP-II)].

**In-Charge  
Plagiarism Committee**

**Project Guide**

**Coordinator**

**Stamp:**



**Date: 13.04.2022**

# **BLOOD BANK MANAGEMENT SYSTEM**

**SUBMITTED BY:**

**SHARMA MOHIT**

**ROLL NO: 67**

**PROJECT GUIDE:**

**PROF.MITHILESH CHAUHAN**

## PREFACE

It gives me enormous pleasure to present this report of project documentation / black book and the application I developed as a project of my last year. The idea came to me when I realized that call log data was saved only for three to four days.

Then, I started looking for ideas and after having skipped several ideas during the Semester 5 and finished partially or almost on other topics, I thought to continue with this.

And after much research, once we got the basic configuration, we discovered other utilities essentially needed to implement my application.

Once, I completed the processing of my modules, I started the research and added additional structures and functionality to make this application healthy and harmonious.

So, in conclusion, I tried to form a dedicated application and this black book is maintained for the same purpose.

Therefore, I hope you are analysis the book and that user appreciate the use of this website and appreciate it because it will be satisfactory for our efforts.

## Acknowledgement

I would really like to explicit my thank you of gratitude to the Management of Bunt's Sangha's S.M. Shetty College as well as Project Guide Mr. Mithilesh Chauhan who me This golden opportunity to do that super undertaking on the subject Blood Bank Management System, which also helped me to know about many new things.

I am very thankful to all those people who have helped me and guided me to complete it. I also thank my parents who have supplied my all the sources required to make this undertaking Project.

-Sharma Mohit

<u>Chapter</u>	<u>Sr.</u>	<u>Topic</u>	<u>Pg.</u>
<u>No.</u>	<u>No.</u>		<u>No.</u>
<b>1.</b>		<b>Introduction</b>	10-11
	1.1	Background	10
	1.2	Objective	10
	1.3.1	Purpose	10
	1.3.2	Scope	11
	1.3.3	Applicability	11
	1.4	Achievement	11
<b>2.</b>		<b>Survey Of Technology</b>	12-14
	2.1	Available technologies	12
	2.2	Technologies used	14
<b>3.</b>		<b>Requirement &amp; Analysis</b>	15-19
	3.1	Problem Definition	15
	3.2	Requirement Specification	15
	3.3	Planning & Scheduling	16
		Gantt Chart	17
	3.4	Software Requirement	18
	3.5	Hardware Requirement	18
	3.6	Functional Requirements in Words	18
<b>4.</b>		<b>System Design</b>	20-37

	4.1	Basic Module	20
	4.2	Data Design	21
	4.3	Class Diagram	22
	4.4	Data Flow Diagram	25
	4.5	ER Diagram	27
	4.6	Use case Diagram	29
	4.7	Activity Diagram	31
	4.8	Sequence Diagram	34
	4.9	System Flowchart	35
<b>5.</b>		<b>Implementation And Testing</b>	<b>38-51</b>
	5.1	Implement Approaches	38
	5.2	Coding Details & Efficiency	38
	5.3	Testing Approaches	41
	5.4	Test cases Design	45
	5.5	Core Segment Code	47
<b>6.</b>		<b>Results And Discussion</b>	<b>52-56</b>
	6.1	Test Reports	52
	6.2	User Documentation	54
<b>7.</b>		<b>Conclusions</b>	<b>57-58</b>
	7.1	Conclusions	57
	7.2	Limitation Of the Project	57

	7.3	Future Scope of the project	57
	7.4	References	58

## Chapter 1

### Introduction

#### **1.1Background:-**

The fundamental point of this venture is to save lives of individuals by giving blood.

Our venture Blood Bank Management System utilizing Visual Studio is grown so client can see the data about the blood bunches that is accessible or not.

This venture is created by four viewpoints for example medical clinic, Blood bank stocks, and volunteer Donor.

This, application lessen the chance to more prominent degree that is looking for the necessary blood in blood donation centre stocks and emergency clinics.

Accordingly, this application gives the expected data quicker than expected and furthermore helps in speedier navigation.

#### **1.2Objectives**

To deal with the subtleties of Blood, Donor, Blood bunch, Blood Bank stock.

It deals with all the data about Blood, Blood Groups stocks.

The undertaking is completely worked at authoritative end and hence just director is ensured the entrance.

The Purpose of the undertaking to lessen the manual work for dealing with the Blood, Donor and Blood Groups.

It tracks every one of the insights regarding the blood gatherings, blood donation centre stock as well as about blood benefactor and so on.

#### **1.3 Purpose, Scope, Applicability and Achievement:-**

##### **1.3.1 Purpose:-**

The purpose of making this application that to reduce the manual work for managing the Blood, Donor, Blood Groups and stocks.

It will help to reduce paper work.

It will maintain the donor information in arrange manner.

Records are being maintained manually in a registered including donor's name and their blood groups.

This will also maintain all the different side of Donor's information that can help to a required patient.

### **1.3.2 Scope:-**

The framework is utilized for keeping up with all the interaction and exercises of blood donation centre administration framework.

The framework can be stretched out to be utilized for keeping up with records of medical clinic, organ gift and other comparative areas.

Keeping up with record for donor is simple.

### **1.3.3 Applicability:-**

Project will applicable in Hospitals, Health-care centres, donor to go their respective donation venue.

It will also applicable to medical centre.

It will help blood camp venue to records all donor information

It will applicable to medical that are in the blood bank management system connected.

It is easy to maintain also and all faculty medical centre can approved this.

### **1.4 Achievement:-**

By this project we get that knowledge about Blood Bank Management system which based-

The donor's information can know who have donated blood rather than going and checking inside the blood bank store room.

It is also help to store or record data in proper way and store the data in database.

It will help to maintain the donor details or record in database.

In this we can achieve the better Blood Bank management through this project.

This project is also going help to environment to reduce paper work, keeping the records files.

It is better to maintain system records rather than keeping files.

## Chapter 2

### Survey of Technology

#### **2.1 Available technologies:-**

Our requirement is to make Desktop Application.

##### **Desktop Application:-**

- It is a standalone application
- This application runs on computer system using RAM and ROM.
- Each desktop application perform different task like some are used to listen music or watch videos and some are used to make document or presentation while some are used to run desktop application on computer
- This application were developed to run on specific OS for e.g. Windows.
- And no internet is required for application.
- Some examples of desktop application are **VLC, GOM player, file explorer, Word, Excel etc.** and some application needs internet to run the application such as **Chrome, Firefox and Zoom** etc.

##### **C#:**

- C# is an object-oriented programming language.
- C# coding can be done with the help of visual studio.
- Programmers can build on existing code rather than repeatedly duplicating of the code.
- The reason for selecting C# is it makes project easy to develop, maintain, less coding, etc.
- C# the designing part is very easy just drags and drop.
- Have to do only background coding which helps controls work.
- Have varieties of database connectivity servers.
- Can create different types of pages and link them easily.

### **2.5.1 Asp.Net:-**

In market there are various advancements, for example, PHP, HTML5 and a few others, ASP.NET of Microsoft has ended up being a famous choice. ASP.NET innovation can be utilized to make sites or web applications that can build the size of your business to a higher degree of progress.

ASP.NET is one among the best software frameworks developed by Microsoft. Engineers can construct a wide range of applications from the least difficult ones to the most mind boggling ones utilizing this programming stage. It is possible to build application with reusable codes using programming languages of C++, C#, JS through the use of ASP and XML.

There are plenty of good reasons to use ASP.NET when developing a website or an application.

Rapid, minimal expense, and huge language support are among the main advantages. ASP.NET is incorporated into the recognizable Windows server climate, requiring less arrangement and design than other web improvement stages that should be introduced and arranged independently. The ubiquity of ASP.NET makes online assets and gifted engineers simple to find. Asp.Net is a system that gives a bunch of instruments. Structures help for creating pages. It gives troubleshooting device. Asp.Net gives usefulness as an apparatus to building.

### **SQL SERVER:**

SQL server is a popular database in our market. SQL server deals with all the information base. It is smidgen simple to execution for our site. SQL server is a GUI device included with SQL server 2005 and later for arranging, endlessly dealing with all parts of SQL server.

This apparatus incorporates a content editorial manager and graphical instruments that work with articles and server capacities. This rendition of SQL server the board studio is likewise accessible in SQL server Express Edition called SQL server Management Studio Express.

SQLCMD is a command line application that ships with Microsoft SQL server and includes SQL server management features. This allows you to create and run SQL queries from the command line. It can also work as a scripting language for building and executing a set of SQL statements with a script. The script is stored as a .SQL file and is used to create a database schema during administration or database deployment.

### **.Net Framework:-**

Microsoft .NET is by far one of the best platforms for building robust, secure, and scalable **web or desktop applications**. Used by a majority of Fortune 500 companies, Microsoft .NET is very popular for large-scale applications. **Microsoft .NET** offers numerous advantages to both developers and end clients. While clients get completely useful and highlight rich applications

with instinctive client experience, engineers and creators get adaptability and dynamic elements to fabricate sites effortlessly.

- Better UI Controls
- Security
- Integration with other Microsoft Applications
- Stability & Scalability
- MVC Architecture
- Quick Deployment

## **2.2 Technologies used:-**

### **Purpose of choosing C# and .Net framework:-**

Microsoft made C# for Microsoft. Along these lines, there is no doubt why it is well known in building Windows applications. It makes your improvement interaction smooth, and functionalities, for example, C# trash assortment performs perfectly. Additionally, designers can rely on the local area backing and documentation of creating applications and projects that are intended for the Microsoft stage's engineering. While C# is the most popular language within the .Net framework, being a part of the .Net framework means C# can be used interchangeably and alongside other adjacent programming languages (Microsoft-developed and otherwise). Why does this matter? Because, unlike some more complicated languages, C# can be used instead of more demanding choices. That makes the ease of C# an asset, without limiting its desirability or applicability.

## Chapter 3

### REQUIREMENT AND ANALYSIS

#### **3.1 Problem Definition:-**

##### **Problem of Existing Systems:-**

Problem definition means it will be the difference between the new one and older one.

In the blood donation centre, shortage of intriguing blood bunch.

Inaccessibility of blood during crisis.

Passing because of absence of blood during activities.

Less mindfulness among individuals about blood gift and blood bonding.

The Blood Bank Management System project plans to make every one of the techniques mechanized and thusly with PC framework it tends to be all the more quick and precise.

This project is the high quality software to manage all these cumbersome jobs.

##### **Human Error:-**

1. The proposed system will provide accurate data with great efficiency.
2. Manual work is reduced to great extend so that there will be less confusion and human error.

##### **Problem solving for existing system:-**

1. In this blood bank system I will solve the problem i.e. creating a web based application to get the blood at the mean time.
2. This project will be helpful for donor and patient as well as doctors.
3. The project will also help to the patient to find out their required amount of blood and easy to find out their blood group.
4. It will also helpful for the environment and go with paper less.

5. For this Desktop application the main thing is to save the time and it will help to patient and doctor.

### **3.2 Requirement Specification:-**

The framework gives security highlights through username-secret phrase coordinating where just approved client can get to the framework with various approval level. Administrator Input:- Username, Password Output: - Invalid or Update Donor Details, logout Profile.

Registration: This allows only admin to register donor details. Input:-Donor id, Name, Date of Birth, Sex, Blood Group, Address, Contact Number, Email Address, City .

Output: - Effectively Registered. Blood donation centre Management System. The blood donation centre Admin can deal with the blood stock beginning from the blood assortment, to blood screening, handling, capacity, blood bottle accessibility through this framework. Each interaction or work process can be followed from the data set. The Admin needed to check the accessibility of blood bottles at whatever point the blood amount is underneath its standard level. The executives the records of all givers and their set of experiences are kept in one incorporated data set and hence decreasing copy information in the data set. The record of gift is kept up with by the framework.

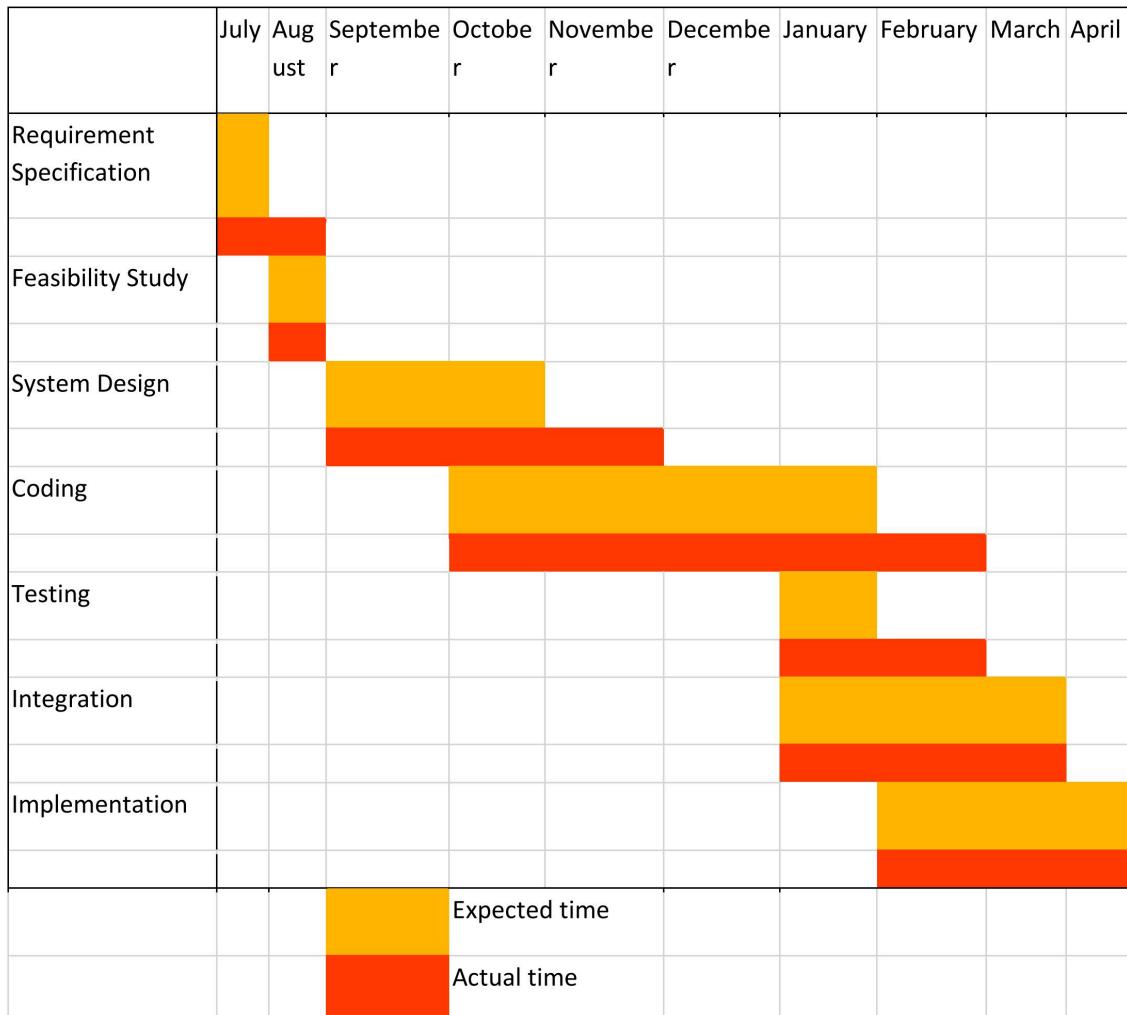
### **3.3 Planning and Scheduling:-**

#### **Gantt chart:-**

It is a bar chart that represents a project schedule. It shows the time and end time of the elements of the project. It is used to display the current planning status. Gantt charts contain helpful tools for evaluating and planning of the complex projects. They help to plan out responsibilities that need to be completed. Help you find a critical route for a project where you must complete it on a specific date.

## GANTT CHART

Stage and who is the task owner. This is useful to keep tasks on track.



### **3.4 Software Requirements:-**

1. Visual Studio.
2. Ms SQL Server Management Studio 18.

### **3.5 Hardware Requirements:-**

1. Operating System: Any Operating System
2. RAM: Minimum 2GB
3. Storage: Minimum 20GB

### **3.6 Functional Requirements in Words:-**

ID: FR1

Title: Run the exe file

Description: User will have to run the BLOOD BANK MANAGEMENT SYSTEM exe file by double clicking it to open the application.

ID: FR1

Title: Registration

Description: User needs to do the registration by providing their username and password and again confirm the password.

ID: FR3

Title: Login

Description: After registration, user login into the application login page their username and password.

ID: FR4

Title: Add Donor Details

Description: The user add the details of donor like name, Fname, dob, mobile no., gender, email, bloodgroup, city and address.

ID: FR5

Title: update details

Description: In these user can update the data of donor.

ID: FR6

Title: All details

Description: In these the user can see or print the all donor details.

ID: FR7

Title: Search blood donor.

Description: In these the user can search donor by their city name or blood groups.

ID: FR8

Title: stock.

Description: In these the user can see the blood stock as well as quantity of a particular blood groups.

ID: FR9

Title: Increase.

Description: In these the user can increase the quantity of a particular blood group.

ID: FR10

Title: Decrease

Description: In these the user can decrease the quantity of a particular blood group.

ID: FR11

Title: Blood details.

Description: In these the user can print or see all data of blood groups quantity.

ID: FR22

Title: Delete Donor information.

Description: In these the user can delete the all information of the particular donor by searching theirs ID.

ID: FR13

Title: Logout.

Description: In these the user can logout or exit from the application.

## Chapter 4

## System Design

### **4.1 Basic Models:-**

There is only one module in this system. i.e.-

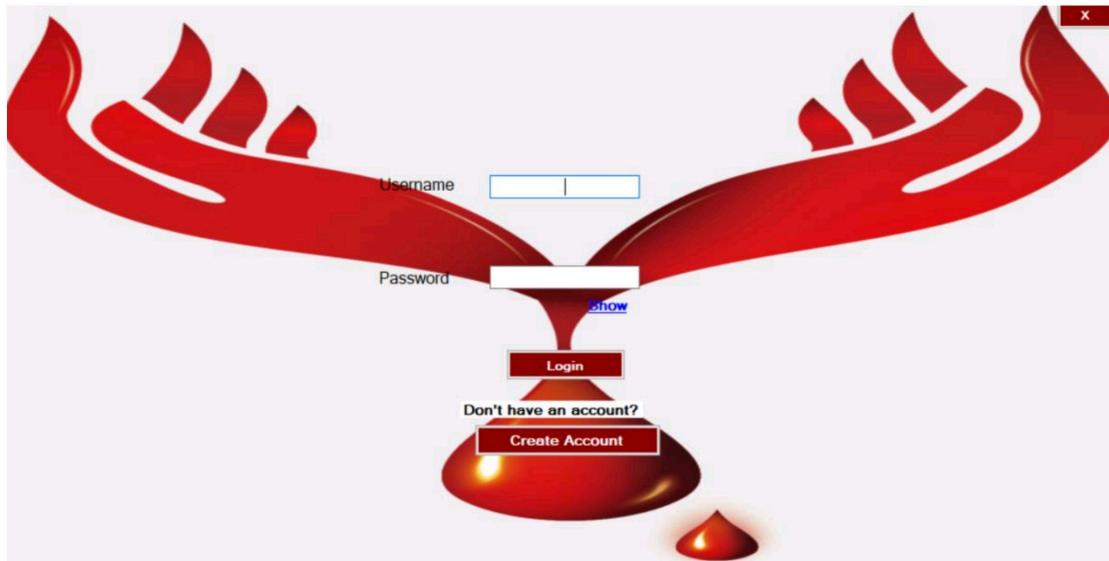
User Module: This module is for the user. The user is permitted to only view the data. They do not have the authority to make modifications in the system. The functions allowed for the user are:

Making accounts and login with the same account.

Viewing donor name and blood group based on the patient required.

Checking the records which blood group is available we have.

### **User Interface Design:-**





## 4.2 Data Design:-

Database is planned in such a way that it manages huge accounts of data and information. To manage data it requires both, definition of structure and the capacity of the structure to handle the data. In this context the data can be structured and controlled.

### Defining the data:-

Information as an overall idea alludes to the way that some current data or information is addressed or scrambled with a specific goal in mind that is appropriate for better use or handling.

### Objective of data design:-

The primary goal of information base planning is to create coherent and actual plans models of the proposed data set frameworks.

### Scheme Design:-

User	
Username	VARCHAR(50)
Password	VARCHAR(50)

### **SCHEMA DESIGN**

This is a model of my database. It has an entity by the name of user that is used to store user details. The user will be authenticated wherever he/she tries to login into the system, and the details will be validated when the user registers for the first time.

### **Data Integrity and constraints:-**

When building database. Care needs to be taken about data security and how to project it. An effective database enforces the integrity of data wherever possible. Data integrity risks are:

A user attempts to enter a data outside an acceptable range.

A user attempts to pass in a mobile number in the incorrect format.

A user tries to change a privacy key value if a foreign key currently exist in a linked table referring to that value.

Which transferring data the developer mistakenly attempts to insert the data in between two databases into the wrong tables?

A user tries to remove a record in the tables a separate table applies to the record as evidence of a relationship.

### **4.3 Class Diagram:-**

The class are used to identify and classify the objects which constitute a system.

It also includes the important attributes of the objects which must be captured.

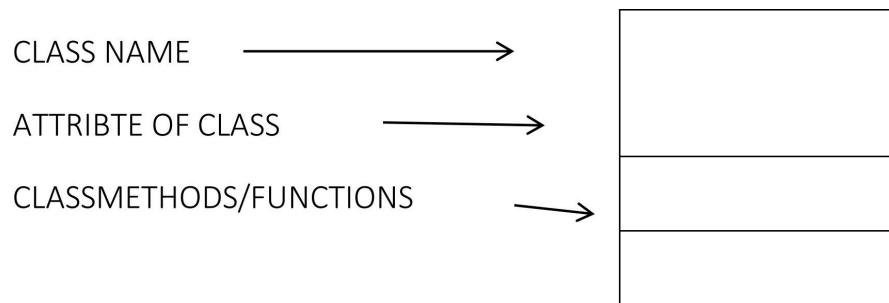
### **Class:-**

It is a collection of objects of same type.

### **Relationship:-**

A naturally occurring association among specific things.

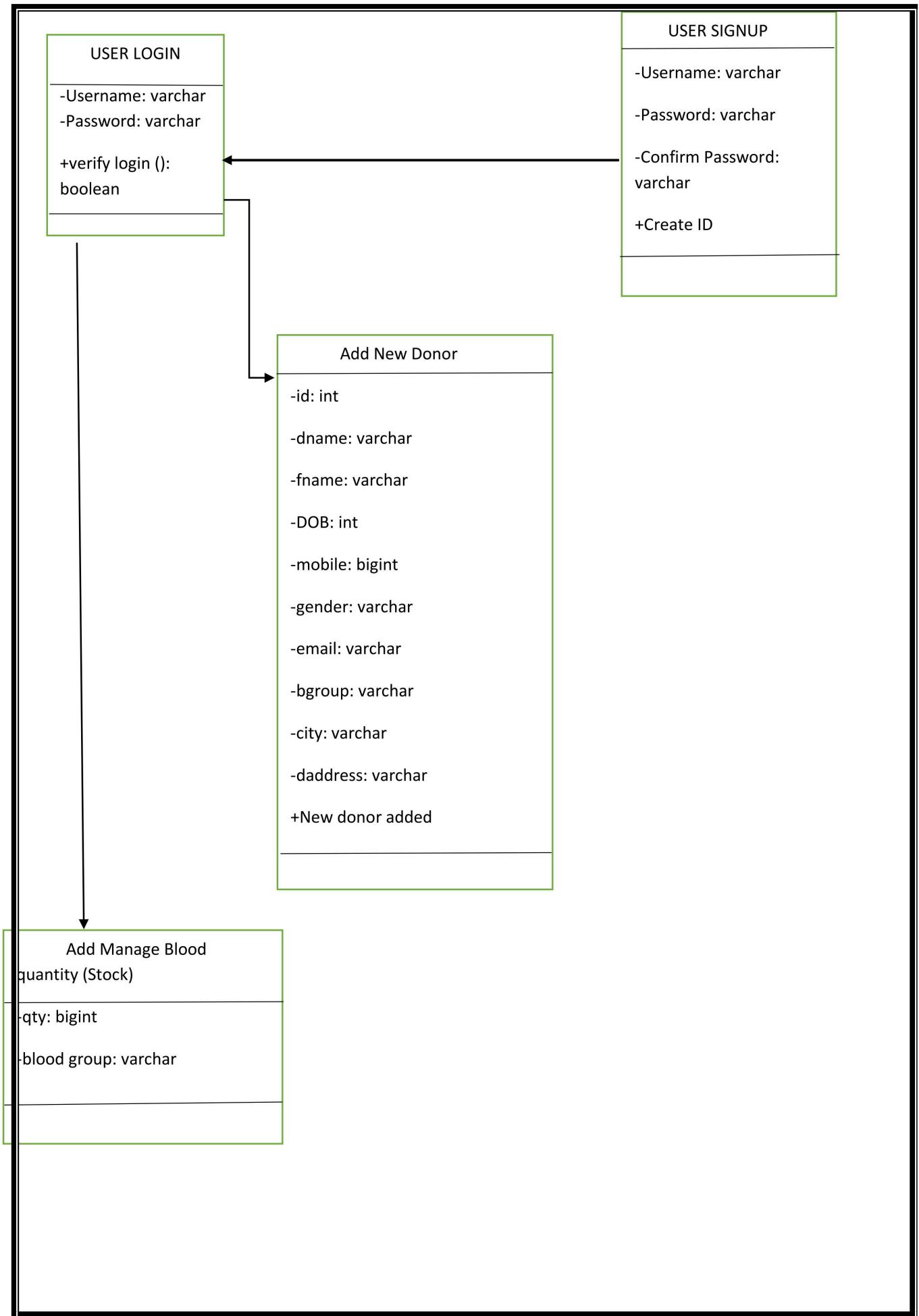
### **Class Diagram Notation:-**



### **Description:-**

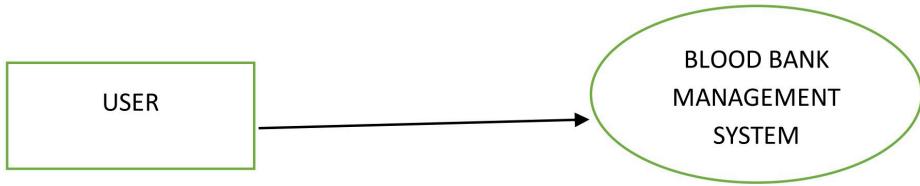
It is a model which is used to show the classes constituting a system and their inter-relationship. It is based on UML (Unified Modelling Language).only the important attributes and method are shown in class diagram. In the initial period of analysis, the important attributes of the classes, which must be captured and the functionalization provided by the class may not be very clear.

As the analysis progress, the attributes and methods may be added. If more focus is on interrelationships of classes, then the attributes and methods may not be shown in the class diagram.

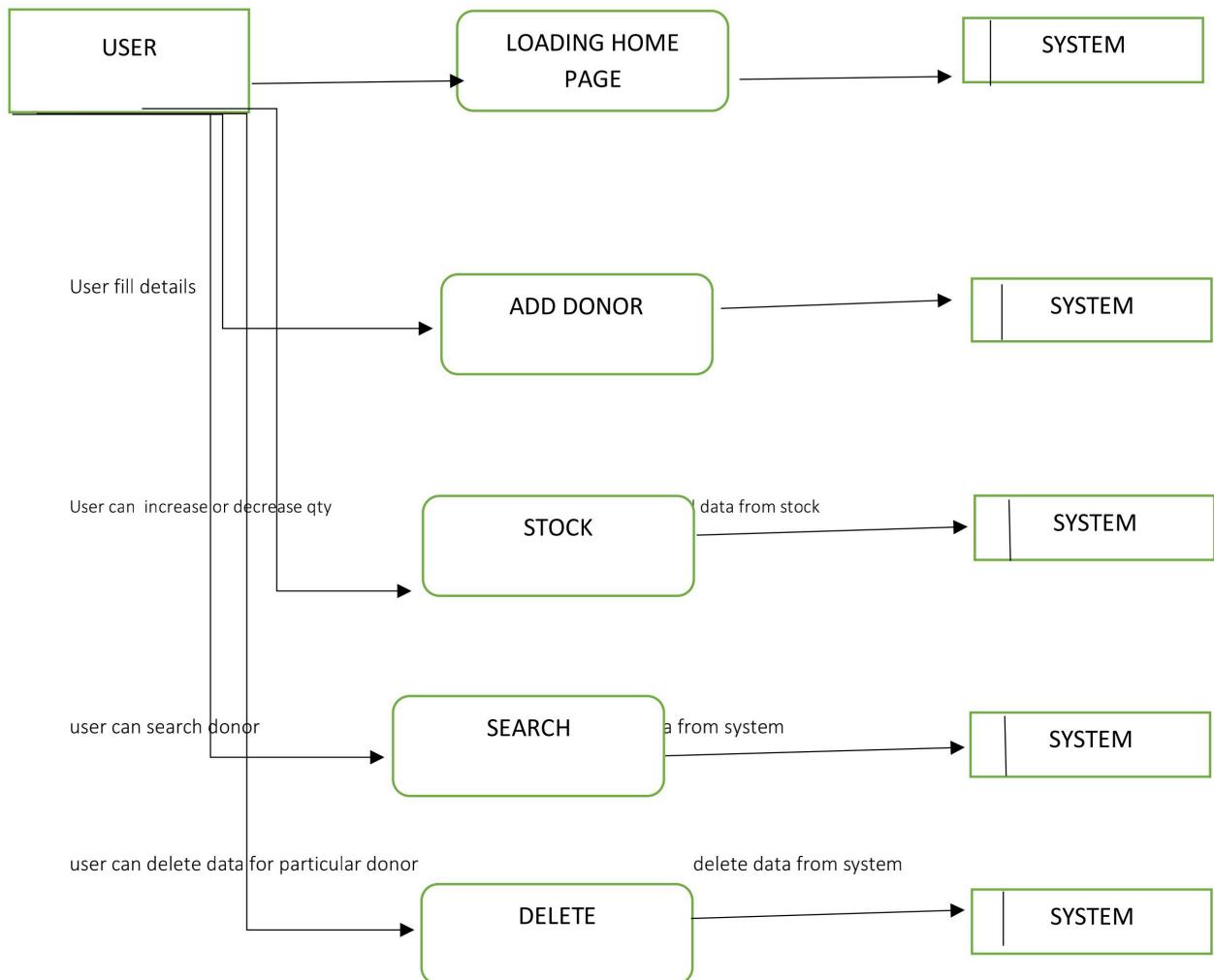


#### **4.4 Data Flow Diagram (DFD):-**

A Data Flow Diagram is an organized assessment and design apparatus that can be utilized for flowcharting instead of, or in relationship with, data orientated and way orientated frameworks flowcharts. A DFD is an organization that portrays the float of measurements and the strategies that change, or redesign, insights for the length of a contraption. This community is built by way of the use of a set of symbols that don't imply a physical implementation. It has the purpose of clarifying gadget necessities and identifying most important alterations that turns into packages in system layout. So, It's miles the place to begin of the layout segment that functionality decomposes the requirement specs right down to the bottom degree of element.



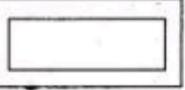
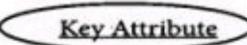
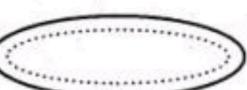
**0- Level DFD**

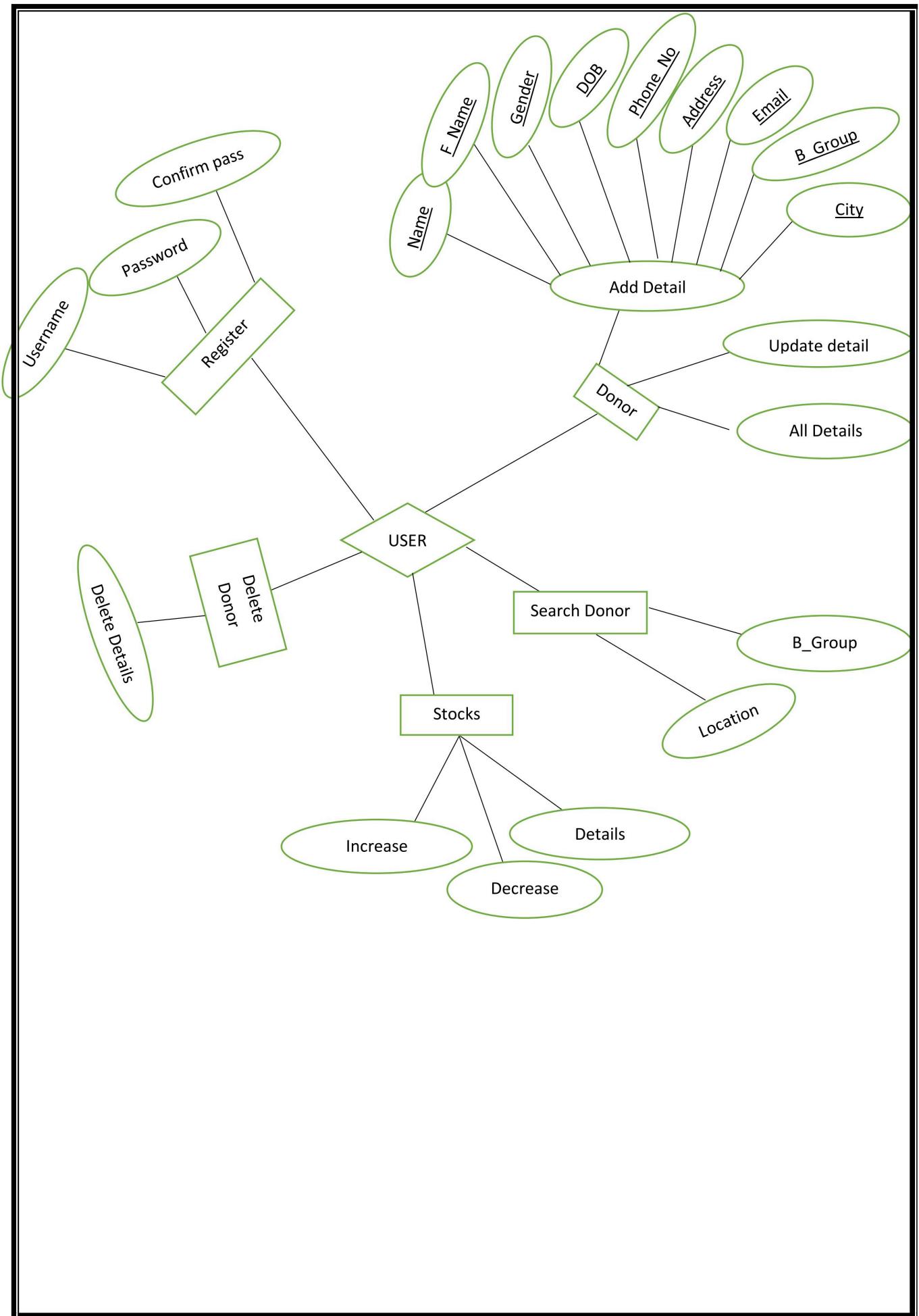


User Level DFD 1 Diagram

## 4.5 Entity Relationship Diagram:-

An E-R diagram model, additionally called an entity relationship(ER) chart, is a graphical portrayal of elements and their connection to one another, commonly utilized in figuring as to the association of information inside database or data frameworks. An element is a bit of information an item or idea about which information is put away.

ER Component	Description (how it is represented)	Notation
Entity – Strong	Simple rectangular box	
Entity – Weak	Double rectangular boxes	
Relationships	Rhombus symbol - Strong	
between Entities	Rhombus within rhombus – Weak	
Attributes	Ellipse Symbol connected to the entity	
Key Attribute for Entity	Underline the attribute name inside Ellipse	
Derived Attribute for	Dotted ellipse inside main ellipse Entity	
Multivalued Attribute	Double Ellipse for Entity	



## 4.6 Use Case Diagram:-

A utilization case outline is easiest portrayal of client's contact with the framework that shows the relationship between the client and the different use cases in which the client is involved. The utilization case graph can distinguish various sorts of program clients and different use case and will frequently be related with different kinds of chart.

Use case outline is utilized to recognize the "utilizations" or use instance of the new framework at the end of the day, to distinguish how the framework will utilized. The Use Case Diagram is basically an expansion of the Event Table.

### Use Case:-

It depicts a movement the framework completes because of an occasion.

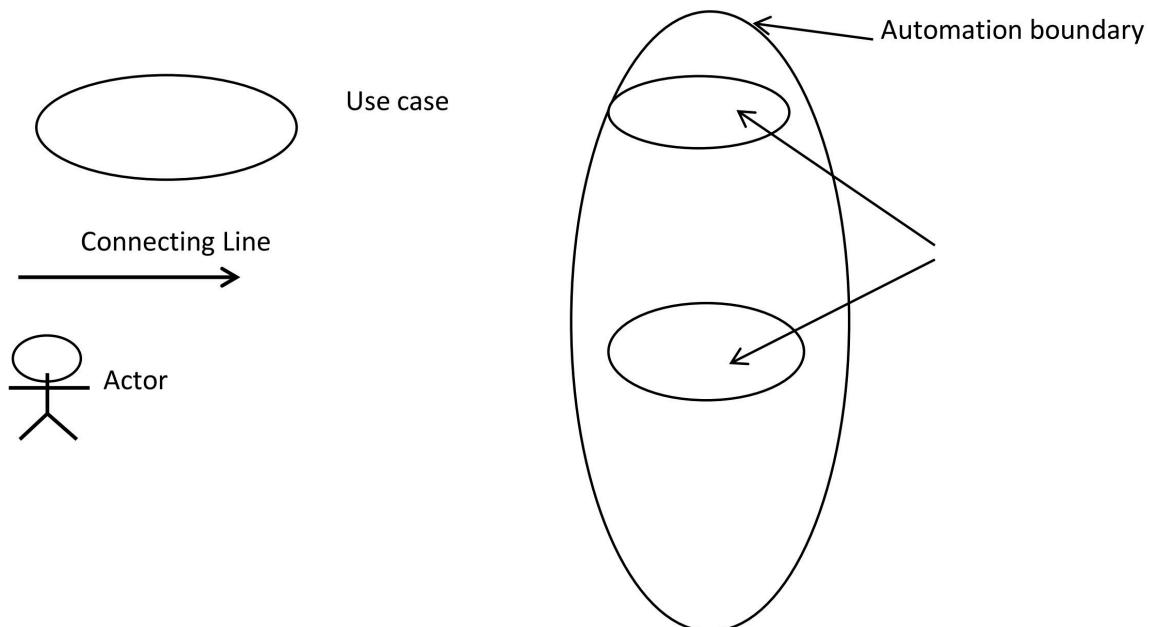
### Actor:-

In UML, the person involved is called an actor. An actor is always outside of the Automation.

### Connecting Line:-

The bolt is utilized to show which Actors take an interest in which use cases

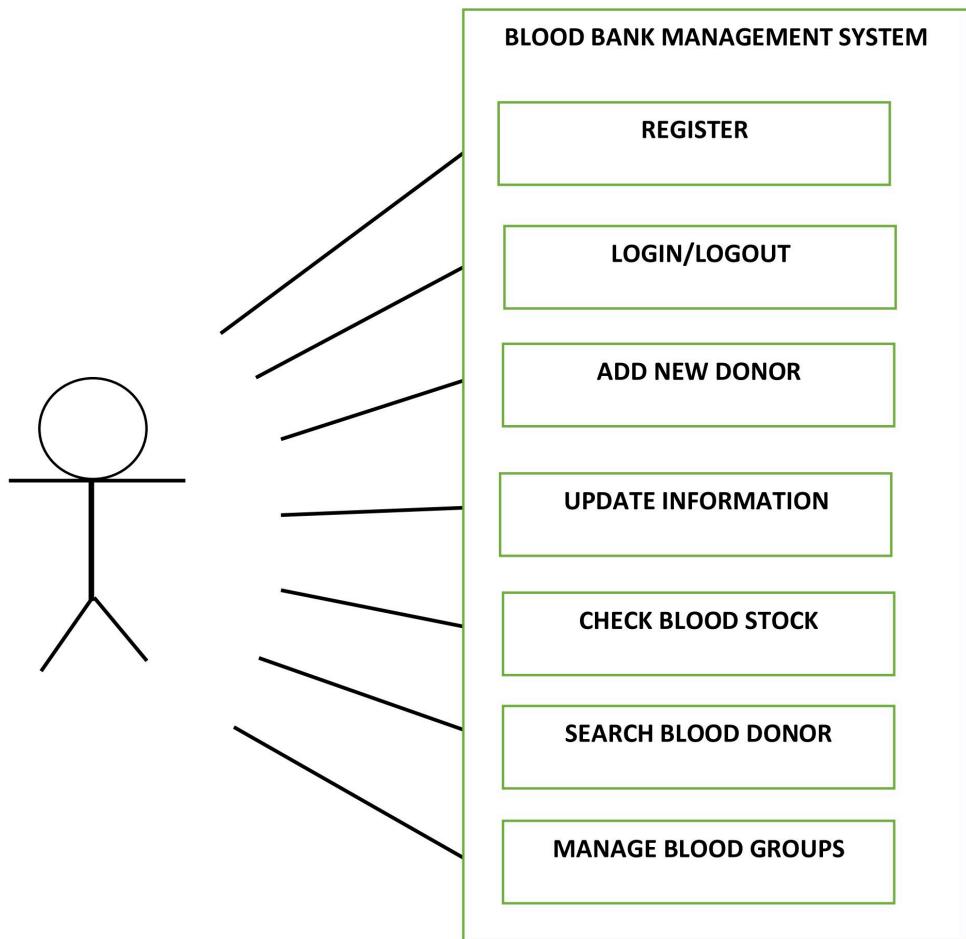
### Use case diagram notation:-



### Description:-

The object Oriented approach uses the term use case to describe an activity the systems carries out in response to an event. One can think of a use case as or situation where the

system is used for some purpose. A use case diagram is a convenient way to document the function that the system must support. Sometimes a single, comprehensive diagram is used to describe the entire system. At other times, a set of smaller use case diagrams make up the use case model.

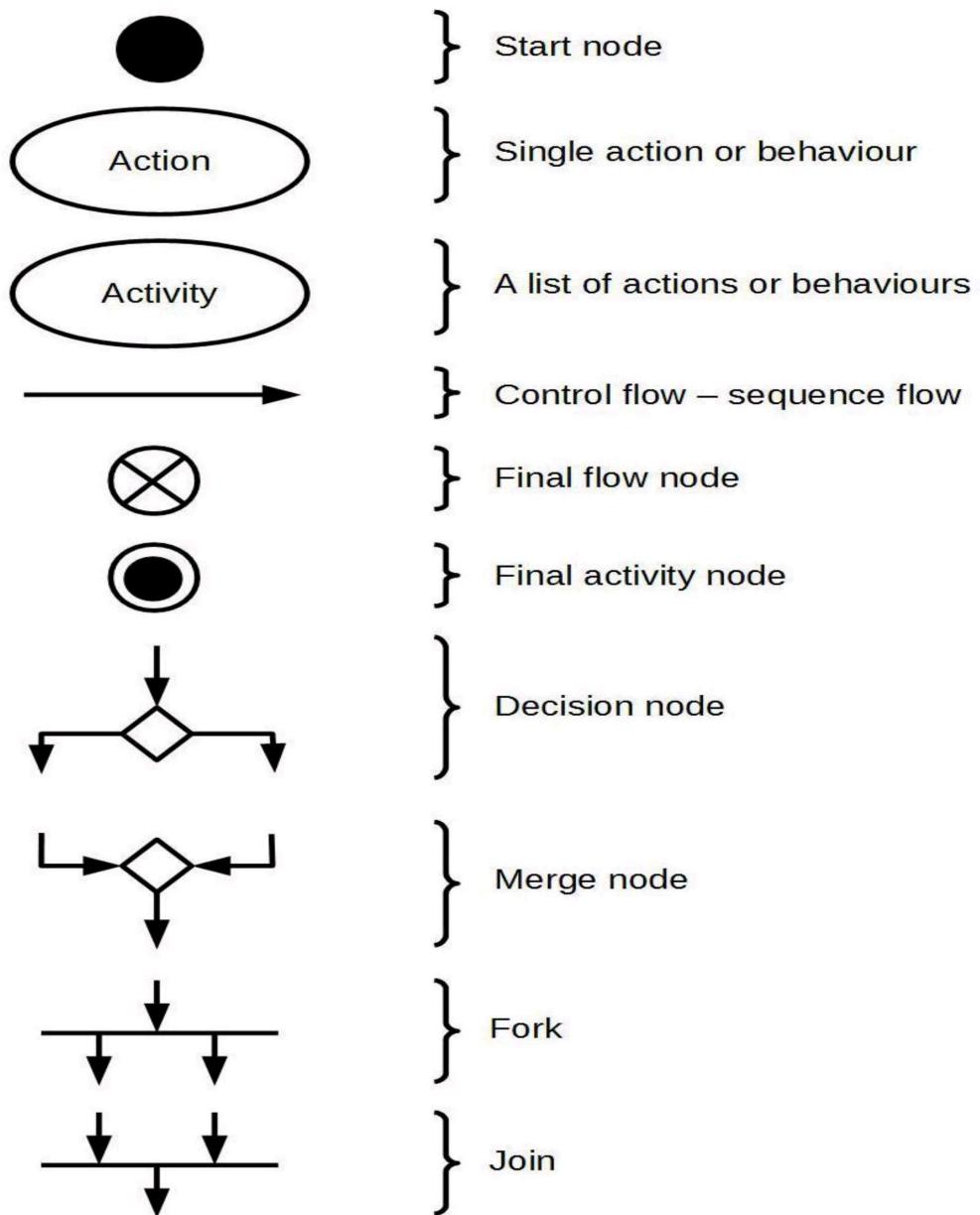


#### 4.7 Activity Diagram:-

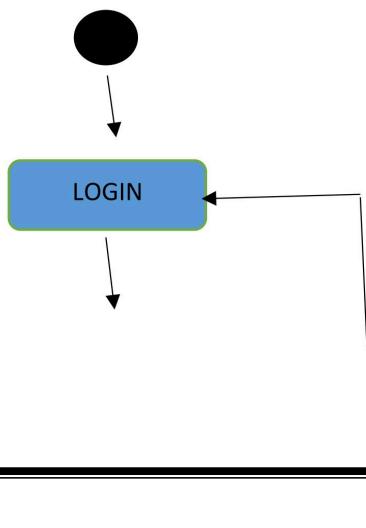
Activity diagram is another vital diagram in UML to explain the dynamic components of the machine.

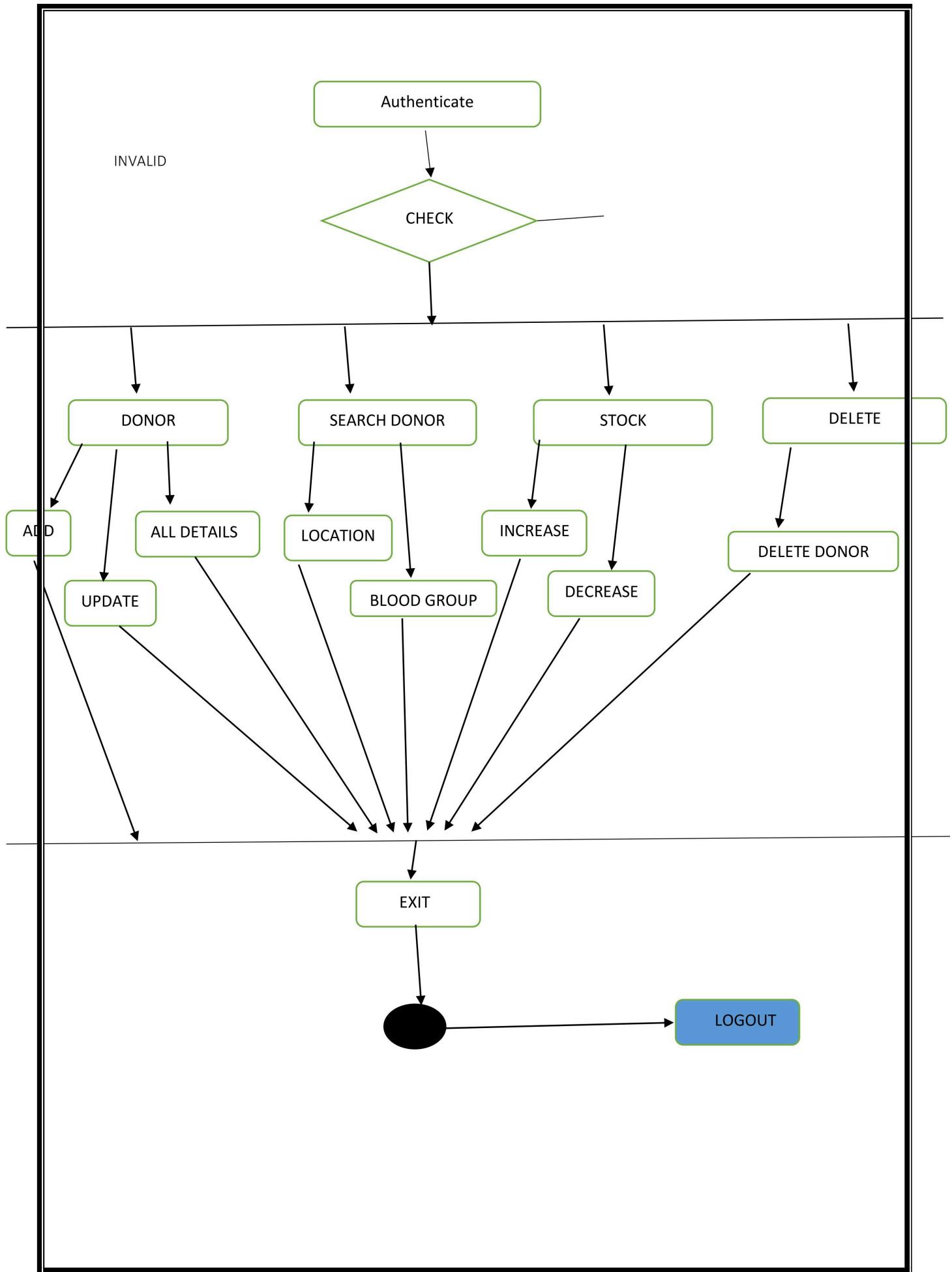
Movement outline is fundamentally a flowchart to represent the accept circumstances for what they are starting with one interest then onto the next action. The interest might be characterized as an activity of the framework.

The oversee stream is attracted from one activity to some other. This Go alongside the skim can be successive, expanded, or simultaneous. Activity diagrams address all kind of drift control via using specific elements including fork, be part of, and many others.



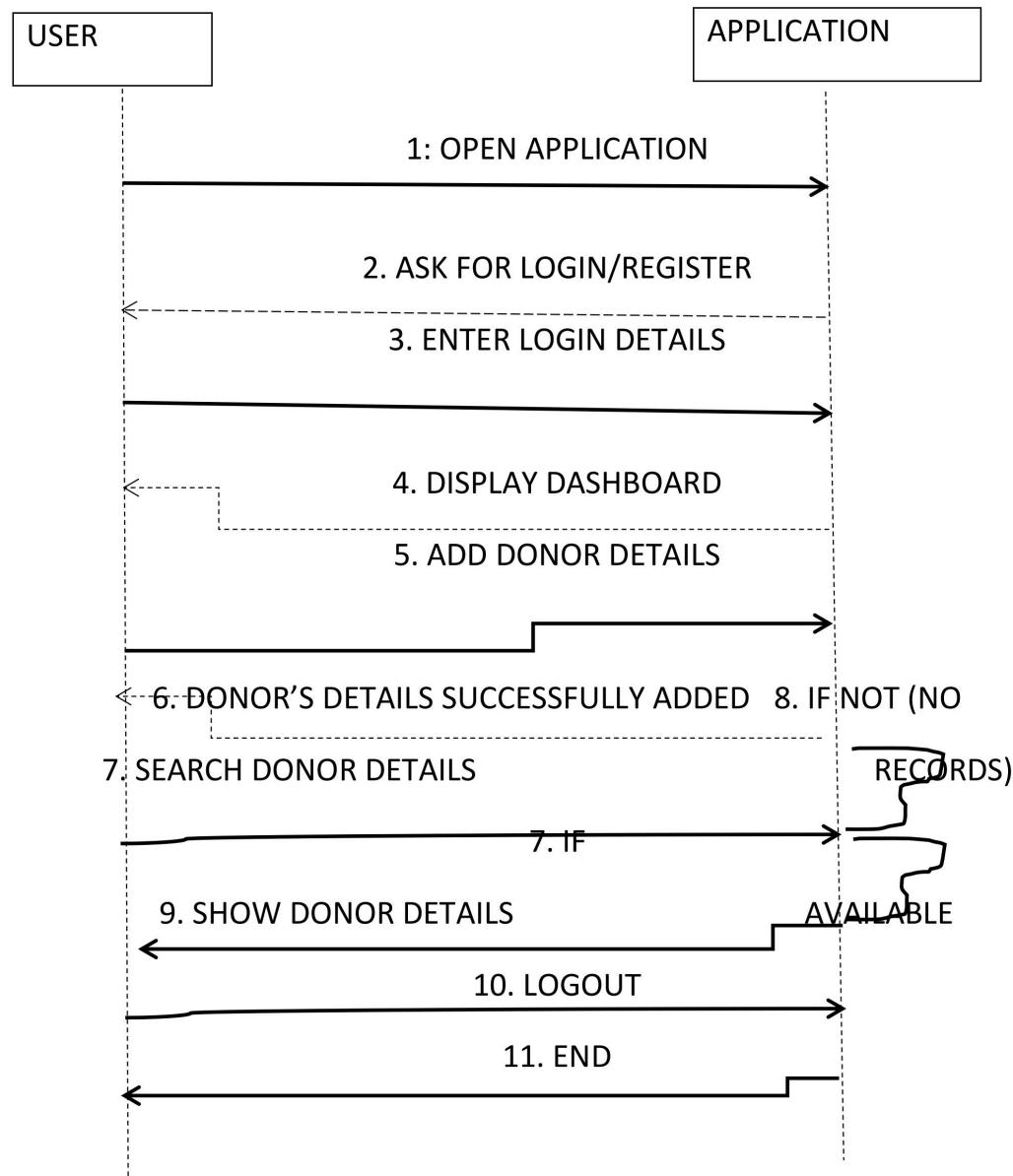
#### Activity Diagram for User Side:





#### 4.8 Sequence diagram:-

An arrangement outline just shows joint effort between objects in a successive request for example the request which these connection occur. We can likewise utilize the terms occasion outlines occasion circumstance to indicate a succession chart.



#### 4.9 System Flowchart:-

A Flowchart is used to show how the process of system, computer and all algorithms are used. The flowchart is widely used in multiple methods to document that flow in easy way to understand or clear that diagram or algorithms.

Flowchart is used to represent or visualization the data flow, this flowchart are used to explain big algorithms and codes.

It can help to organize the visualize picture, thinking and it will helpful in the following ways:-

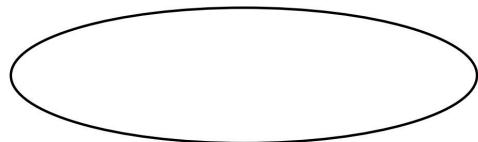
It will help to organize code.

It will help to visualize the flow of code within the program.

It shows the structure/product.

### **There are flowing notations are used in flow chart:-**

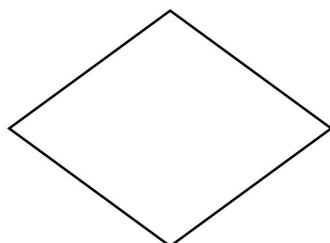
Terminal/Terminator



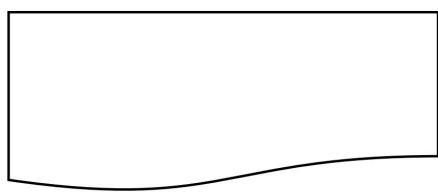
Process



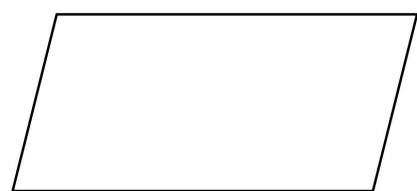
Decision



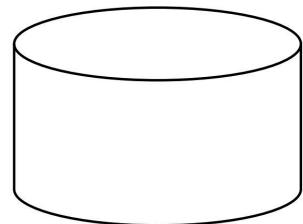
Document



Data or Input/output



Store Data



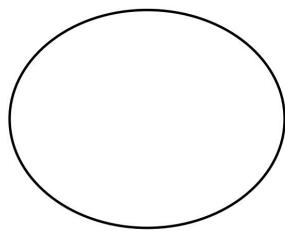
Flow Arrow



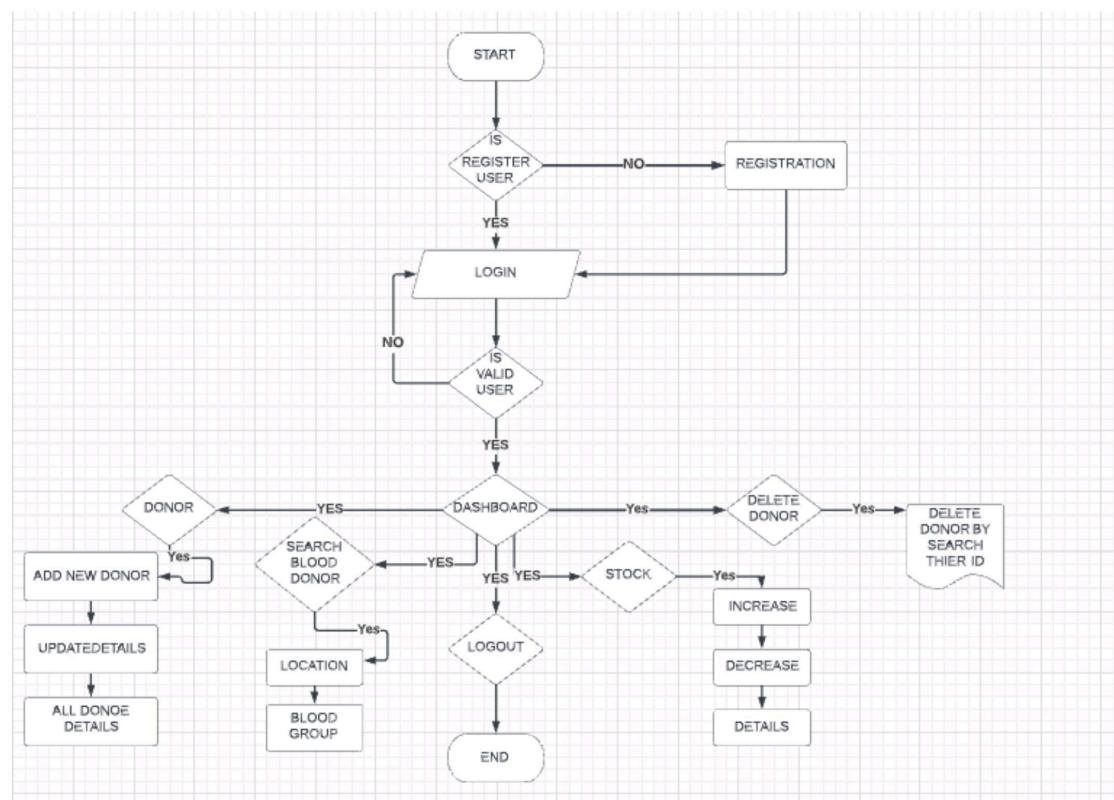
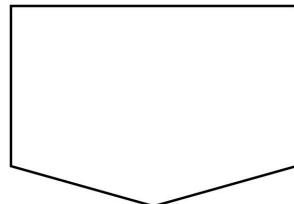
Comment or Annotation



On-page connector/reference



Off-page connector/reference



## Chapter 5

### Implementation and Testing

#### **5.1 Implementation Approaches:-**

A software implementation method is an analytically structured approach to efficiently fill in a software based package or module within a workflow of an organizational structure or function framework of a specific end-user. To enforce a project means the task suggested in the application form with such a view to achieving project goals and producing outcomes and outputs. Lots of internal and external causes directs the implementation to success. Most important points of a well implemented project are a project team and effective monitoring of project progress.

#### **5.2 Coding Details and Code Efficiency:-**

##### **5.2.1 Code Details:-**

C# is an object-oriented, **component-oriented** programming language. C# provides language constructs to directly support these concepts, making C# a natural language in which to create and use software components. Since its origin, C# has added features to support new workloads and emerging software design practices. At its core, C# is an **object-oriented** language. You define types and their behaviour.

C# emphasizes **versioning** to ensure programs and libraries can evolve over time in a compatible manner. Aspects of C#'s design that were directly influenced by versioning considerations include the separate virtual and override modifiers, the rules for method overload resolution, and support for explicit interface member declarations.

##### **5.2.1.1 Database Connectivity:-**

SqlCommand in C# grant the client to request and send the orders to the informational index. SQL not entirely settled by the SQL affiliation object. Two techniques are utilized, ExecuteReader strategy for aftereffects of inquiry and ExecuteNonQuery for addition, Update, and erase orders. The technique is best for the various orders.

**Connectivity Code:**

```

1  using System;
2  using System.Collections.Generic;
3  using System.Data;
4  using System.Data.SqlClient;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8  using System.Windows;
9  using System.Windows.Forms;
10 using MessageBox = System.Windows.MessageBox;
11
12 namespace WindowsFormsApp1
13 {
14     class function
15     {
16         ...
17         protected SqlConnection getConnection()
18         {
19             SqlConnection con = new SqlConnection();
20             con.ConnectionString = ("Data Source = (LocalDB)\MSSQLLocalDB; Initial Catalog = BBMS; Integrated Security = True");
21             return con;
22         }
23
24         public DataSet getData(String query)
25         {
26             SqlConnection con = getConnection();
27             SqlCommand cmd = new SqlCommand();
28             cmd.Connection = con;
29             cmd.CommandText = query;
30             SqlDataAdapter da = new SqlDataAdapter(cmd);
31             DataSet ds = new DataSet();
32             da.Fill(ds);
33         }
34
35         public void setData(string query)
36         {
37             ...
38         }
39     }
40 }

```

## 5.2.2 Code Efficiency:-

Code efficiency is an overall idea used to portray the dependability, speed and programming techniques utilized in planning codes for an application.

Some are the ways to achieve a good code efficiency:

- Remove unwanted and unnecessary code.
- To execute the significant algorithm utilizing the best keywords, data types and variables, and other accessible programming ideas.
- To ensure the fastest possible completion time for the algorithm.
- Use the memory and non-volatile storage optimally.



### **5.3 Testing Approaches:-**

Since we accept that our work will contain mistakes, we should all double check our own work. In any case, since specific mistakes are the consequence of broken assumptions and vulnerable sides, we can make similar blunders while surveying our own work as we did when we finished it. Thus, we were unable to see the inadequacies in our work.

The testing is done through all the phase of the advancement life cycle. It is a confirmation and approval approach (V&V Model).

#### **Verification:-**

Confirmation is the way toward surveying work-things from the headway stage to check whether they meet the predefined essentials. Check guarantees that the part is implicit agreement with the set-up guidelines and standards.

#### **Validation:-**

Approval is a system used toward the completion of the improvement stage to ensure that the thing meets the foreordained necessities. To put it another way, to guarantee that the thing meets the client's necessities.

To guarantee that the framework is exact and proficient, testing and execution go inseparably. The most well-known sort of testing is finished by the client who runs the program. It is important to make an authentic application. In this period of the product advancement life cycle, the entirety of the bugs and blunders from the past advance are fixed.

System testing aids the revelation of defects that were not recognized during before phases of the improvement life cycle. The program is deliberately raced to discover mistakes. Stress testing, limit esteem examination, and other comparative methodology fall under this class. To test, different experiments are made.

There are different ways to test this software:

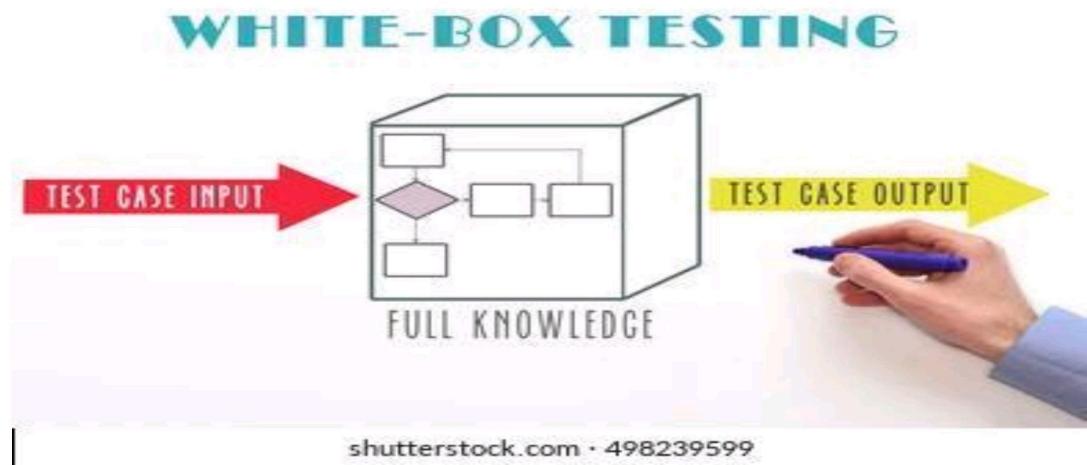
#### **5.3.1 Unit Testing:-**

This kind of testing is performed by designers before the arrangement is given over the testing group to officially execute the experiments. It is performed on the singular units of source code relegated region. It is a level of the product testing process where individual units/parts of a product/framework are tried. The objective of unit testing is to detach each piece of the program and show that singular parts are right concerning necessities and usefulness.

Types of Unit Testing:

White Box Testing: Testing depends on an investigation of the inside construction of parts framework. White Box Testing otherwise called clear box testing, Glass Box Testing,

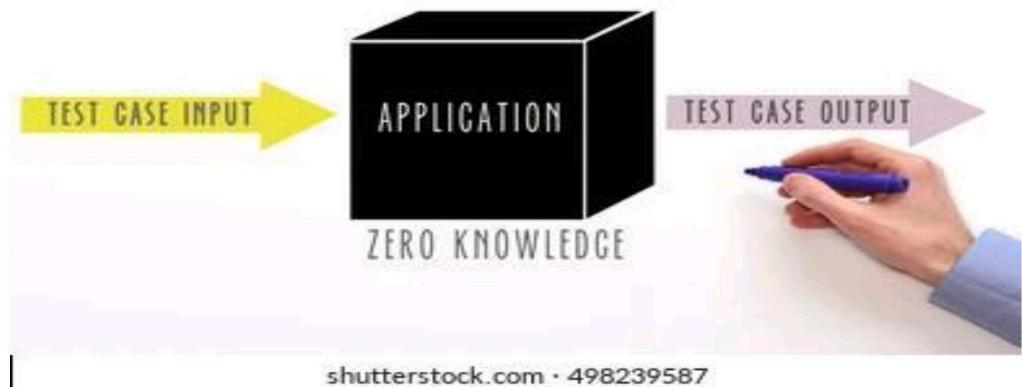
Transparent Box Testing and organized Based testing. It very well may be applied at all degrees of SDLC. Tests depend on the inclusion of code explanations, Branches, ways, and conditions.



**Black Box Testing:** It is methodology for programming testing that examination the helpfulness of an application without researching its internal developments and capacities. This methodology is applied at every level of programming testing: unit, blend, system and affirmation. Tests rely upon necessities and value. It is done to test usefulness of the program additionally called as "Social Testing".

The Tester in this case, has a set of input values and get desired results.

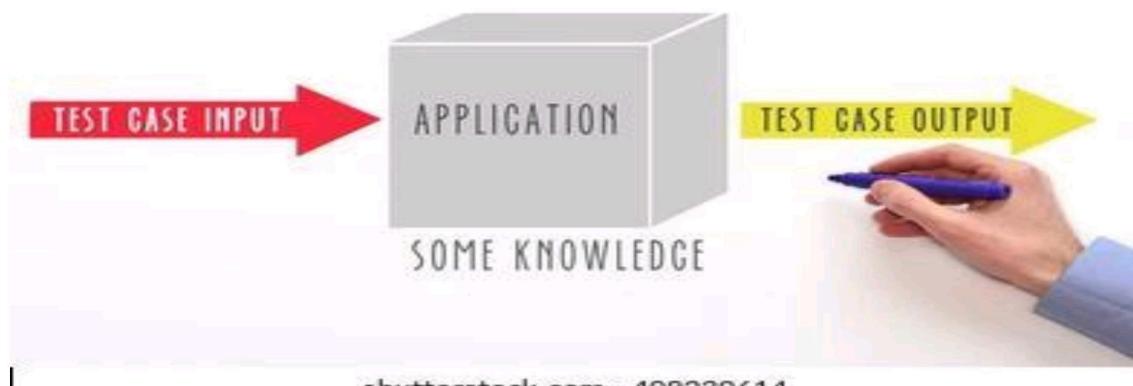
## BLACK-BOX TESTING



shutterstock.com • 498239587

Grey Box Testing: The testing wherein analyser have restricted information about the inner design of the application. It is something which is in the middle of White and Black Box test. It is to some extent transparent implies you can see into it yet not completely and appropriately

## GRAY-BOX TESTING



shutterstock.com • 498239614

The techniques of unit testing applied in the Blood Bank Management System are as follows:

- All the modules are tested individually.
- All the errors are founded and corrected.
- It was seen that the buttons were properly functioning.
- It was checked that devices would run when the input is passed.

### **5.3.2 Integration Testing:-**

The motivation behind the coordination testing is to uncover deficiencies in the connection between incorporated units. It is the most common way of testing the point of interaction between programming units and modules. This testing of united bits of an application to conclude whether they cooperate precisely.

There are three approaches in Integration testing:

- **Big Bang Approach:** In Big Bang testing approach, it consolidates every one of the modules once and checking the usefulness after finishing of individual module testing.
- **Top-Down Approach:** In Top-Down approach, Testing happens through and through. Undeniable level modules are tried first and afterward low-level modules lastly coordinating the low level modules to significant level to guarantee the framework is filling in as expected.
- **Bottom-Up Approach:** In granular perspective, Testing happens from base to up. Most reduced level modules are tried first and afterward undeniable level modules lastly incorporation the general modules to low even out to guarantees the framework is functioning as expected.

The techniques of Integration Testing Approach in the blood bank management system are as follows:

- This project uses top-down approach.
- It verifies the various decision points.
- The feasibility is determined at a very early stage of the development lifecycle.
- The major flows are detected by taking input from users.

## **5.4 Test Case Design:-**

### **Case 1: For Login Details:-**

Launching Application

1. Purpose: Launch the project
2. Opens the Login Page
3. Shows three buttons Login, Signup.

Clicking on signup button goes to the signup Page.

SR.NO	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	Desktop	Home screen of BBMS app	The home screen of the app is displayed	Pass
2	Enter invalid Username and password	Username:MOHIT12 password: qhatgs	Invalid username or password	Invalid username or password	Pass
3	Enter valid Username and Password	Username:MOHIT12 password: SHARMA12	Login Successful!	Dashboard displayed	Pass

### **Case 2: For Signup Details:-**

Purpose: Test that users can register with the proper username and password

User's Login credentials

Enter Username.

Enter password.

And Enter Confirm Password.

Click on Register button.

SR.NO	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	Desktop	Home screen of BBMS app	The home screen of the app is displayed	Pass
2	Enter Username and password mismatch confirm Password	Username:MOHIT12 password: SHARMA12 Confirm Password:SHRAMA1	Password did not matched!	Password did not matched!	Pass
3	Enter Username and Password Same confirm password	Username:MOHIT12 password: SHARMA12 confirm password:SHARMA12	Register Successful!	Register Successful!	Pass

### **Case 3: Donor Details:-**

SR.NO	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Click on Add New Donor		Add form displayed	Add form Displayed	Pass
2	Fill the Details	Some filed is empty	Fill the details!	Fill the details!	Pass
3	Fill the Details	Filled correctly as required	Donor Added Successful!	Donor Added Successful!	Pass
4	Update donor details (Search Id)	Incorrect id	No record exist!	No record exist	pass
5	Update donor details (Search Id)	Filled correct id	Donor details displayed	Donor details displayed	pass
6	Update donor details	Changed the information	Updated Successful!	Updated Successful!	pass

## 5.5 Core Segment Code:-

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for the SIGNUP.cs file. The code handles button click events for registration and sign-up. It includes validation logic to ensure all fields are populated and password matches. If successful, it inserts data into a database table named 'signup'. The Solution Explorer on the right shows the project structure with various files like frmSignup.cs, function.cs, and Program.cs.

```
1 reference
17 public partial class SIGNUP : Form
18 {
19     InitializeComponent();
20 }
21
22 private void btnRegister_Click(object sender, EventArgs e)
23 {
24     if (txtUsername.Text != "" && txtPassword.Text != "" && txtConfirmPassword.Text != "")
25     {
26         string username = txtUsername.Text;
27         string password = txtPassword.Text;
28         string cfnpassword = txtConfirmPassword.Text;
29
30         if (txtPassword.Text == txtConfirmPassword.Text)
31         {
32             String query = "insert into signup_(username,password,cfnpassword)" + "values('" + username + "','" + password + "','" + cfnpassword + "')";
33             fn.setData(query);
34         }
35         else
36         {
37             MessageBox.Show("Password did not matched!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
38         }
39     }
40 }
41
42 }
43
44 private void label5_Click(object sender, EventArgs e)
45 {
46 }
47
48
49 private void button1_Click(object sender, EventArgs e)
50 {
51     frmSignup frm = new frmSignup();
52     frm.Show();
53 }
```

Sign Up Page:

```

    69     }
    70   }
    71 }
    72
    73 1 reference
    74 private void btnLogin_Click(object sender, EventArgs e)
    75 {
    76   if (txtUsername.Text == "")
    77   {
    78     MessageBox.Show("Enter the username");
    79   }
    80   else if (txtPassword.Text == "")
    81   {
    82     MessageBox.Show("Enter the password");
    83   }
    84   else
    85   {
    86     try
    87     {
    88       SqlConnection con = new SqlConnection("Data Source = (LocalDB)\MSSQLLocalDB; Initial Catalog = BBMS; Integrated Security = True");
    89       SqlCommand cmd = new SqlCommand("select*from signup_1 where Username=@username and Password=@password ", con);
    90       cmd.Parameters.AddWithValue("@username", txtUsername.Text);
    91       cmd.Parameters.AddWithValue("@password", txtPassword.Text);
    92       SqlDataAdapter da = new SqlDataAdapter(cmd);
    93       DataTable dt = new DataTable();
    94       da.Fill(dt);
    95       if (dt.Rows.Count > 0)
    96       {
    97         Dashboard db = new Dashboard();
    98         db.show();
    99         this.Hide();
    100      }
    101      else
    102      {
    103        MessageBox.Show("Invalid Username or Password,Please Try Again", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error);
    104      }
    105    }
    106  }
    107 }

    100 %  No issues found
  
```

Login Page:

```

    16     1 reference
    17     public Add_New_Donor()
    18     {
    19       InitializeComponent();
    20     }
    21
    22     1 reference
    23     private void btnClose_Click(object sender, EventArgs e)
    24     {
    25       this.Close();
    26     }
    27
    28     1 reference
    29     private void Add_New_Donor_Load(object sender, EventArgs e)
    30     {
    31
    32       String query = "select max(did) from newDonor";
    33       DataSet ds = fn.getData(query);
    34       int count=Int32.Parse(ds.Tables[0].Rows[0][0].ToString());
    35       labelNewID.text = (count+1).ToString();
    36
    37     }
    38
    39
    40
    41
    42
    43     1 reference
    44     private void btnSave_Click(object sender, EventArgs e)
    45     {
    46       if (txtName.Text != "" && txtFather.Text != "" && txtDOB.Text != "" && txtMobile.Text != "" && txtGender.Text != "" && txtMobile.Text != "" && txtFname.Text != "" && txtDname.Text != "" && txtDob.Text != "" && txtMobile.Text != "" && txtGender.Text != "")
    47       {
    48         String dname = txtName.Text;
    49         String fname = txtFather.Text;
    50         String dob = txtDOB.Text;
    51         Int32 mobile = Int32.Parse(txtMobile.Text);
    52         String gender = txtGender.Text;
    53
    54         fn.AddNewDonor(dname, fname, dob, mobile, gender);
    55
    56         MessageBox.Show("Record Added");
    57         this.Close();
    58       }
    59     }
  
```

Add new Donor Page:

```

1 reference
2 public DeleteDonor()
3 {
4     InitializeComponent();
5 }
6
7 reference
8 private void btnClose_Click(object sender, EventArgs e)
9 {
10    this.Close();
11 }
12
13 reference
14 private void btnSearch_Click(object sender, EventArgs e)
15 {
16    if(txtDonorID.Text!="")
17        query="Select*from newDonor where did='"+txtDonorID.Text+"'";
18    DataSet ds = fn.getData(query);
19
20    if(ds.Tables[0].Rows.Count!=0)
21    {
22        txtName.Text = ds.Tables[0].Rows[0][1].ToString();
23        txtFather.Text = ds.Tables[0].Rows[0][2].ToString();
24        txtDOB.Text = ds.Tables[0].Rows[0][3].ToString();
25        txtMobile.Text = ds.Tables[0].Rows[0][4].ToString();
26        txtGender.Text = ds.Tables[0].Rows[0][5].ToString();
27        txtEmail.Text = ds.Tables[0].Rows[0][6].ToString();
28        txtBloodGroup.Text = ds.Tables[0].Rows[0][7].ToString();
29        txtCity.Text = ds.Tables[0].Rows[0][8].ToString();
30        txtAddress.Text = ds.Tables[0].Rows[0][9].ToString();
31    }
32    else
33    {
34        MessageBox.Show("No Record Exist", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
35        txtDonorID.Clear();
36    }
37 }
38
39 reference
40 
```

Delete Donor Page:

```

1 reference
2 public partial class Stock : Form
3 {
4     function fn = new function();
5     string query;
6     reference
7     public Stock()
8     {
9         InitializeComponent();
10    }
11
12    reference
13    private void btnClose_Click(object sender, EventArgs e)
14    {
15        this.Close();
16    }
17
18    reference
19    private void Stock_Load(object sender, EventArgs e)
20    {
21        query = " select*from stock";
22        DataSet ds = fn.getData(query);
23        dataGridView1.DataSource = ds.Tables[0];
24    }
25
26    reference
27    private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintEventArgs e)
28    {
29        Bitmap bm = new Bitmap(this.dataGridView1.Width, this.dataGridView1.Height);
30        dataGridView1.DrawToBitmap(bm, new Rectangle(0, 0, this.dataGridView1.Width, this.dataGridView1.Height));
31        e.Graphics.DrawImage(bm, 0, 0);
32    }
33
34    reference
35    private void btnPrint_Click(object sender, EventArgs e)
36    {
37        printDocument1.Print();
38    }
39
40    reference
41 
```

Stock Page:

```

26
27
28
29     reference
30     private void btnSearch_Click(object sender, EventArgs e)
31     {
32         int id = int.Parse(txtDonorID.Text.ToString());
33         string query = "select*from newDonor where did=" + id + "";
34
35         DataSet dsfn = fn.getData(query);
36
37         if(ds.Tables[0].Rows.Count==0)
38         {
39             txtName.Text = ds.Tables[0].Rows[0][1].ToString();
40             txtFather.Text = ds.Tables[0].Rows[0][2].ToString();
41             txtDOB.Text = ds.Tables[0].Rows[0][3].ToString();
42             txtMobile.Text = ds.Tables[0].Rows[0][4].ToString();
43             txtEmail.Text = ds.Tables[0].Rows[0][6].ToString();
44             txtBloodGroup.Text = ds.Tables[0].Rows[0][7].ToString();
45             txtCity.Text = ds.Tables[0].Rows[0][8].ToString();
46             txtAddress.Text = ds.Tables[0].Rows[0][9].ToString();
47         }
48         else
49         {
50             MessageBox.Show("Invalid Id", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
51         }
52     }
53
54
55     reference
56     private void txtDonorID_TextChanged(object sender, EventArgs e)
57     {
58         if(txtDonorID.Text=="")
59         {
60             txtName.Clear();
61             txtFather.Clear();
62             txtDOB.ResetText();
63             txtMobile.Clear();
64             txtGender.ResetText();
65         }
66     }

```

### Update Donor Details Page:

```

26
27
28
29     reference
30     private void SearchDonorByBlood_Load(object sender, EventArgs e)
31     {
32         query = "select*from newDonor";
33         DataSet dsfn = fn.getData(query);
34         dataGridView1.DataSource = ds.Tables[0];
35
36     }
37
38     reference
39     private void txtSearchBlood_TextChanged(object sender, EventArgs e)
40     {
41         if(txtSearchBlood.Text!="")
42         {
43             query = "select*from newDonor where bloodgroup Like'" + txtSearchBlood.Text + "%'";
44             DataSet ds = fn.getData(query);
45             dataGridView1.DataSource = ds.Tables[0];
46         }
47         else
48         {
49             query = "select*from newDonor";
50             DataSet ds = fn.getData(query);
51             dataGridView1.DataSource = ds.Tables[0];
52         }
53
54
55     reference
56     private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintEventArgs e)
57     {
58         Bitmap bm = new Bitmap(this.dataGridView1.Width, this.dataGridView1.Height);
59         dataGridView1.DrawToBitmap(bm, new Rectangle(0, 0, this.dataGridView1.Width, this.dataGridView1.Height));
60         e.Graphics.DrawImage(bm, 0, 0);
61     }
62
63     reference

```

### Search Donor By Blood Group Page:

## CHAPTER 6

### Results and Discussions

#### **6.1 Test Reports:-**

Test Report summarizes testing outcomes in a formal way, which gives us a chance to evaluate test results quickly. It is an archive document that contains the evaluation approach in order, describes the environment for testing. It is a communication that is used among the Test Manager and the stakeholder. From the report generated at the close of the testing lifecycle; the stakeholder can recognize the condition, value of the product.



Test Report should be:

**Detailed:** It ought to provide a detailed summary of the testing operation, which showcases the testing performed on the product. No abstract information is included in this document.

**Clear:** The test report should contain that must have clarity.

**Standard:** It must follow a standard template.

**Specific:** Summarize the document in clear and concise points.

### 6.1.1 Project Information:-

Following is the overview of the system:

PROJECT OVERVIEW	
Project Name	Blood Bank Management system
Name of the Product	Blood Bank Management system
Product Description	Information storing about Donor and their blood groups
Project Description	In these application, it will store information about donors and also will show us about the blood groups. It will show stock that means quantity about bloods and as well as their blood group, we have available or not.
PROJECT DURATION	
Start Date	16/7/2020
End Date	25/3/2021

### 6.1.2 Test Objective:-

- The main objectives of the test case are as follows:
- Find the defects in the system at an earlier stage of the developing process.
- To have the confidence about the system end result and its performance.
- To ensure the quality of the system.
- To prevent defects in future.

## 6.2 User Documentation:-

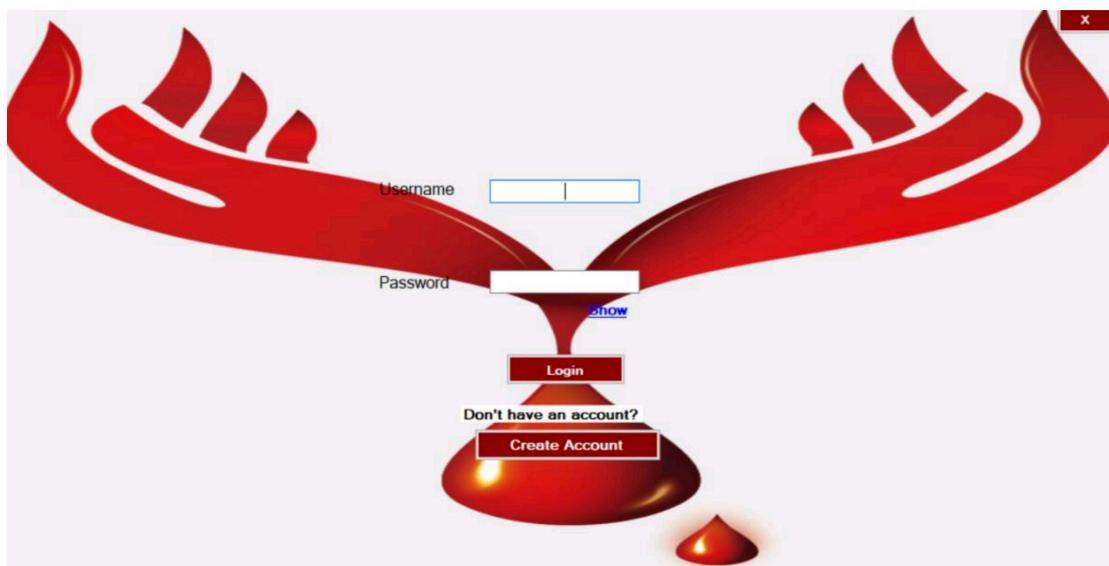
### 1. Signup for a new User:-

Open Blood Bank Management System application & click on the Create new account button. Then this login window will close and the Register Window will open. Then you have to submit valid info to successfully register. When a user registers for the first time, after registration the login page will open then Login in the System.



### 2. Login for a new User:-

To Login into System, submit the valid details to access your account.



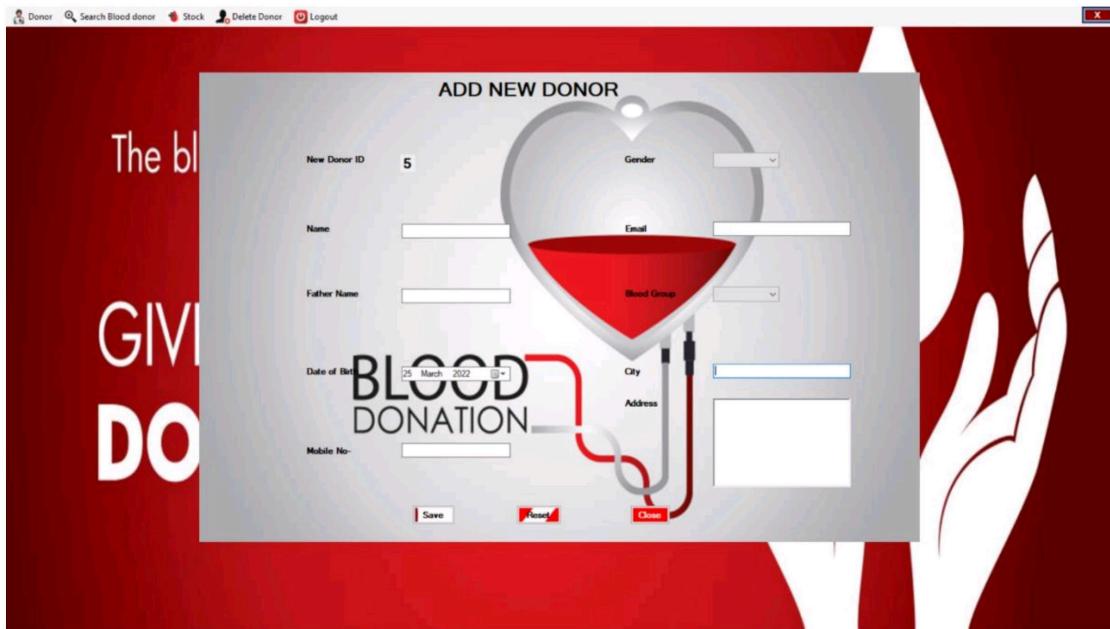
### **3. Home Page:-**

After logging in the user will be redirected to home page. This page helps the user to add Detail, Update and view details. It shows the different Packages and hotels for bookings and many more.



### **Add Donor Page:-**

In these Page, the user can add new Donor details:



## **CHAPTER 7**

### **Conclusions**

#### **7.1 Conclusion:-**

In light of results, this concentrate on that web-based blood donation centre administration framework is obviously superior to the manual framework. The discoveries showed that respondents like to utilize online blood donation centre administration framework instead of the manual framework since it offers many benefits and advantages that prompts its adequacy, and efficiency. Because of the expanded confidence on the clients on the framework it tends to be inferred that the blood donation centre administration framework improve blood bonding security since it gives better approaches to taking care of the different cycles in blood donation centre.

This project will also help me advance in my professional career. I have learnt how to implement things practically and to understand the beauty of the code. It has helped me in developing my communication skills which has played a big role in my academic career. I have become more productive. I have also learnt the various factors that are responsible for an active completion of the project.

I have developed more skills. I also have learnt to deal with the project with a lot of patience. It helped to organise the things in various patterns and ways. The project has helped me to develop my focusing and gasping skills which will help me towards gaining more knowledge about new things in future.

#### **7.2 Limitation of the Project:-**

There are only gathering information about donor.

Only one person can use and cannot connect to other person via this application.

This is not an online portal that anyone can use it.

#### **7.3 Future Scope:-**

This System isn't just for business reason however also for social administrations so the degrees can be this framework can be utilized for keeping up with all the interaction and exercises of blood bank management system ,try not to aggregate blood from people who could likewise be unsatisfactory because of the danger of wellbeing factor that also hurt the patients through the blood subtleties of the contributor this framework can give a proficient giver and blood stock administration capacity to the blood donation centre

This framework can be utilized to oversee and control the supply of the blood

In the advancement of this framework there will be the space for future adjustment.

## **REFERENCES:-**

### **Websites:-**

- 1) en.wikipedia.org
- 2) <https://dotnet.microsoft.com>
- 3) <http://www.tutorialspoint.com>
- 4) For font and style used: <https://www.icons8.com>
- 5) [www.github.com](https://github.com)
- 6) [www.w3schools.com](https://www.w3schools.com)