ABSTRACT

This project has been developed for AMA Blood Bank situated in Yeshwantpur, Bangalore. Since all the transactions were recorded manually in the blood bank the process used to be very tedious. There is no easy access to find the availability of blood in case of emergency .It used to be difficult to get the donor details if he misplaced or forget to bring the membership card given to him by the blood bank.

As the stored data increases manual handling becomes very tedious and tiring job also it consumes a lot of time. As the data is stored in the registers and files, it makes it hard to handle all papers, thus wastage of space. The files in which data are stored may be misplaced which can create problem. Separate files have to maintained for customer details, supplier details etc. Updating the record becomes difficult. Unauthorized person can go through the data and can change them. Report generation has to be done manually. Searching the record become a tedious job as their can many files and register.

1.INTRODUCTION

Introduction to the topic:

- The software is developed using the VB.Net as front end and SQL Server 2008 as back end and is compatible with any user microprocessor.
- It helps to maintain day to day transactions in a blood bank.
- It will also provide faster retrieval of data in the form of reports and search.
- The system will also provide reports on stocks of blood, blood collection details, blood issue details plasma, donor, hospital and employee details etc.

Different modules:

Donor:

- This module gives details about the donor. In addition to adding a donor, donor details can also be edited, deleted and updated.
- The module is also validated in order to take care of the accuracy of data.

  For ex. a) The employee won't be able to enter characters in numeric fields like contact number, age, weight etc. The donor details can also be searched easily and the results are shown with the help of data grid view.

  b) The age of the donor is checked to see whether the he/she is not under aged.

  c) The value for specific fields like hb, bp, etc., should be within a specified range.

  d) Each donor is identified by a unique ID in order to avoid duplication of data.

Recipient:

- This module gives details about the recipient including name, contact number, address, his/her medical condition or the disease he/she is suffering from. In addition to adding a recipient, recipient details can also be edited, deleted and updated.
- The module is also validated in order to take care of the accuracy of data.

For ex. a) The employee won't be able to enter characters in numeric fields like contact number, age, weight etc. The recipient details can also be searched easily and the results are shown with the help of data grid view.

Blood Test Report:

- This module helps to maintain the records of blood bags that have undergone tests. If the blood bag is acceptable and eligible to be donated then those bags are placed in a separate table indicating stock. The ones that are not acceptable will be stored in a separate table indicating their rejection and will be discarded later.
- The stock table also maintains information regarding the expiry date for all the blood bags.
- If a particular blood bag has expired then the same is intimated when the blood bag is about to be issued thus taking care of wrong blood bags being supplied.

Employee:

- This module gives details about the employee. In addition to adding a employee, employee details can also be edited, deleted and updated.
- The employee details can also be searched easily and the results are shown with the help of data grid view.

3. LITERATURE SURVEY

3.1. Existing system

- All the transactions were recorded manually in the blood bank the process used to be very tedious.
- There is no easy access to find the availability of blood in case of emergency.
- It used to be difficult to get the donor details if he misplaced or forget to bring the membership card given to him by the blood bank.

Drawbacks

- As the stored data increases manual handling becomes very tedious and tiring job also it consumes a lot of time.
- As the data is stored in the registers and files, it makes it hard to handle all papers, thus wastage of space. The files in which data are stored may be misplaced which can create problem.
- Separate files have to maintained for customer details, supplier details etc. Updating the record becomes difficult.
- Unauthorized person can go through the data and can change them. Report generation has to be done manually.
- Searching the record become a tedious job as their can many files and register.

Capabilities of the proposed system:

- No paper works. All data entry is computerized.
- Though the new proposed system the company can store their data in computer so wastage of space through lots of registers and files will not be there.
- Through the new proposed system the employee members will be able to view (search) any records and update the record will become simple.
- Data will be secured and protected.

- Confidential record can see only by the Administrator. These records and password protected.
- The reports will be auto generated.
- The calculation of data is automatically.

3.2 Literature Survey on the software used

Visual Basic.Net

Visual Studio.NET is an environment for developing Windows and Web applications. VisualBasic.NET is just one of the languages you can use to program your applications. Visual Studio .NET was designed to host any language, and many companies are working on languages that will be integrated in Visual Studio .NET. Some people will develop Windows applications in Visual Studio .NET with COBOL, or FORTRAN. So Visual Studio .NET is the environment that provides all the necessary tools for developing applications. The language is only one aspect of a Windows application. The visual interface of the application isn't tied to a specific language, and the same tools you'll use to develop your application's interface will also be used by all programmers, regardless of the language they'll use to code the application. The tools you'll use to access databases are also independent of the language. Visual Studio provides tools that allow you to connect to a database, inspect its objects, retrieve the information you're interested in, and even store it in objects that can be accessed from within any language. There are many visual tools in the IDE, like the Menu Designer. This tool allows you to visually design menus and to set their names and basic properties (such as checking, enabling, or disabling certain options). Designing a menu doesn't involve any code, and it's carried out with point-and-click operations. Of course, you will have to insert some code behind the commands of your menus, and (again) you can use any language to program them.

To simplify the process of application development, Visual Studio .NET provides an environment that's common to all languages, which is known as integrated development environment (IDE). The purpose of the IDE is to enable the developer to do as much as possible with visual tools, before writing code. The IDE provides tools for designing, executing, and debugging your applications. It's your second desktop, and you'll be spending most of your productive hours in this environment.

At Startup control, is where you define what you want Visual Studio .NET to do when it starts.

The choices are the following:

- ❖ Show Start Page Every time you start Visual Studio .NET, this page will appear.
- ❖ Load Last Loaded Solution Once you start working on a real project (a project that will take you from a few days to a few months to complete), select this option so that the project will be loaded automatically every time you start Visual Studio .NET.
- ❖ Show Open Project Dialog Box Every time you start Visual Studio .NET, the Open Project dialog box will appear, where you can select a project to open.
- ❖ Show New Project Dialog Box Every time you start Visual Studio .NET, the New Project dialog box will appear, where you can specify the name of a new project—a setting to avoid.
- ❖ Show Empty Environment This option instructs Visual Studio .NET to start a new empty solution, and you're responsible for adding new or existing projects to the solution and new or existing items to a project.

Project Types

All the project types supported by Visual Studio are displayed on the New Project dialog box, and they're the following:

Class library: A class library is a basic code-building component, which has no visible interface

and adds specific functionality to your project. Simply put, a class is a collection of functions that will be used in other projects beyond the current one. With classes, however, you don't have to distribute source code.

Windows control library: A Windows control (or simply control), such as a TextBox or Button, is a basic element of the user interface. If the controls that come with Visual Basic (the ones that appear in the Toolbox by default) don't provide the functionality you need, you can build your own custom controls. People design their own custom controls for very specific operations to simplify the development of large applications in a team environment. If you have a good idea for a custom control, you can market it—the pages of the computer trade magazines are full of ads for advanced custom controls that complement the existing ones.

Console application: A Console application is an application with a very limited user interface. This type of application displays its output on a Command Prompt window and receives input from the same window.

Windows service: A Windows service is a new name for the old NT services, and they're long running applications that don't have a visible interface. These services can be started automatically when the computer is turned on, paused, and restarted. An application that monitors and reacts to changes in the file system is a prime candidate for implementing as a Windows service. When users upload files to a specific folder, the Windows service might initiate some processing (copy the file, read its contents and update a database, and so on).

ASP.NET Web application: Web applications are among the most exciting new features of Visual Studio. A Web application is an app that resides on a Web server and services requests made through a browser. An online bookstore, for example, is a Web application. The application that runs on the Web server must accept requests made by a client (a remote computer with a browser) and return its responses to the requests in the form of HTML pages.

SQL SERVER 2005

SQL Server 2005 can be used to store information for personal use, for departmental use, for mid-size company use, or for enterprise use. SQL Server 2005 has editions to meet the needs in each of those scenarios:

Enterprise: Provides a relational database to meet the exacting needs of the largest enterprises and busiest online databases. The Enterprise Edition includes high-end business intelligence support and clustering.

Workgroup: Meets the needs of small- to medium-sized businesses that don't require the features of Standard Edition.

SQL Server 2005 is a client-server database. Typically, the SQL Server 2005 database engine is installed on a server machine to which you connect anything from a few machines to many hundreds or thousands of client machines. A client-server architecture can handle large amounts of data better than a desktop database such as Microsoft Access. The SQL Server instance provides security, availability, and reliability features that are absent from databases such as Access. Client-server architecture also can reduce network traffic.

Secured Database: If the data on which your business depends is stored in SQL Server, you need to keep the wrong people from accessing the data or, worse, changing or deleting the data. SQL Server 2005 implements Microsoft's recent emphasis on security. Unlike its predecessor (SQL Server 2000), SQL Server 2005 is much more secure by default.

Key Security features in SQL Server 2005:

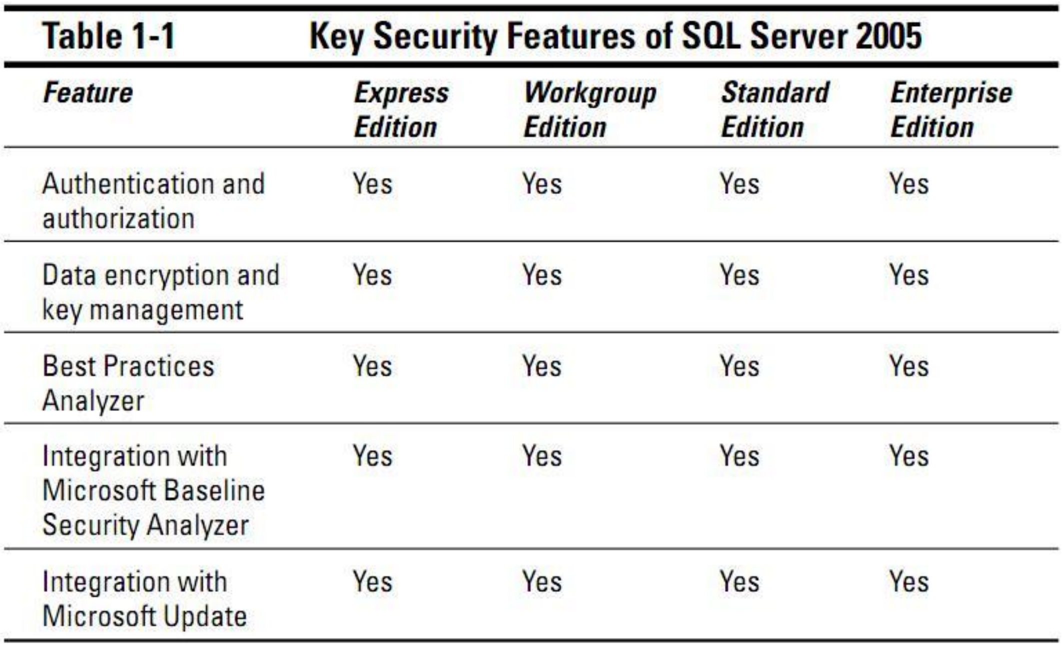


Table 1-1 Key Security Features of SQL Server 2005

Feature	*Express Edition*	*Workgroup Edition*	*Standard Edition*	*Enterprise Edition*
Authentication and authorization	Yes	Yes	Yes	Yes
Data encryption and key management	Yes	Yes	Yes	Yes
Best Practices Analyzer	Yes	Yes	Yes	Yes
Integration with Microsoft Baseline Security Analyzer	Yes	Yes	Yes	Yes
Integration with Microsoft Update	Yes	Yes	Yes	Yes

4. HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirement:

- Processor : Intel Pentium 4 or more
- RAM : 64 MB or more
- Hard Disk : 100 MB or more

4.2 Software Environment:

- Operating system : Microsoft Windows XP
- Processor : Intel Core Duo
- Database Tool : SQL Server 2005
- Development Tools : .Net Framework
- Technology : VB.Net

5. SRS (SOFTWARE REQUIREMENT SPECIFICATION)

5.1 Purpose of the project:

The main objective of this project is automated management of the Blood Bank. With the help of this project we can manage & store the Blood donor's as well as recipient's information . It also helps to keep a track of the blood stock and availability.

5.2 Scope of the project:

The main purpose for developing this project is managing the operations in a blood bank including blood donor's information & provides the good service for patients.

In most of the small scale and medium scale blood banks the reports are stored on the basis of paper work. There can be infinite number of errors in calculating the reports. Also the report cannot be given on time as the number of patients increases. The software is developed using the .Net framework and is compatible with any user microcomputer. Computerization requires less space because large Amount of data can be stored in small floppies or discs.

The small scale & medium scale blood bank management system lack in the following:

- Timely information cannot be produced in the case of large number of recipients.
- Work was not fast, accurate and secure.
- Many types of useful reports cannot be generated for management making decisions.
- Also the software should be flexible enough so that it can be integrated without any trouble.

5.3 Functional requirement

The problem statement of this project others the grate scope for developing the Blood Bank management system for the following system:

- Faster and efficiency in the processing of information.
- More timely information can be produced.

- Many types of useful reports can be generated for management for making decisions.
- In this project we can search donor's information depending on the donor's blood group wise.
- We can get information about the donor as well as recipient.

5.4 Non functional requirements

5.4.1 Efficiency of use:

The most important aspect of any GUI project is how efficient it is for the user to use the system. The interface for all the user must be very friendly and still the look and feel must be professional.

5.4.2 Learning environment and case of remembering:

The complete set must be very easy to understand and remember so that it would not take long for the user to get habitual of the project in his daily use of the project. This is extremely important because it will save time and reduce the paper work.

5.4.3 Performance Requirement:

Speed and Latency:

The motto of this project is how fast the information is store and save the Blood donors records and the complete information about the patients. With the help of this project we can manage the Blood bank record. The efficient use of database holds the key to successfully accomplish this mission.

Accuracy:

There is no chance for any kind of error or wrong information provided by the software. The accuracy completely depends on the information given by the administrator as the input to the software. So accuracy provided by the software cannot be compromised at any cost.

Security:

The security of the database and user access must be provided at all user level. "Who can access what" must be clearly defined with other security issue Multi user vulnerability and memory leakage.

Users Classes:

This project can be used by the small or medium scale Blood Bank for manages the donor's information. It can be more than hardly for the large scale organisations as well. The area of this project is as general that it can be used in various fields not necessary for a particular organization.

One of the most important advantages of computerized system is the marvelous speed. Instead of searching through record book for information which we want instantly Apart from the computerization involves. User operation that will surely increase the efficiency of the system.

6. SYSTEM DEFINITIONS

6.1 Existing System:

- All the transactions were recorded manually in the blood bank the process used to be very tedious.
- There is no easy access to find the availability of blood in case of emergency .
- It used to be difficult to get the donor details if he misplaced or forget to bring the membership card given to him by the blood bank.

6.2 Proposed system:

The software developed for Blood bank is Simple without complexities. The front-end used is developed in Microsoft Visual Basic. Net and the back-end is Microsoft SQL Server 2005 which offers the primary key concept & data security.

System will provide the following features.

- More user Friendly since Visual Basic. Net is the best GUI
- Lack of redundancy in well maintained data.
- Data Security
- Easy generation of reports

6.3 Advantages:

- Security is assured. Any unauthorized person cannot access the information since the system has separate login information for each employee. Therefore the system becomes more secure.
- System is using MS-SQL Server 2005 as backend thus data is saved permanently and not lost.
- Report generation is computerized which saves times.
- Modifying and updating record become simple.
- It takes less times and it is not tiring job.
- Record can be maintained easily.

- ❖ Searching of records becomes easy.
- ❖ No data redundancy.
- ❖ No delay in work.
- ❖ Data cannot be misused.
- ❖ No paper work involved.
- ❖ Data is stored in database in integrated format.
- ❖ Data is stored in ventral place such that anyone can use the data.
- ❖ Referential integrity of data is maintained.
- ❖ Provides quick access to data in database.

6.4 System Design

Design is the technical kernel of the software. During design, progressive refinements of data structure, program structure, and procedural detail are developed, reviewed and documented. Design results in representations of software that can be assessed for quality.

Modularity and the concept of abstraction enable the designer to simplify and reuse software components. Refinement provides a mechanism for representing successive layers of functional details. Program and data structure contribute to an overall view of software architecture, while procedure provides the details necessary for algorithm implementation. Information hiding and functional independence provide heuristics for achieving effective modularity.

Software design can be viewed from either a technical or project management perspective. Design notation, coupled with structural programming concepts, enables the designer to represent procedural detail in a manner that facilitates translation to cod, graphical tabular textual notation are available.

Data designs

Data design is the first of three design activities that are conducted during the development of the software. The impact of data structure on program structure and procedural complexity causes data design to have a profound influence on software quality.

User Interface Design

The user interface of a system is often the yardstick by which that system is judged. Software engineers must often take the responsibility for user interface design as well as the design of the software to implement that interface.

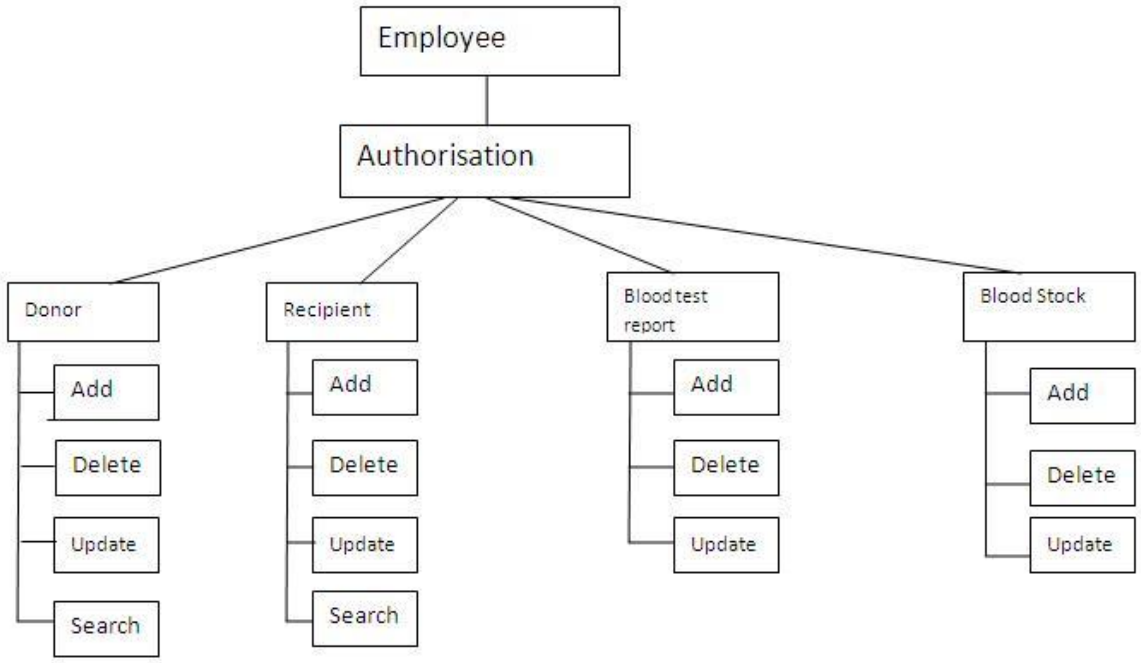
Procedural Design

Procedural design occurs after data and program structure have been established. Procedural design must specify procedural detail unambiguously, and a lack of ambiguity in a natural language is not natural. Using a natural language we can write a set of procedural steps in too many different ways. We frequently rely on context to get a point across.

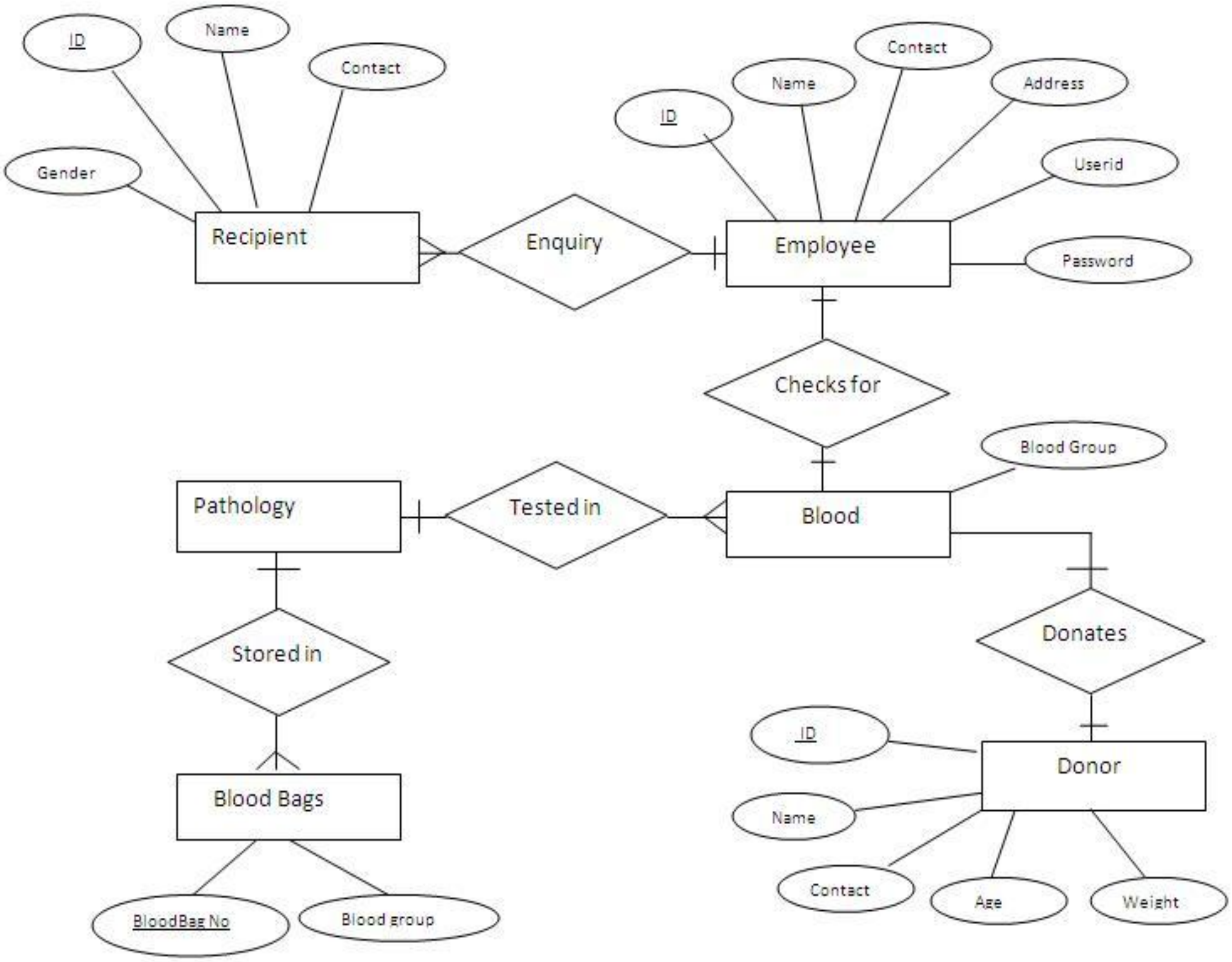
Architectural Design

The primary objective of architectural design is to develop a modular program structure and represent the control relationship between modules. In addition architectural design melds program structure defining interfaces that enable data to flow throughout the program.

Architectural design for blood bank management system



6.5 ER Diagram



7. DETAILED DESIGN

Detailed design of the system is the last design activity before implementation begins. The hardest design problems must be addressed by the detailed design or the design is not complete. The detailed design is still an abstraction as compared to source code, but should be detailed enough to ensure that translation to source is a precise mapping instead of a rough interpretation.

The detailed design should represent the system design in a variety of views where each view uses a different modeling technique. By using a variety of views, different parts of the system can be made clearer by different views. Some views are better at elaborating a systems states whereas other views are better at showing how data flows within the system.

7.1 Algorithms

Donor :

1. Start
2. Login form is displayed
3. Employee enters his login details
4. MDI form is displayed
5. Select the View menu
6. If the donor has not been registered select the option “donor” else go to step 8.1
7. Enter the details of the donor and save
8. Go to MDI form

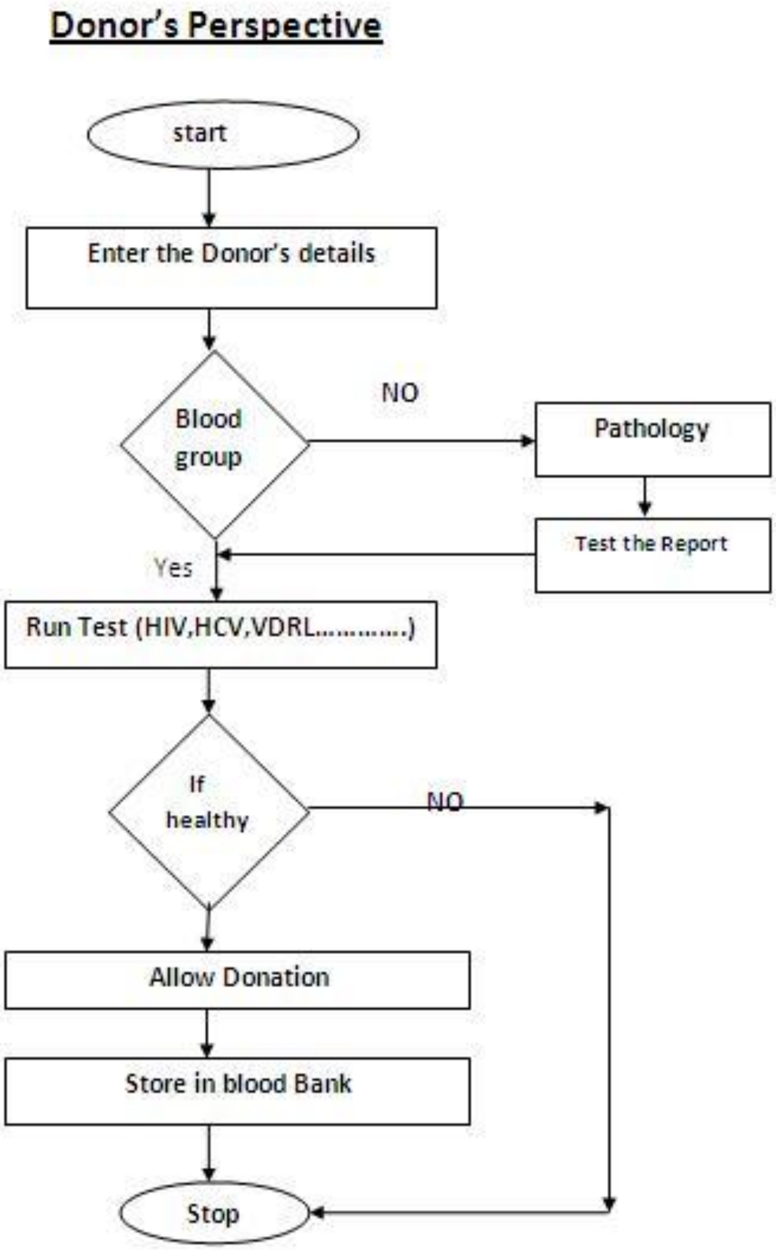
   8.1 select the donation option under “view” menu
9. Donation form is displayed
10. Enter the donation details like blood bag no. medical officer’s name etc.,
11. Save the record
12. Stop

Recipient:

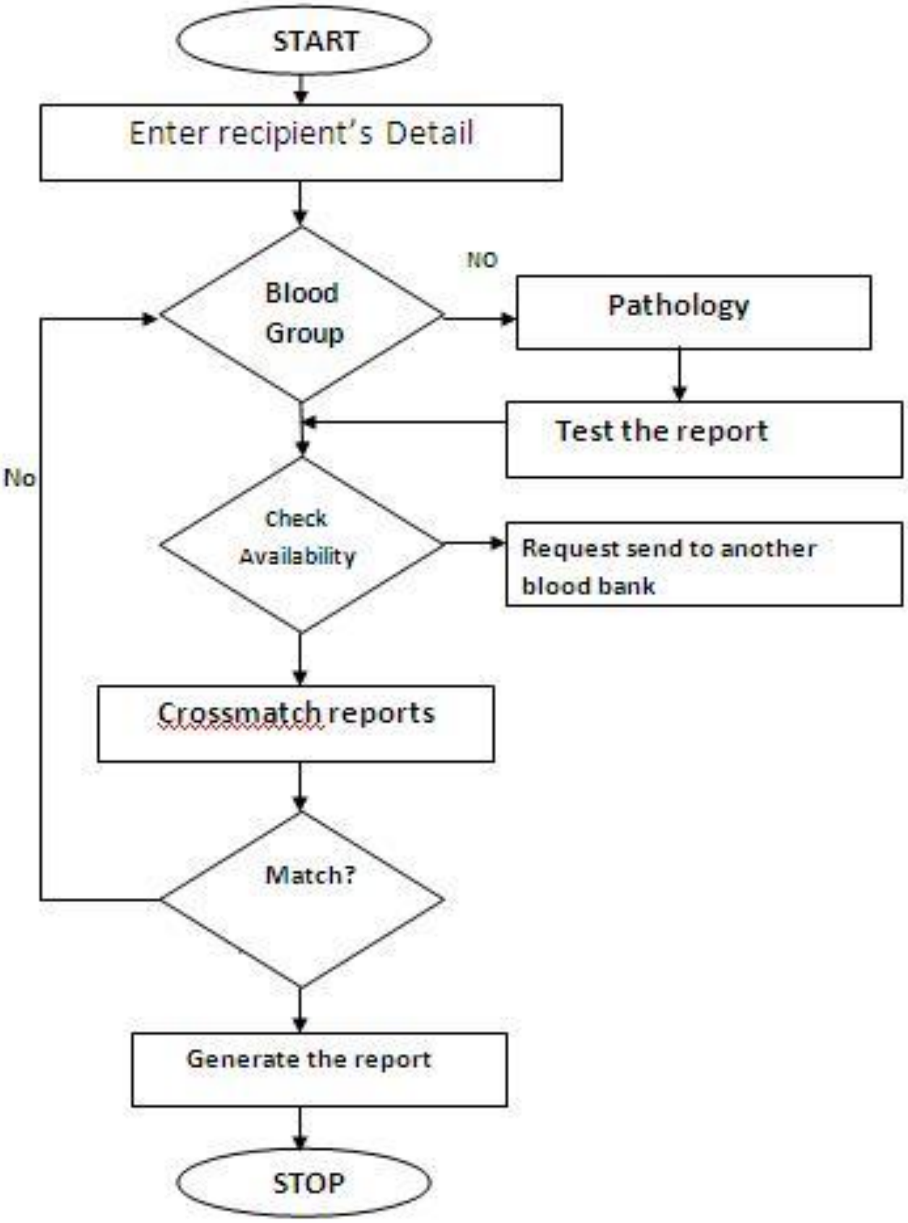
1. Start
2. Login form is displayed
3. Employee enters his login details
4. MDI form is displayed
5. Select the View menu
6. If the recipient has not been registered select the option “recipient” else go to step 8.1
7. Enter the details of the recipient and save
8. Go to MDI form

   8.1 select the blood issue option under “view” menu
9. Blood issue form is displayed
10. Enter the blood issue details like blood bag no. , recipient id etc.,
11. Save the record
12. Stop

7.2 Flowchart

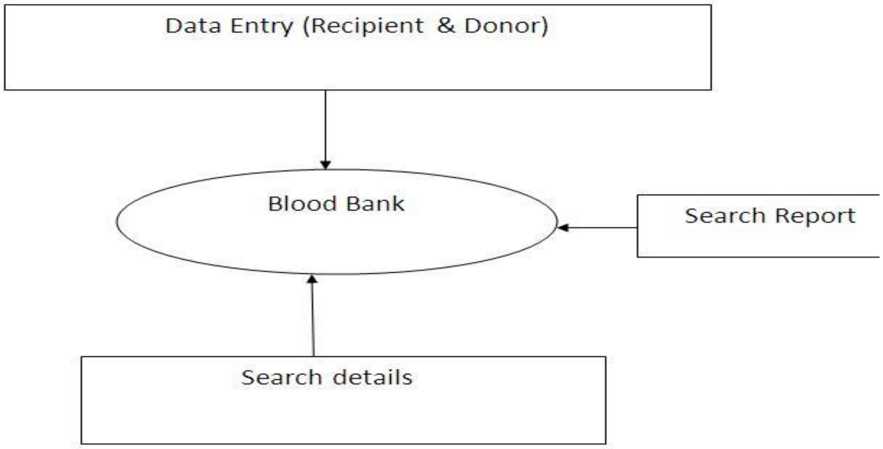


Recipient's Perspective

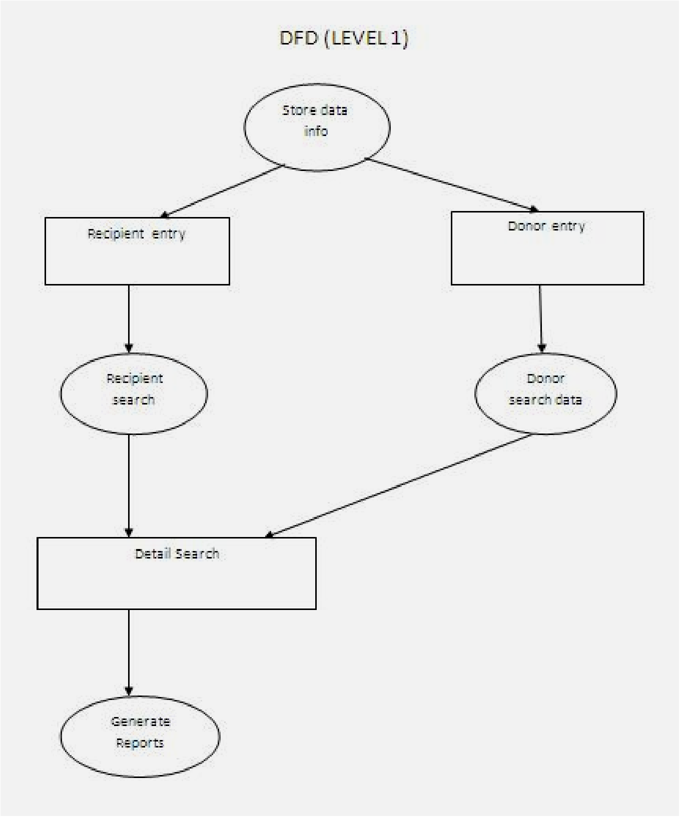


7.3 Data Flow Diagrams

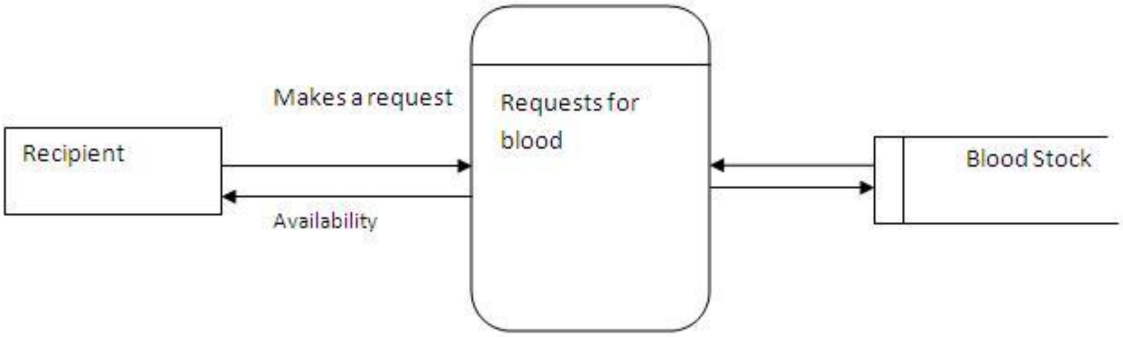
DFD(Context Free)

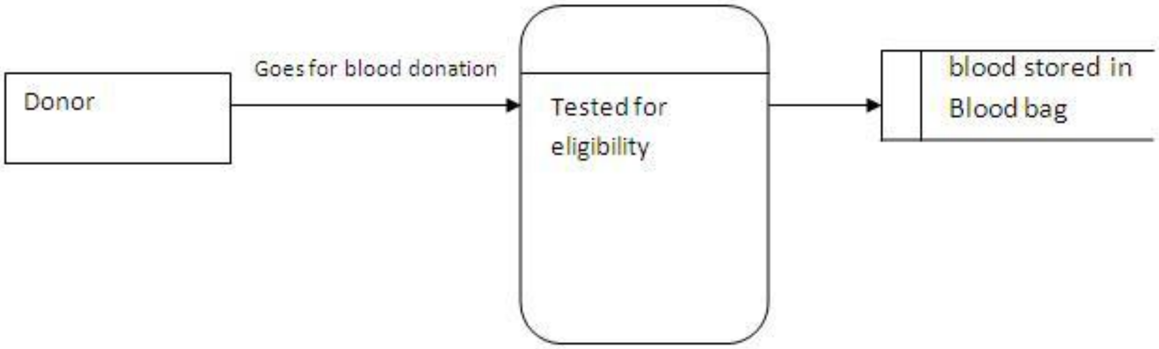


DFD(LEVEL 1)



DFD Fragments





7.4 Design of the proposed system

Functionalities of modules:

Donor:

Module	**Control**	**Name**	**Function**
DONOR	Textbox	Textbox1	Accept id
		txtdname	Accepts name
		txtadd	Accepts address
		txtage	Accepts age
		txtwt	Accepts weight
		txtcontact	Accepts contact
	Button	btnadd	To add
		cmdsave	To save
		btncancel	To cancel
		btnedit	To edit
		btnupdate	To update
		btndel	To delete
		btnnext,btnprev Btnfirst,btnlast	To navigate
	Datepicker	dtpic	To set the date
	Combobox	combobg	To select the blood group
	Radiobutton	Radiomale radiofemale	To select gender

Recipient:

Module	**Control**	**Name**	**Function**
RECIPIENT	Textbox	txtid	Accept id
		txtname	Accepts name
		txtaddress	Accepts address
		txtage	Accepts age
		txtcontact	Accepts contact
		txtit	Accepts medical condition
		txthos	Accepts hospital name
	Button	btnadd	To add
		cmdsave	To save
		btncancel	To cancel
		btnedit	To edit
		btnupdate	To update
		btndel	To delete
		btnnext,btnprev Btnfirst,btnlast	To navigate
	Datepicker	dtpic	To set the date
	Combobox	combobg	To select the blood group
	Radiobutton	Radiomale radiofemale	To select gender

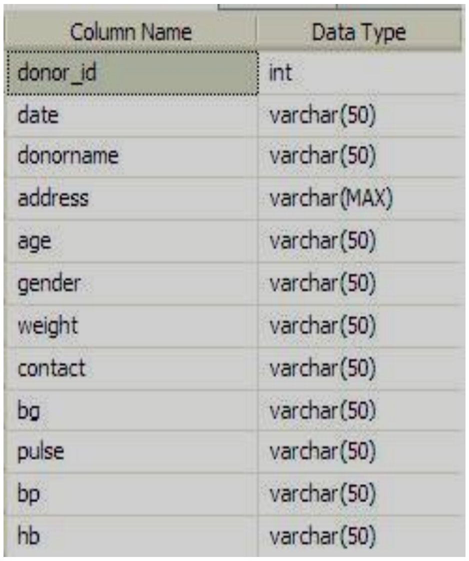
Blood Issue:

Module	Control	Name	Function
BLOOD ISSUE	Textbox	txtid	Accept id
		txtcost	To enter the cost
	Button	btnadd	To add
		cmdsave	To save
		btncancel	To cancel
		btnedit	To edit
		btnupdate	To update
		btndel	To delete
		btnnext,btnprev btnfirst,btnlast	To navigate
	Datepicker	dtpic	To set the date
	Combobox	cbobg	To select the blood group
		cbobbno	To select the blood bag no.
		cborecid	To select the recipient id
	Radiobutton	Radiorep radiobuy	To select the mode

7.5 Table structure

7.5.1 Donor Table

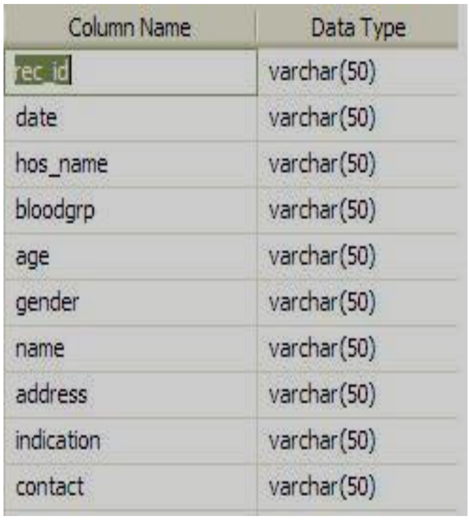
Table name	tbldonor
Description	This form helps to maintain the details about the donor
Primary key	donor_id



Column Name	Data Type
donor_id	int
date	varchar(50)
donorname	varchar(50)
address	varchar(MAX)
age	varchar(50)
gender	varchar(50)
weight	varchar(50)
contact	varchar(50)
bg	varchar(50)
pulse	varchar(50)
bp	varchar(50)
hb	varchar(50)

7.5.2 Recipient Table

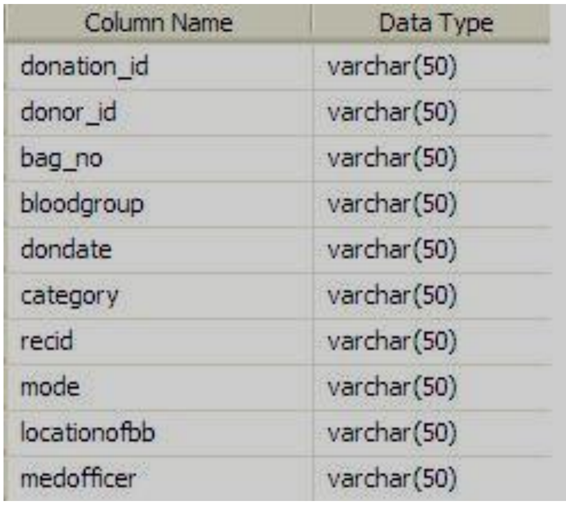
Table name	tblrecipient
Description	This form helps to maintain the details about the recipient
Primary key	rec_id



Column Name	Data Type
rec_id	varchar(50)
date	varchar(50)
hos_name	varchar(50)
bloodgrp	varchar(50)
age	varchar(50)
gender	varchar(50)
name	varchar(50)
address	varchar(50)
indication	varchar(50)
contact	varchar(50)

7.5.3 Donation Table

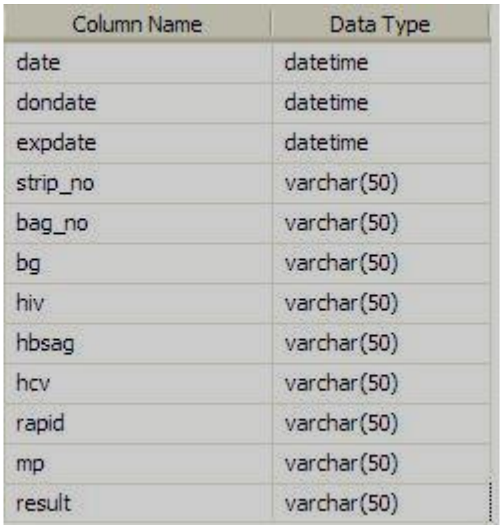
Table name	tbldonation
Description	This form helps to maintain the details about the blood donation
Primary key	donation_id



Column Name	Data Type
donation_id	varchar(50)
donor_id	varchar(50)
bag_no	varchar(50)
bloodgroup	varchar(50)
dondate	varchar(50)
category	varchar(50)
recid	varchar(50)
mode	varchar(50)
locationofbb	varchar(50)
medofficer	varchar(50)

7.5.4 TestReport table

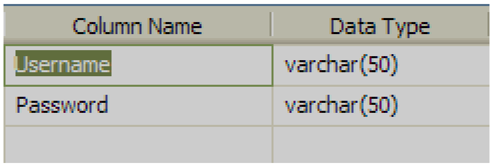
Table name	tbltestreport
Description	This form helps to maintain the details about the blood test reports
Primary key	Strip_no



Column Name	Data Type
date	datetime
dondate	datetime
expdate	datetime
strip_no	varchar(50)
bag_no	varchar(50)
bg	varchar(50)
hiv	varchar(50)
hbsag	varchar(50)
hcv	varchar(50)
rapid	varchar(50)
mp	varchar(50)
result	varchar(50)

7.5.5 Login Table

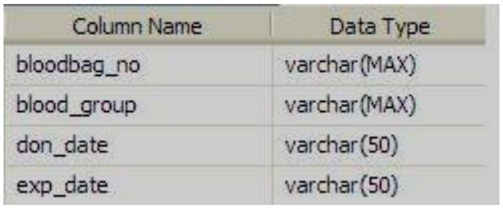
Table name	tbllogin
Description	This form helps to maintain the details about the donor
Primary key	donor_id



Column Name	Data Type
Username	varchar(50)
Password	varchar(50)

7.5.6 Accepted Table

Table name	tblaccepted
Description	This form helps to maintain the details about the blood bags that have been tested and accepted
Primary key	Bloodbag_no



Column Name	Data Type
bloodbag_no	varchar(MAX)
blood_group	varchar(MAX)
don_date	varchar(50)
exp_date	varchar(50)

8. IMPLEMENTATION

8.1 Software Installation

In order to run the project software's like Visual Studio 2008 and SQL Server 2005 must be installed on the computer.

The Tools used for implementation are

- Windows XP or above Operating System
- Microsoft SQL Server 2005 as the back-end developer.
- Visual Studio 2008 as the front-end interface.
- Report generation using Visual Studio Crystal Report Designer**.**

For establishing a connection with the database, we need to establish a connection with the DBMS which involves two steps:

- Loading the SQL Server connector.
- Establishing the connection.

Once the Connection is done all the necessary tables are created and the value for login table is inserted. Now login screen is opened and values are given according to the respective fields.

9. TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

Test case design

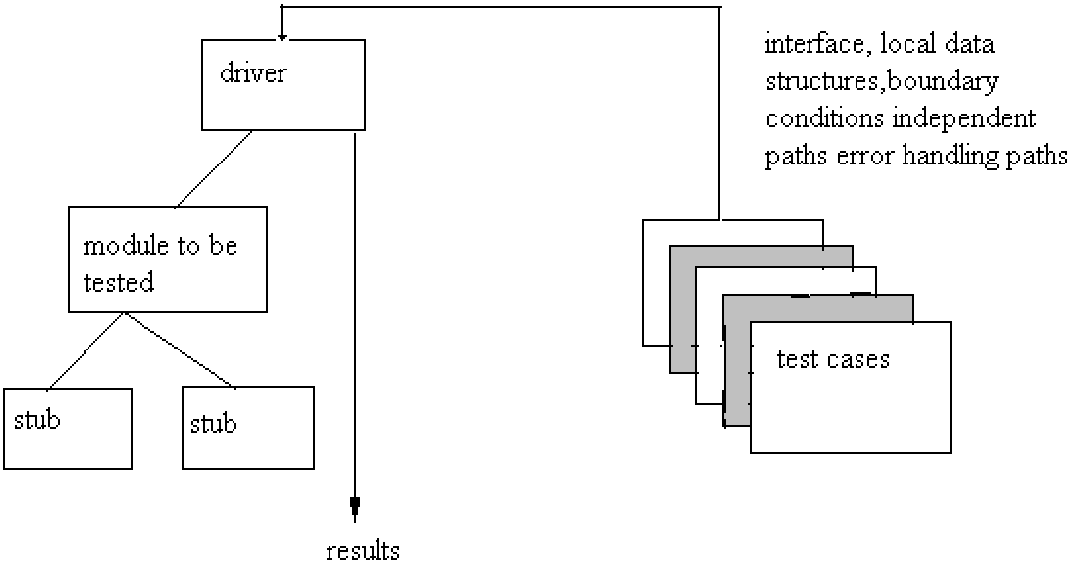
The Software was tested using Black Box Testing. The Black Box Tests were performed to check whether the software functions are operational, that input is properly accepted and the output is correctly produced. It was also used to check that the integrity of the external information is maintained.

Test Case For Blood Bank Management System

- If the user initially executes the software , the system should respond by requesting the password
- If the password is incorrect, the system must display an error message and asks the user to try again.
- After the login is successful, the system should show the main menu i.e the MDI Form.
- If a person comes for donating blood, he is first asked to fill the registration form and then initial checkup is done to see whether he is eligible for donating blood. The details are then entered in the Donor form
- After every application form filled by the user the database must get updated
- During cancellation, passenger information should be deleted from all relations to avoid inconsistency.
- After every updation of the database the system must go back to the main menu.

Unit Testing

The Unit Testing focuses verification effort on the smallest unit of software design- the software component or the module. All the interface modules were checked to see whether information flows in and out as desired .The relations in the database are examined to ensure that data stored maintains its integrity during all steps in an algorithm's execution.


driver
module to be tested
stub
stub
results
interface, local data structures,boundary conditions independent paths error handling paths
test cases

Integration Testing

Integration Testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

For Integration Testing the Top-Down method was used . Beginning with the main control module(main menu) modules(user interface forms) are integrated by moving downward through the control hierarchy.

The Top-Down Integration Testing method allowed verification of decision points early in the test process. The decision making occurs at upper-level and hence made it possible for early recognition of major control problems .

Validation Testing

The Validation Testing is used to check if the software functions in a manner that is expected by the user. Since the project developed was not built for any commercial use it didn’t undergo severe Validation Testing. However a reasonable amount of beta testing was done by making some people use the software and their feedback were taken into account. Thus beta testing helped to uncover those problems which could only be found out by the end-users .

Donor Module

Module Tested	Test case	Expected Output	Error fixed
Donor Module Entry	1)Blank input	Error Message	Bug free
	2)Age field less than<18	Error Message	Bug Free
	3)Numeric input in name field	Error Message	Bug free
	4)Character input in numeric field	Error Message	Bug free

Recipient Module

Module Tested	Test case	Expected Output	Error fixed
Donor Module Entry	1)Blank input	Error Message	Bug free
	2) Numeric input in name field	Error Message	Bug Free
	4)Character input in numeric field	Error Message	Bug free

Test Cases:

Login Form:

Test case No.	Test case	Application Form	Expected Result	Status
1	Username not entered	Login	Pops up client side message requesting entry of username	Tested ok
2	Incorrect username entered	Login	Client side message requesting entry of correct username is displayed	
3	Password not entered	Login	Pops up client side message requesting entry of password	Tested ok
4	Incorrect password entered	Login	Client side message requesting entry of correct password is displayed	

Donor Form:

Test case No.	Test case	Application Form	Expected Result	Status
1	Donor name not entered	Donor	Pops up client side message requesting entry of donor's name	Tested ok
2	Donor's age not entered	Donor	Client side message requesting entry of age is displayed	Tested ok
3	Donor's weight not entered	Donor	Pops up client side message requesting entry of weight	Tested ok
4	Contact number not entered	Donor	Client side message requesting entry of contact number	Tested ok
5	Blood group not selected	Donor	Client side message requesting selection of blood group from the combo box	Tested ok
6	Pulse/min field is left empty	Donor	Client side message requesting entry	Tested ok

			of pulse field	

Blood issue Form:

Test case No.	Test case	Application Form	Expected Result	Status
1	Recipient id not selected	Blood Issue	Pops up client side message requesting selection of recipient id	Tested ok
2	Blood group not selected	Blood issue	Client side message requesting selection of blood group from the combo box	Tested ok
3	Cost not entered	Blood issue	Pops up client side message requesting entry of cost	Tested ok

10. CONCLUSION

Blood bank management system fulfills the need of such a desk based system that can efficiently automated the complete process of the day to day activities of a blood bank such as generating blood donation reports, cross blood group matching of donor & patient & many such activities. The interface used in the software are extremely user friendly. Apart from being very efficient automation software, it also serves the purpose of a website for the blood bank. With the help of this software, the employees of the blood bank can work much faster as it serves them a lot of time which is of utmost importance in these places.

FUTURE ENHANCEMENTS

Apart from the existing facilities that has been introduced in this software that makes the user to do away with the manual work, there are some additional features that can be added to it in the future. As of now, the details of the donor, recipient, blood stock, donation, blood issue are maintained in the system. There are other things like blood bags, surgical and other equipments kept in the blood banks. Blood bank also keeps a record of the temperature bath. Those things are also maintained in the record books. Separate forms can be introduced for the same in the future and thus this software can be enhanced further.

11.BIBLIOGRAPHY

The following were referred during the analysis and execution phase of the project,

Reference Books:

1. HM.Deitel**,VB.Net (How to program)**
2. Raghu Ramkrishna: **Database Management System**, 3rd Edition, Tata McGraw-hill.
3. Jain: **SQL FOR PROFESSIONALS**, 3rd Edition, JSDN
4. Rogger Pressman: **Software Engineering**, 8th Edition, Tata McGraw-hill.

Web references:

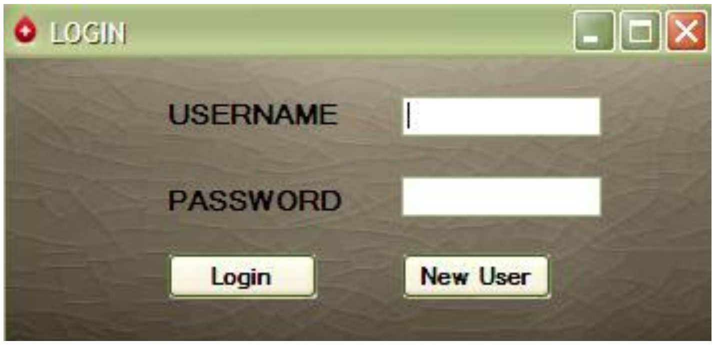
5. http://www.codeproject.com/Articles/19246/VB-net-to-mySQL-Database-Connection
6. http://www.codeproject.com/Questions/377052/validation-code-in-vb-net
7. http://www.codeproject.com/Articles/2162/AutoComplete-ComboBox-in-VB-Net

12. USER MANUAL

The Blood bank management system is developed to be user friendly so that the employees can use it with ease. However, while operating to understand and implement the system more effectively, the user manual can be referred by the candidate of the system.

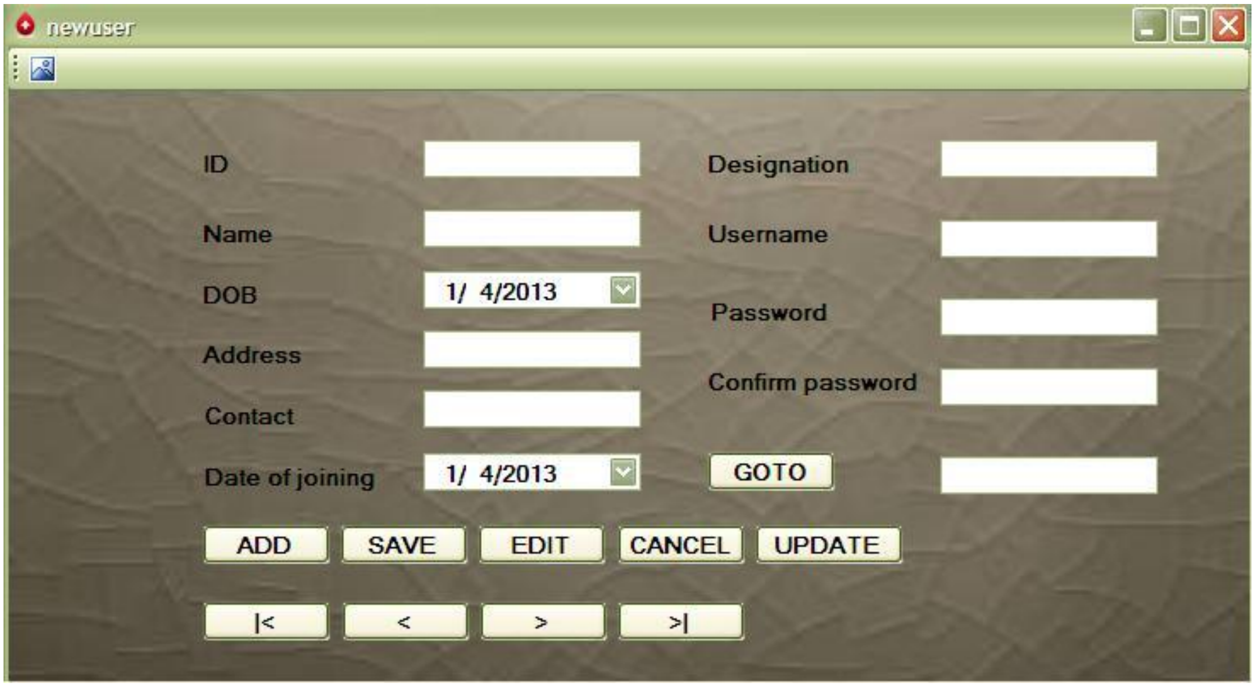
The software consists of Forms, reports developed using VB.net with backend as SQL Server database. The user manual projects all the major screens used in the application. The screens displayed in this section are divided as follows:-

Login Form:



The user starts by entering the login details like username and password. Click the button login after entering the details or else if the employee is a newly hired one then click on the button new user.

New User:

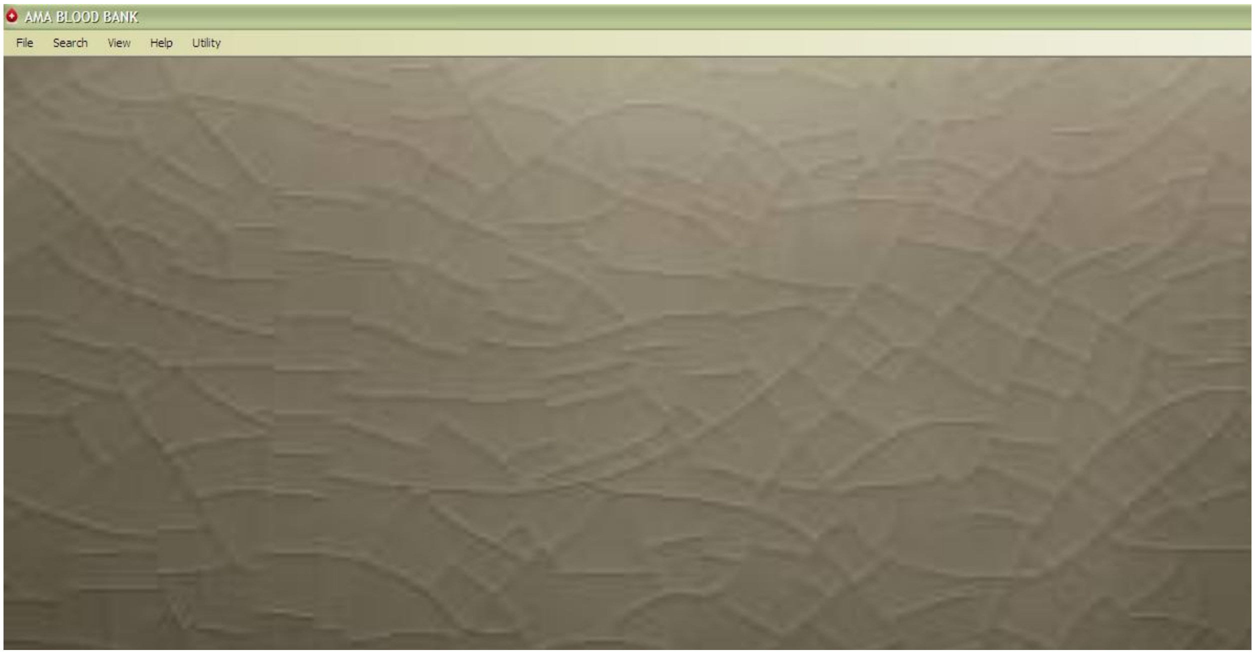


DOB 1/ 4/2013

Date of joining 1/ 4/2013

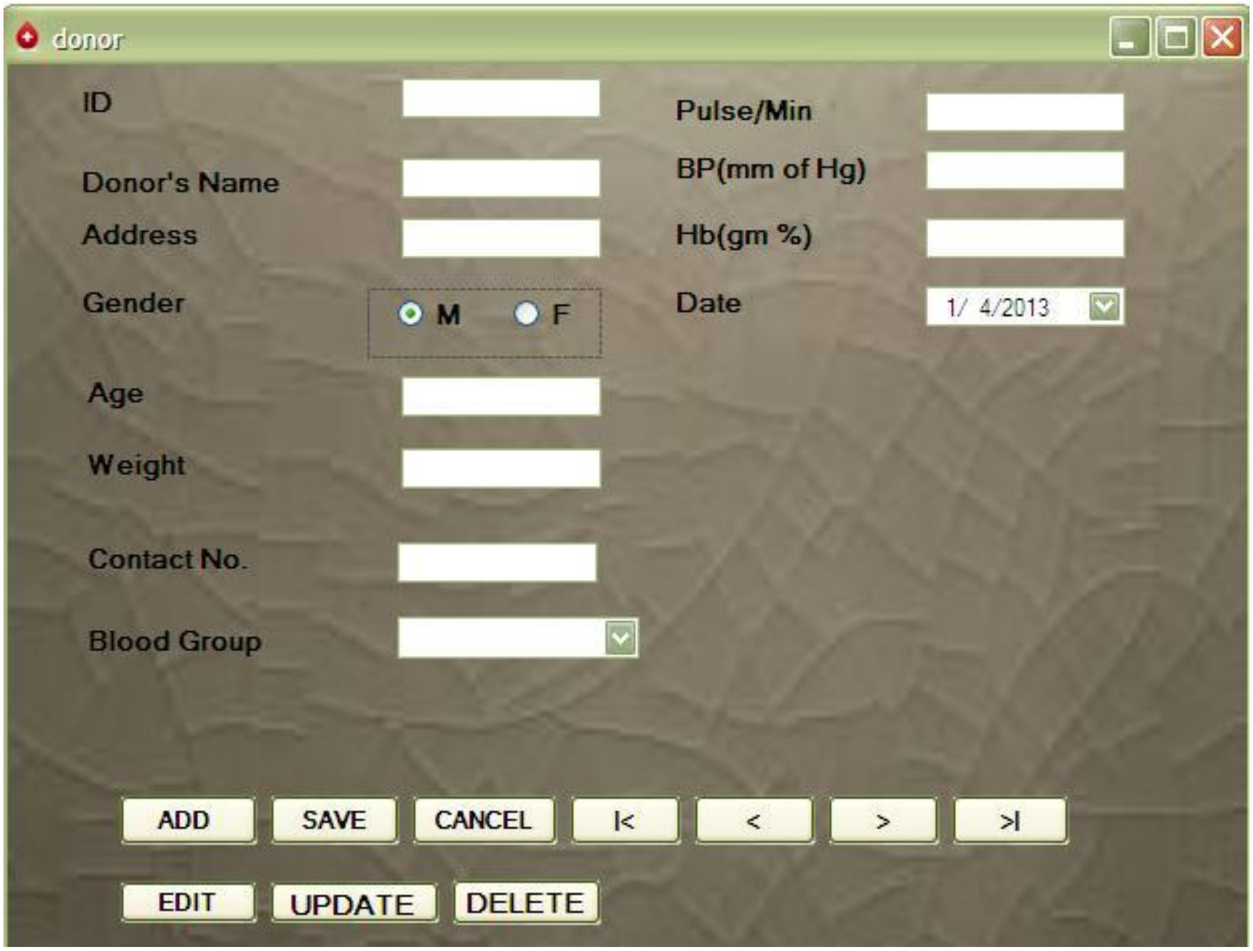
The details of the new employee are added to this form. There several buttons like new, save, edit, update, cancel which help us to add and edit the new user details. The navigation buttons helps to navigate through the records.

MDI Form:



The MDI form is the main form for all the activities. It consists of a menu bar through which we can navigate through all the forms like donor, recipient, donation, blood issue. The search menu helps us to search the donor, recipient, blood stock details.

Donor form:

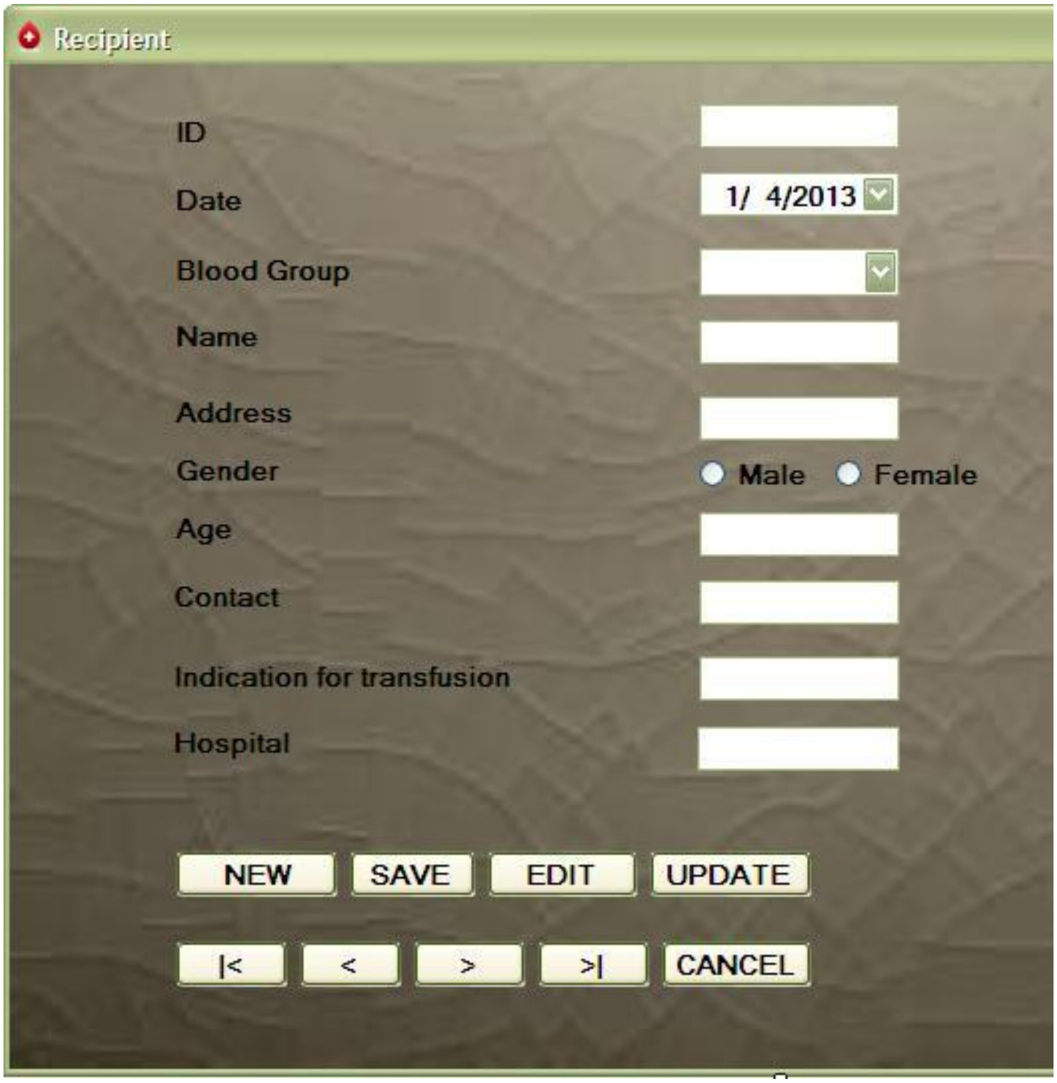


Gender ● M ○ F

Date 1/ 4/2013

This form makes it possible to enter the donor details. This form is available under the view option of the MDI form. There several buttons like new, save, edit, update, cancel which help us to add and edit the donor details. The navigation buttons helps to navigate through the records.

Recipient form:



Recipient

ID

Date 1/ 4/2013

Blood Group

Name

Address

Gender ○ Male ○ Female

Age

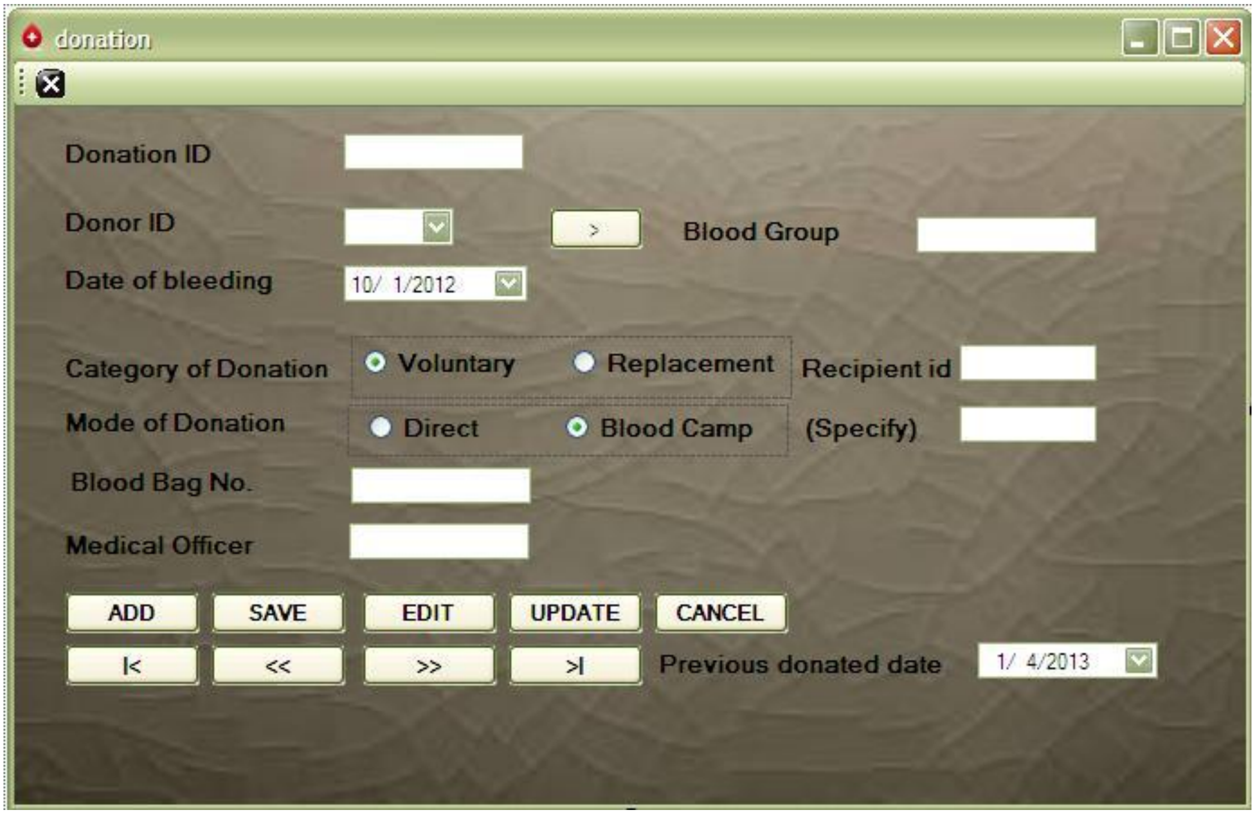
Contact

Indication for transfusion

Hospital

This form makes it possible to enter the recipient details. This form is available under the view option of the MDI form. There several buttons like new, save, edit, update, cancel which help us to edit the recipient details. The navigation buttons helps to navigate through the records.

Donation form:



Donation ID

Donor ID

Blood Group

Date of bleeding 10/ 1/2012

Category of Donation ● Voluntary ○ Replacement

Recipient id

Mode of Donation ○ Direct ● Blood Camp

(Specify)

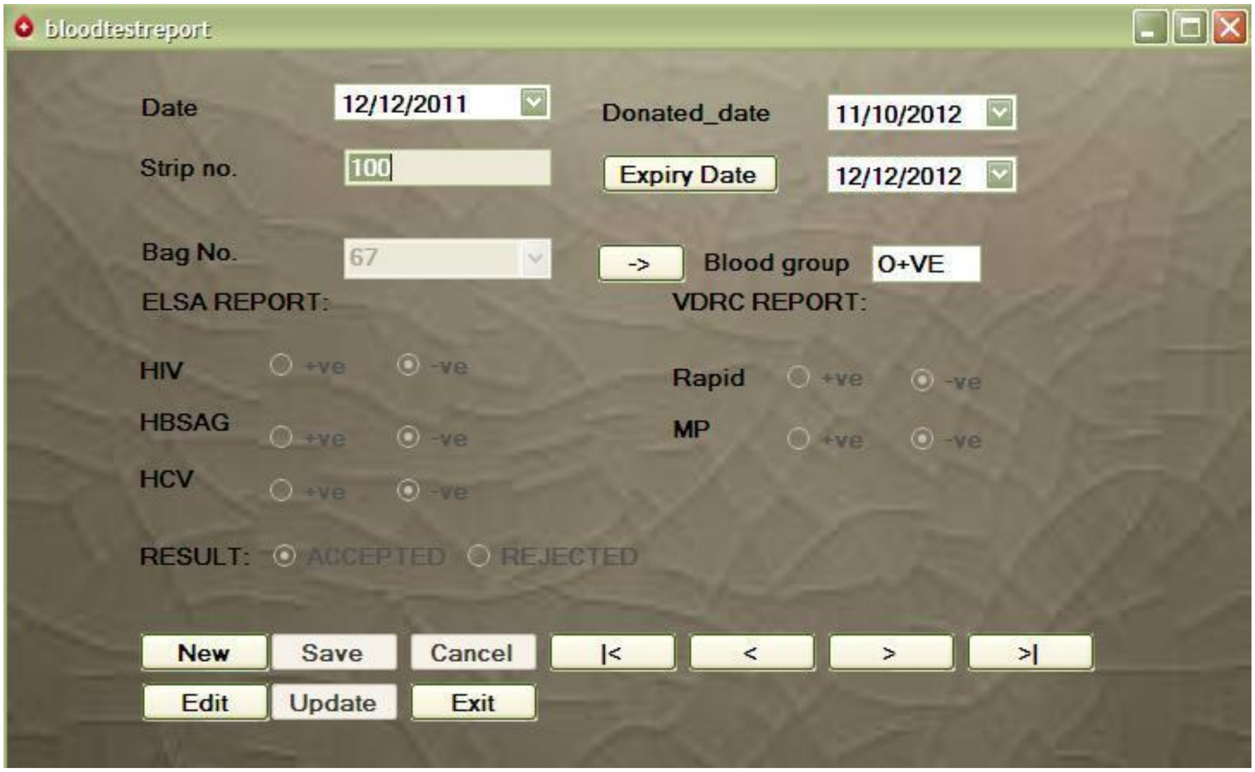
Blood Bag No.

Medical Officer

Previous donated date 1/ 4/2013

This form helps to enter the blood donation details. The donation details can be entered only after a donor has been registered or after saving his record in the donor form. This form can be accessed through the view menu in the MDI form.

Blood test report:



Date 12/12/2011

Donated_date 11/10/2012

Strip no. 100

Expiry Date 12/12/2012

Bag No. 67

Blood group O+VE

ELSA REPORT:

HIV ○ +ve ● -ve

HBSAG ○ +ve ● -ve

HCV ○ +ve ● -ve

VDRC REPORT:

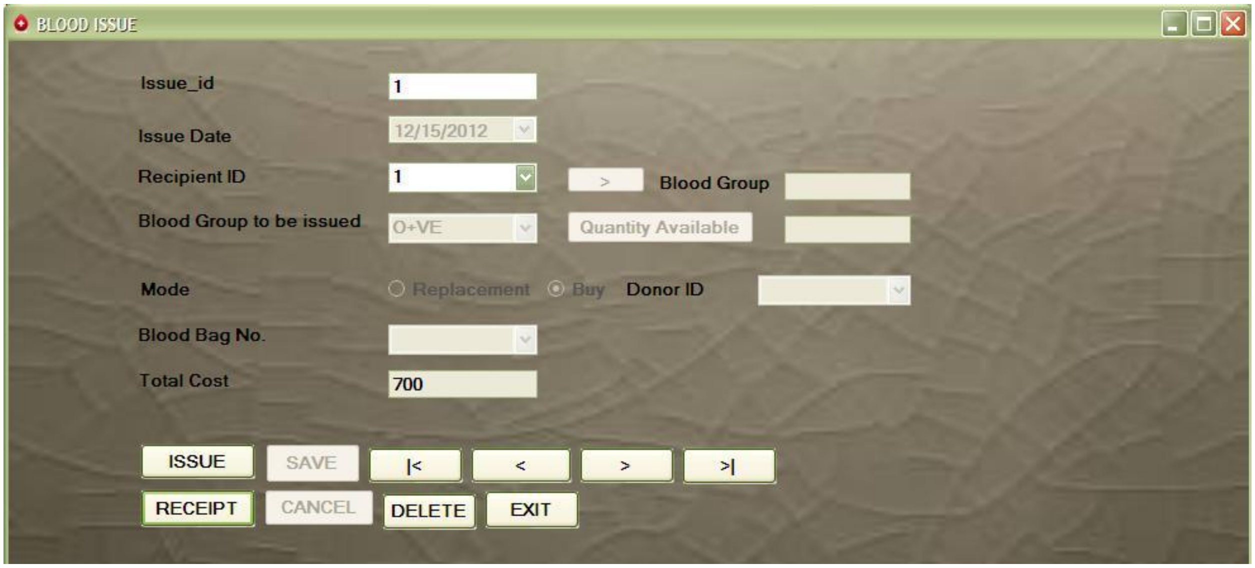
Rapid ○ +ve ● -ve

MP ○ +ve ● -ve

RESULT: ● ACCEPTED ○ REJECTED

This form helps to enter the blood test report details. After the donor has donated the blood the donation details are entered according to the blood bag no. The blood bag number field is populated from the database from the donation table. If the particular blood bag is accepted the record gets saved in the table named called Accepted and the rejected blood bags go to the table named Rejected.

Blood issue:



BLOOD ISSUE

Issue_id 1

Issue Date 12/15/2012

Recipient ID 1

Blood Group

Blood Group to be issued O+VE

Quantity Available

Mode ○ Replacement ● Buy

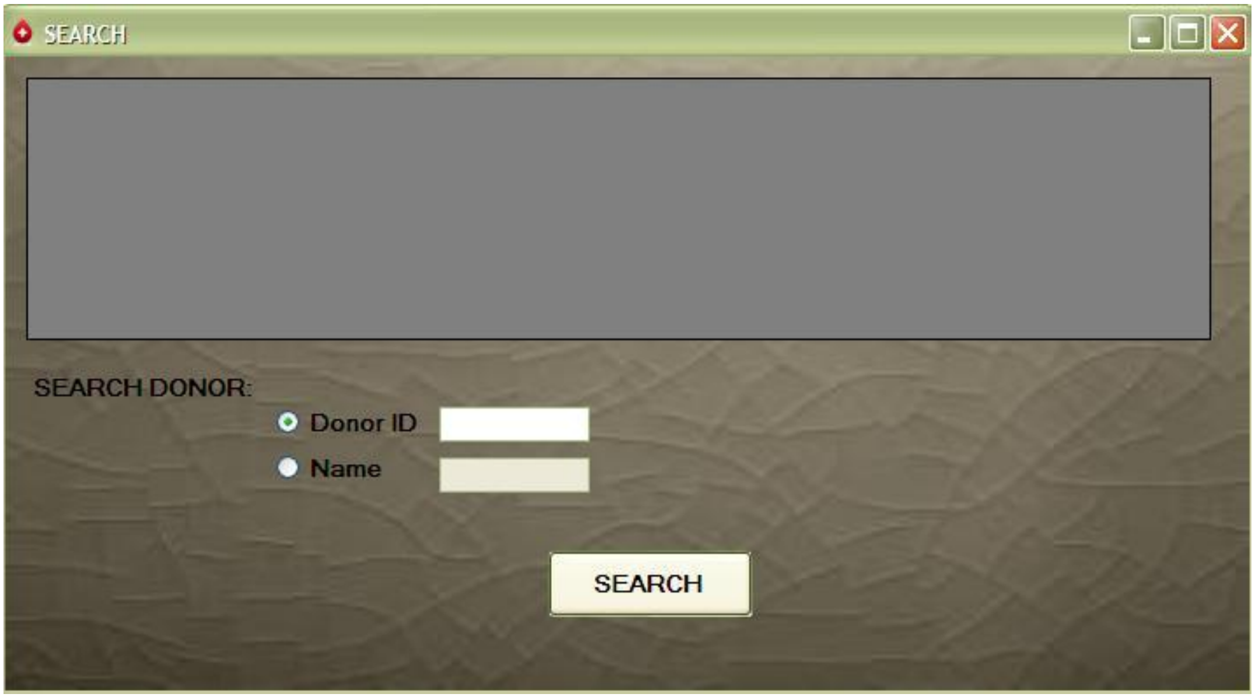
Donor ID

Blood Bag No.

Total Cost 700

This form helps to keep a record of the blood issues. When a patient or a recipient seeks for a particular blood group, he/she has to be registered first by entering the details in the recipient form. On selecting the blood group and on clicking the button “quantity available” The recipient ID comes from recipient table. This form can be accessed through the view menu in the MDI form.

Search forms:



This form is available under the search menu in the MDI form. Search forms are available for the donor, recipient. The blood stock can also be checked using the search form.

13. SOURCE CODE

Login:

```
Public Class Login

   Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click

       Dim dbcon As New SqlConnection("Data Source=ADMIN-PC\SQLEXPRESS;Initial
Catalog=bloodbank;Integrated Security=True")
       dbcon.open()
       Dim dbcmd As New SQLCommand
       dbcmd.connection = dbcon
       dbcmd.CommandText = "Select Count(*) From tbllogin where Username='" & TextBox1.Text & "' AND
Password='" & TextBox2.Text & "'"
       If dbcmd.ExecuteScalar = 1 Then
          MDIParent1.Show()
       Else
          MsgBox("Login failed")
          TextBox1.Clear()
          TextBox2.Clear()
       End If

       dbcmd.Dispose()
       dbcon.dispose()

   End Sub
```

Donor:

```
Imports System.Data.SqlClient

Public Class donor


   Dim str1 As String
   Dim str2 As String
   Dim str3 As String
   Dim dated As String
   Dim id As Integer
   Dim dr As SqlDataReader
   Dim ds As New DataSet
   Dim ds1 As New DataSet
   Dim da As New SqlDataAdapter
   Dim da1 As New SqlDataAdapter
   Dim i As Integer
   Dim ii As Integer
   Dim flagdonor As Boolean
   Dim flagage As Boolean

   Private Sub donor_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

       disControl()
       Call conecDB()
       Call initCMD()
       da = New SqlDataAdapter("select * from tbldonor", connDB)
       da.Fill(ds, "tbldonor")
       showrecords()

   End Sub
   Private Sub cleartxtbox()
       Me.txtdname.Text = " "
       Me.txtadd.Text = " "
       Me.txtadd.Text = " "
       Me.txtage.Text = " "
       Me.txtwt.Text = ""
       Me.txtcontact.Text = " "
       Me.cbobg.Text = " "
       Me.txtpulse.Text = " "
       Me.txtbp.Text = " "
       Me.txthb.Text = " "

   End Sub
```

```
Private Sub Label9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label9.Click

End Sub

Private Sub disControl()

    Me.dtpic.Enabled = False
    Me.txtdname.ReadOnly = True
    Me.txtadd.ReadOnly = True
    Me.radiomale.Enabled = False
    Me.radiofemale.Enabled = False

    Me.txtage.ReadOnly = True
    Me.txtwt.ReadOnly = True
    Me.txtcontact.ReadOnly = True
    Me.cbobg.Enabled = False
    Me.txtpulse.ReadOnly = True
    Me.txtbp.ReadOnly = True
    Me.txthb.ReadOnly = True
    'Me.txtpname.Enabled = False

    Me.btncancel.Enabled = False
    Me.cmdsave.Enabled = False
End Sub


Private Sub enaControl()
    Me.dtpic.Enabled = True
    Me.txtdname.ReadOnly = False
    Me.txtadd.ReadOnly = False
    Me.radiomale.Enabled = True
    Me.radiofemale.Enabled = True

    Me.txtage.ReadOnly = False
    Me.txtwt.ReadOnly = False
    Me.txtcontact.ReadOnly = False
    Me.cbobg.Enabled = True
```

```
        Me.txtpulse.ReadOnly = False
        Me.txtbp.ReadOnly = False
        Me.txthb.ReadOnly = False
        'Me.txtpname.Enabled = False

        Me.btncancel.Enabled = True
        Me.cmdsave.Enabled = True
    End Sub

    'Private Function invalidSaveEntry() As Boolean
    'Make sure that all fields have values
    'If Me.txtFname.Text.Trim = "" Or Me.txtMname.Text.Trim = "" Or Me.txtLname.Text.Trim = "" Or
Me.txtAge.Text.Trim = "" Then
    ' MsgBox("All fields are required!", MsgBoxStyle.Exclamation, "Insufficient Data")
    'Return True
    ' End If
    'Check if age is numeric
    'If IsNumeric(Me.txtAge.Text) = False Then
    'MsgBox("Age must be numeric!", MsgBoxStyle.Exclamation, "Invalid Age")
    'Return True
    'End If
    'End Function
    Public Sub checkvalidate()
        'For validation

        If txtdname.Text = " " Or cbobg.SelectedIndex = -1 Or txtage.Text = " " Or txtwt.Text = " " Or txtcontact.Text
= " " Or txthb.Text = " " Or txtbp.Text = " " Or txtpulse.Text = " " Or txtadd.Text = " " Then
        End If
        flagdonor = False
    End Sub


    Private Sub cmdsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
cmdsave.Click
        flagdonor = True

        If radiomale.Checked = True Then  'for male female radiobutton
```

```
            str1 = radiomale.Text
        Else
            str1 = radiofemale.Text
        End If


        dated = dtpic.Value
        Call conecDB()
        Call initCMD()
        'Call disControl()
        Call checkvalidate()

        If flagdonor = True Then

            SQL = "Insert into tbldonor(donor_id,date,donorname, address, age,gender, weight, contact, bg, pulse, bp,
hb)values('" & TextBox1.Text & "','" & dated & "','" & txtdname.Text & "','" & txtadd.Text & "','" & txtage.Text &
"','" & str1 & "','" & txtwt.Text & "','" & txtcontact.Text & "','" & cbobg.Text & "','" & txtpulse.Text & "','" &
txtbp.Text & "','" & txthb.Text & "')"
            Call execComDB(SQL)     'Execute the insert query

            ds.Clear()
            da = New SqlDataAdapter("select * from tbldonor", connDB)
            da.Fill(ds, "tbldonor")
            If (ds.Tables(0).Rows.Count > 0) Then
                i = 0
                showrecords()
                cmdsave.Enabled = False
                btncancel.Enabled = False
                btnedit.Enabled = True
            End If
        Else
            MsgBox("All fields required")
        End If

        closeDB()
    End Sub
    Private Sub btncancel_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btncancel.Click

        disControl()
        Call conecDB()
        Call initCMD()
        ds.Clear()
        da = New SqlDataAdapter("select * from tbldonor", connDB)
        da.Fill(ds, "tbldonor")
        i = 0
        showrecords()
        btnadd.Enabled = True
        btnedit.Enabled = True
    End Sub



    Private Sub btnadd_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnadd.Click

        Me.btncancel.Enabled = True
        Me.cmdsave.Enabled = True
        Me.btnedit.Enabled = False
        Me.btnupdate.Enabled = False
        dtpic.Value = Now()
        da = New SqlDataAdapter("select * from tbldonor", connDB)

        da.Fill(ds, "tbldonor")
        i = ds.Tables(0).Rows.Count - 1 'go to last record
        TextBox1.Text = ds.Tables(0).Rows(i)("donor_id").ToString + 1
        cleartxtbox()
        enaControl()

    End Sub
    Private Sub btnfirst_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnfirst.Click
        If (ds.Tables(0).Rows.Count > 0) Then

            i = 0
            showrecords()

        End If
    End Sub

    Private Sub btnprev_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnprev.Click
        ' i = ds.Tables(0).Rows.Count
        If (i.Equals(ds.Tables(0).Rows.Count - 1) Or (i > 0)) Then


            i = i - 1
            showrecords()
        Else
            MsgBox("You have reached the first record", MsgBoxStyle.Critical)
        End If
    End Sub
    Public Sub showrecords()

        TextBox1.Text = ds.Tables(0).Rows(i)("donor_id").ToString()
        dtpic.Value = ds.Tables(0).Rows(i)("date").ToString()
        txtdname.Text = ds.Tables(0).Rows(i)("donorname").ToString()
        txtadd.Text = ds.Tables(0).Rows(i)("address").ToString()
        txtage.Text = ds.Tables(0).Rows(i)("age").ToString()
        If (ds.Tables(0).Rows(i)("gender").ToString.Equals("M")) Then
            radiomale.Checked = True
        Else
            radiofemale.Checked = True

        End If
        txtwt.Text = ds.Tables(0).Rows(i)("weight").ToString()
        txtcontact.Text = ds.Tables(0).Rows(i)("contact").ToString()
        cbobg.Text = ds.Tables(0).Rows(i)("bg").ToString()
        txtpulse.Text = ds.Tables(0).Rows(i)("pulse").ToString()
        txtbp.Text = ds.Tables(0).Rows(i)("bp").ToString()
        txthb.Text = ds.Tables(0).Rows(i)("hb").ToString()

    End Sub
    Public Sub showrecords1()

        TextBox1.Text = ds.Tables(0).Rows(i)("donor_id").ToString()
        dtpic.Value = ds.Tables(0).Rows(i)("date").ToString()
        txtdname.Text = ds.Tables(0).Rows(i)("donorname").ToString()
        txtadd.Text = ds.Tables(0).Rows(i)("address").ToString()
        txtage.Text = ds.Tables(0).Rows(i)("age").ToString()
        If (ds.Tables(0).Rows(i)("gender").ToString.Equals("M")) Then
            radiomale.Checked = True
        Else
            radiofemale.Checked = True

        End If
        txtwt.Text = ds.Tables(0).Rows(i)("weight").ToString()
        txtcontact.Text = ds.Tables(0).Rows(i)("contact").ToString()
        cbobg.Text = ds.Tables(0).Rows(i)("bg").ToString()
        txtpulse.Text = ds1.Tables(0).Rows(i)("pulse").ToString()
        txtbp.Text = ds.Tables(0).Rows(i)("bp").ToString()
        txthb.Text = ds.Tables(0).Rows(i)("hb").ToString()



    End Sub
    Private Sub btnlast_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnlast.Click
        da = New SqlDataAdapter("select * from tbldonor", connDB)
        'SqlCommandBuilder = New SqlCommandBuilder(da)
        da.Fill(ds, "tbldonor")

        i = ds.Tables(0).Rows.Count - 1
        showrecords()
    End Sub

    Private Sub btnnext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnnext.Click

        If (i < ds.Tables(0).Rows.Count - 1) Then
            i = i + 1
            showrecords()
        Else
            MsgBox("You have reached the last record", MsgBoxStyle.Critical)

        End If
    End Sub



    Private Sub btnedit_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnedit.Click
        enaControl()
        btnadd.Enabled = False
        cmdsave.Enabled = False
        btnupdate.Enabled = True
    End Sub

    Private Sub btnupdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnupdate.Click
        flagdonor = True

        dated = dtpic.Value
        Call conecDB()
        Call initCMD()
        Call disControl()
        Call checkvalidate()
        If flagdonor = True Then
            If MsgBox("Are you sure you want to update the selected record?", CType(MsgBoxStyle.YesNo +
MsgBoxStyle.DefaultButton2 + MsgBoxStyle.Question, MsgBoxStyle), "Update") = MsgBoxResult.Yes Then
```

```
                'Start Save

                SQL = "update tbldonor set date='" & dated & "',donorname='" & txtdname.Text & "', address='" &
txtadd.Text & "', age='" & txtage.Text & "',gender='" & str1 & "', weight='" & txtwt.Text & "', contact='" &
txtcontact.Text & "', bg='" & cbobg.Text & "', pulse='" & txtpulse.Text & "', bp='" & txtbp.Text & "', hb='" &
txthb.Text & "' where donor_id='" & TextBox1.Text & "'"
                Call execComDB(SQL)     'Execute the insert query
            End If
            ds.Clear()
            da = New SqlDataAdapter("select * from tbldonor", connDB)
            da.Fill(ds, "tbldonor")
            i = 0
            showrecords()
            btnadd.Enabled = True
        Else
            MsgBox("All fields required")
        End If
        closeDB()
    End Sub


    Private Sub btndel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndel.Click

        Call conecDB()
        Call initCMD()
        Call disControl()
        If MsgBox("Are you sure you want to delete the selected record?", CType(MsgBoxStyle.YesNo +
MsgBoxStyle.DefaultButton2 + MsgBoxStyle.Question, MsgBoxStyle), "Update") = MsgBoxResult.Yes Then
            '    'Update query
            'Validation
            'If invalidSaveEntry() = True Then
            'Call enaControl()
            'Exit Sub
            'End If
            'Start Save
            SQL = "delete  from tbldonor where donor_id='" & TextBox1.Text & "'"
            Call execComDB(SQL)     'Execute the insert query
            MsgBox("Record deleted successfully")
        End If
        ds.Clear()
        da = New SqlDataAdapter("select * from tbldonor", connDB)
        da.Fill(ds, "tbldonor")
        i = 0
        showrecords()
        btnadd.Enabled = True
    End Sub


    'VALIDATION
    Dim obj1 As New global2  'character
    Dim obj2 As New global1  'integer
    Dim obj3 As New global1
    Dim obj4 As New global1
    Dim obj5 As New global1
    Dim obj6 As New global1
    Dim obj7 As New global1
    'character validation
    'name
    Private Sub txtdname_keypress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtdname.KeyPress
        obj1.txtCharacterValidate(e)
    End Sub

    Private Sub txtage_keypress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs)
Handles txtage.KeyPress
        obj2.txtIntegerValidate(e)
    End Sub

    Private Sub txtwt_keypress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs)
Handles txtwt.KeyPress
        obj3.txtIntegerValidate(e)
    End Sub
    Private Sub txtcontact_keypress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtcontact.KeyPress
        obj4.txtIntegerValidate(e)
    End Sub
    Private Sub txthb_keypress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs)
Handles txthb.KeyPress
        obj5.txtIntegerValidate(e)
    End Sub
    Private Sub txtbp_keypress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs)
Handles txtbp.KeyPress
        obj6.txtIntegerValidate(e)
    End Sub
    Private Sub txtpulse_keypress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtpulse.KeyPress
        obj7.txtIntegerValidate(e)
    End Sub

    Private Sub txtage_Validating(ByVal sender As Object, ByVal e As System.ComponentModel.CancelEventArgs)
Handles txtage.Validating
        If (txtage.Text < 18) Then
            MsgBox("Donor is underaged!!!!")
            txtage.Focus()
        End If

    End Sub

End Class
```

Blood Test Report:

```
Imports System.Data.SqlClient
Public Class bloodtestreport

    Dim str1 As String
    Dim str2 As String
    Dim str3 As String
    Dim str4 As String
    Dim str5 As String
    Dim str6 As String
    Dim str7 As String
    Dim dated1 As String
    Dim donateddate As String
    Dim expirydate As String
    Dim dr As SqlDataReader
    Dim ds As New DataSet
    Dim ds1 As New DataSet
    Dim da As New SqlDataAdapter
    Dim da1 As New SqlDataAdapter
    Dim i As Integer


     Private Sub bloodtestreport_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        disControl()
        Call conecDB()
        Call initCMD()
        da = New SqlDataAdapter("select * from tbltestreport", connDB)
        da.Fill(ds, "tbltestreport")
        showrecords()

    End Sub

    Private Sub disControl()
        txtstrip.ReadOnly = True
        cbobag.Enabled = False
        radhivp.Enabled = False
        radhivn.Enabled = False
        radhbsagp.Enabled = False
        radhbsagn.Enabled = False
        radmpp.Enabled = False
        radmpn.Enabled = False
        radrapidp.Enabled = False
        radrapidn.Enabled = False
        radacc.Enabled = False
        radrej.Enabled = False
        radhcvn.Enabled = False
        radhcvp.Enabled = False
        btnsave.Enabled = False
        btnupdate.Enabled = False
        btncancel.Enabled = False
    End Sub
    Private Sub enaControl()
        txtstrip.ReadOnly = False
        cbobag.Enabled = True
        radhivp.Enabled = True
        radhivn.Enabled = True
        radhbsagp.Enabled = True
        radhbsagn.Enabled = True
        radmpp.Enabled = True
        radmpn.Enabled = True
        radrapidp.Enabled = True
        radrapidn.Enabled = True
        radacc.Enabled = True
        radrej.Enabled = True
        radhcvp.Enabled = True
        radhcvn.Enabled = True
    End Sub
    Public Sub showrecords()
        dtpic1.Value = ds.Tables(0).Rows(i)("date").ToString()
        dtdonated.Value = ds.Tables(0).Rows(i)("dondate").ToString()
        dtexpiry.Value = ds.Tables(0).Rows(i)("expdate").ToString
        txtstrip.Text = ds.Tables(0).Rows(i)("strip_no").ToString()
        cbobag.Text = ds.Tables(0).Rows(i)("bag_no").ToString()
        txtbg.Text = ds.Tables(0).Rows(i)("bg").ToString()

        If (ds.Tables(0).Rows(i)("hiv").ToString.Equals("+ve")) Then
            radhivp.Checked = True
        Else
            radhivn.Checked = True
        End If
        If (ds.Tables(0).Rows(i)("hbsag").ToString.Equals("+ve")) Then
            radhbsagp.Checked = True
        Else
            radhbsagn.Checked = True

        End If

        If (ds.Tables(0).Rows(i)("hcv").ToString.Equals("+ve")) Then
            radhcvp.Checked = True
        Else
            radhcvn.Checked = True
        End If

        If (ds.Tables(0).Rows(i)("rapid").ToString.Equals("+ve")) Then
            radrapidp.Checked = True
        Else
            radrapidn.Checked = True
        End If

        If (ds.Tables(0).Rows(i)("mp").ToString.Equals("+ve")) Then
            radmpp.Checked = True
        Else
            radmpn.Checked = True

        End If

        If (ds.Tables(0).Rows(i)("result").ToString.Equals("ACCEPTED")) Then
            radacc.Checked = True
        Else
            radrej.Checked = True
        End If

    End Sub
```

```
    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnsave.Click

        If radhivp.Checked = True Then  'for hiv +ve radiobutton
            str1 = radhivp.Text
        Else
            str1 = radhivn.Text
        End If

        If radhcvp.Checked = True Then 'for hcv
            str2 = radhcvp.Text
        Else
            str2 = radhcvn.Text
        End If

        If radmpp.Checked = True Then 'for mp
            str3 = radmpp.Text
        Else
            str3 = radmpn.Text
        End If

        If radhbsagp.Checked = True Then 'for hbsag
            str4 = radhbsagp.Text
        Else
            str4 = radhbsagn.Text
        End If

        If radacc.Checked = True Then 'for accepted
            str5 = radacc.Text
        Else
            str5 = radrej.Text   'for rejected
        End If

        If radrapidp.Checked = True Then 'for rapid
            str6 = radrapidp.Text
        Else
```

```
            str6 = radrapidn.Text
        End If

        dated1 = dtpic1.Value
        donateddate = dtdonated.Value
        expirydate = dtexpiry.Value
        Call conecDB()
        Call initCMD()
        Call disControl()
        'Validation
        'If invalidSaveEntry() = True Then
        'Call enaControl()
        'Exit Sub
        'End If
        'Start Save

        SQL = "Insert into
tbltestreport(date,dondate,expdate,strip_no,bag_no,bg,hiv,hbsag,hcv,rapid,mp,result)values('" & dated1 & "','" &
donateddate & "','" & expirydate & "','" & txtstrip.Text & "','" & cbobag.Text & "','" & txtbg.Text & "','" & str1 &
"','" & str4 & "','" & str2 & "','" & str6 & "','" & str3 & "','" & str5 & "')"
        Call execComDB(SQL)     'Execute the insert query

        If radacc.Checked = True Then
            SQL = "Insert into tblaccepted(bloodbag_no,blood_group,don_date,exp_date)values('" & cbobag.Text & "','"
& txtbg.Text & "','" & donateddate & "','" & expirydate & "')"
            Call execComDB(SQL)
        Else
            SQL = "Insert into tblrejected(bloodbag_no,blood_group,don_date)values('" & cbobag.Text & "','" &
txtbg.Text & "','" & donateddate & "')"
            Call execComDB(SQL)

        End If




        Me.Close()
        MDIParent1.Show()
    End Sub

    Private Sub btncancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btncancel.Click
        disControl()
        Call conecDB()
        Call initCMD()
        ds.Clear()
        da = New SqlDataAdapter("select * from tbltestreport", connDB)
        da.Fill(ds, "tbltestreport")
        showrecords()
        btnnew.Enabled = True
        btnedit.Enabled = True
        btnsave.Enabled = False
    End Sub
    Public Sub clearfields()
        cbobag.Text = " "
        radhivp.Checked = False
        radhivn.Checked = False

        radhbsagn.Checked = False
        radhbsagp.Checked = False

        radhcvn.Checked = False
        radhcvp.Checked = False

        radmpp.Checked = False
        radmpn.Checked = False

        radacc.Checked = False
        radrej.Checked = False

        radrapidn.Checked = False
        radrapidp.Checked = False

    End Sub
    Private Sub btnnew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnnew.Click
        btncancel.Enabled = True
        btnsave.Enabled = True
        Call conecDB()
        Call initCMD()
        ds.Clear()
        da = New SqlDataAdapter("select bag_no from tbldonation", connDB)
        da.Fill(ds, "tbldonation")
        cbobag.DataSource = ds.Tables("tbldonation")
        cbobag.ValueMember = "bag_no"
        cbobag.DisplayMember = "bag_no"

        Me.btncancel.Enabled = True
        Me.btnsave.Enabled = True
        Me.btnedit.Enabled = False
        Me.btnupdate.Enabled = False
        dtpic1.Value = Now()
        da = New SqlDataAdapter("select * from tbltestreport", connDB)

        da.Fill(ds, "tbltestreport")
        i = ds.Tables(0).Rows.Count - 1 'go to last record
        txtstrip.Text = ds.Tables(0).Rows(i)("strip_no").ToString + 1
        clearfields()
        enaControl()
    End Sub

    Private Sub btnupdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnupdate.Click

        If radhivp.Checked = True Then  'for hiv +ve radiobutton
            str1 = radhivp.Text
        Else
            str1 = radhivn.Text
        End If

        If radhcvp.Checked = True Then 'for hcv
```

```
            str2 = radhcvp.Text
        Else
            str2 = radhcvn.Text
        End If

        If radmpp.Checked = True Then 'for mp
            str3 = radmpp.Text
        Else
            str3 = radmpn.Text
        End If

        If radhbsagp.Checked = True Then 'for hbsag
            str4 = radhbsagp.Text
        Else
            str4 = radhbsagn.Text
        End If

        If radacc.Checked = True Then 'for accepted
            str5 = radacc.Text
        Else
            str5 = radrej.Text   'for rejected
        End If

        If radrapidp.Checked = True Then 'for rapid
            str6 = radrapidp.Text
        Else
            str6 = radrapidn.Text
        End If

        dated1 = dtpic1.Value
        donateddate = dtdonated.Value
        expirydate = dtexpiry.Value
        Call conecDB()
        Call initCMD()
        Call disControl()
        If MsgBox("Are you sure you want to update the selected record?", CType(MsgBoxStyle.YesNo +
MsgBoxStyle.DefaultButton2 + MsgBoxStyle.Question, MsgBoxStyle), "Update") = MsgBoxResult.Yes Then
```

```
            SQL = "update tbltestreport set date='" & dated1 & "',dondate='" & donateddate & "',expdate='" & expirydate & "',bag_no='" & cbobag.Text & "', bg='" & txtbg.Text & "',hiv='" & str1 & "', hbsag='" & str4 & "', hcv='" & str2 & "', rapid='" & str6 & "', mp='" & str3 & "', result='" & str5 & "' where strip_no='" & txtstrip.Text & "'"
            Call execComDB(SQL)     'Execute the insert query
        End If
        ds.Clear()
        da = New SqlDataAdapter("select * from tbltestreport", connDB)
        da.Fill(ds, "tbltestreport")
        showrecords()
        btnnew.Enabled = True
    End Sub

    Private Sub btnfirst_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnfirst.Click
        If (ds.Tables(0).Rows.Count > 0) Then
            i = 0
            showrecords()
        End If
    End Sub

    Private Sub btnprev_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprev.Click
        If (i.Equals(ds.Tables(0).Rows.Count - 1) Or (i > 0)) Then
            i = i - 1
            showrecords()
        Else
            MsgBox("You have reached the first record", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub btnnext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnnext.Click
        If (i < ds.Tables(0).Rows.Count - 1) Then
            i = i + 1
            showrecords()
        Else
```

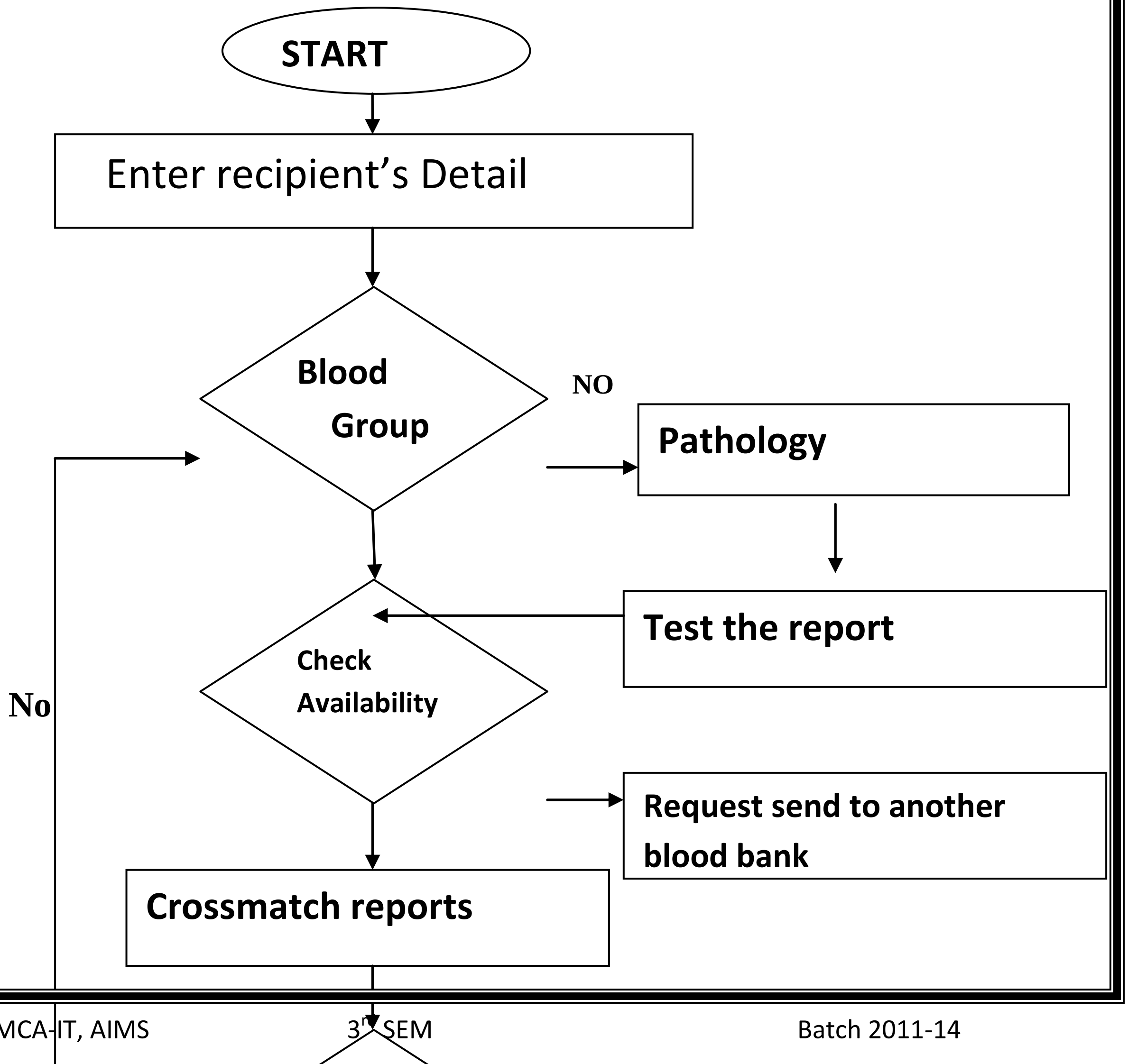
```
            MsgBox("You have reached the last record", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub btnlast_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnlast.Click
        da = New SqlDataAdapter("select * from tbldonor", connDB)
        da.Fill(ds, "tbldonor")
        i = ds.Tables(0).Rows.Count - 1
        showrecords()
    End Sub

    Private Sub btnedit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnedit.Click
        enaControl()
        txtstrip.Enabled = False
        btnnew.Enabled = False
        btnsave.Enabled = False
        btnupdate.Enabled = True
        btncancel.Enabled = True
    End Sub

    Private Sub btnexpiry_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexpiry.Click
        dtexpiry.Value = dtdonated.Value.AddDays(34)
    End Sub

    Private Sub btnbg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnbg.Click
        Call conecDB()
        Call initCMD()

        SQL = "Select bloodgroup,dondate from tbldonation where bag_no='" & cbobag.Text & "'"
        Call execComDB(SQL)
        dr = comDB.ExecuteReader
        dr.Read()
        txtbg.Text = dr.Item("bloodgroup")
        dtdonated.Value = dr.Item("dondate")
        dr.Close()
    End Sub
    Private Sub txtbg_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtbg.TextChanged

    End Sub

    Private Sub Label14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub
End Class
```

Blood Issue:

```
Imports System.Data.SqlClient
Public Class blood_issue


    Dim str1 As String
    Dim dated As String
    Dim id As Integer
    Dim dr As SqlDataReader
    Dim expirydate As DateTime
    Dim issuedate As DateTime
    Dim flag123 As Boolean

    Dim ds As New DataSet
    Dim ds1 As New DataSet
    Dim ds2 As New DataSet
    Dim ds3 As New DataSet

    Dim da As New SqlDataAdapter
    Dim da1 As New SqlDataAdapter
    Dim da2 As New SqlDataAdapter
    Dim da3 As New SqlDataAdapter

    Dim i As Integer
    Dim chk As Boolean
    Dim strbg As String
    Dim count As String


    Private Sub cleartxtbox()
        Me.cborecid.Text = " "
        Me.txtbg.Text = " "
        Me.cbobg.Text = " "
        Me.cbodonorid.Text = " "
        Me.cbobbno.Text = " "
        Me.txtcost.Text = " "
        Me.cbodonorid.Text = " "

    End Sub

    Private Sub disControl()
        Me.txtdatepic.Enabled = False
        Me.txtbg.ReadOnly = True
        Me.cbodonorid.Enabled = False
        Me.radrep.Enabled = False
        Me.radbuy.Enabled = False
        Me.cbobg.Enabled = False
        Me.cbobbno.Enabled = False
        Me.txtcost.ReadOnly = True
        Me.btncancel.Enabled = False
        Me.btnsave.Enabled = False
        Me.btnbg.Enabled = False
        Me.btnqty.Enabled = False
        Me.txtbg.ReadOnly = True
        Me.txtqty.ReadOnly = True
    End Sub

    Private Sub enaControl()
        Me.txtdatepic.Enabled = True
        Me.txtbg.ReadOnly = False
        Me.cbodonorid.Enabled = True
        Me.radrep.Enabled = True
```

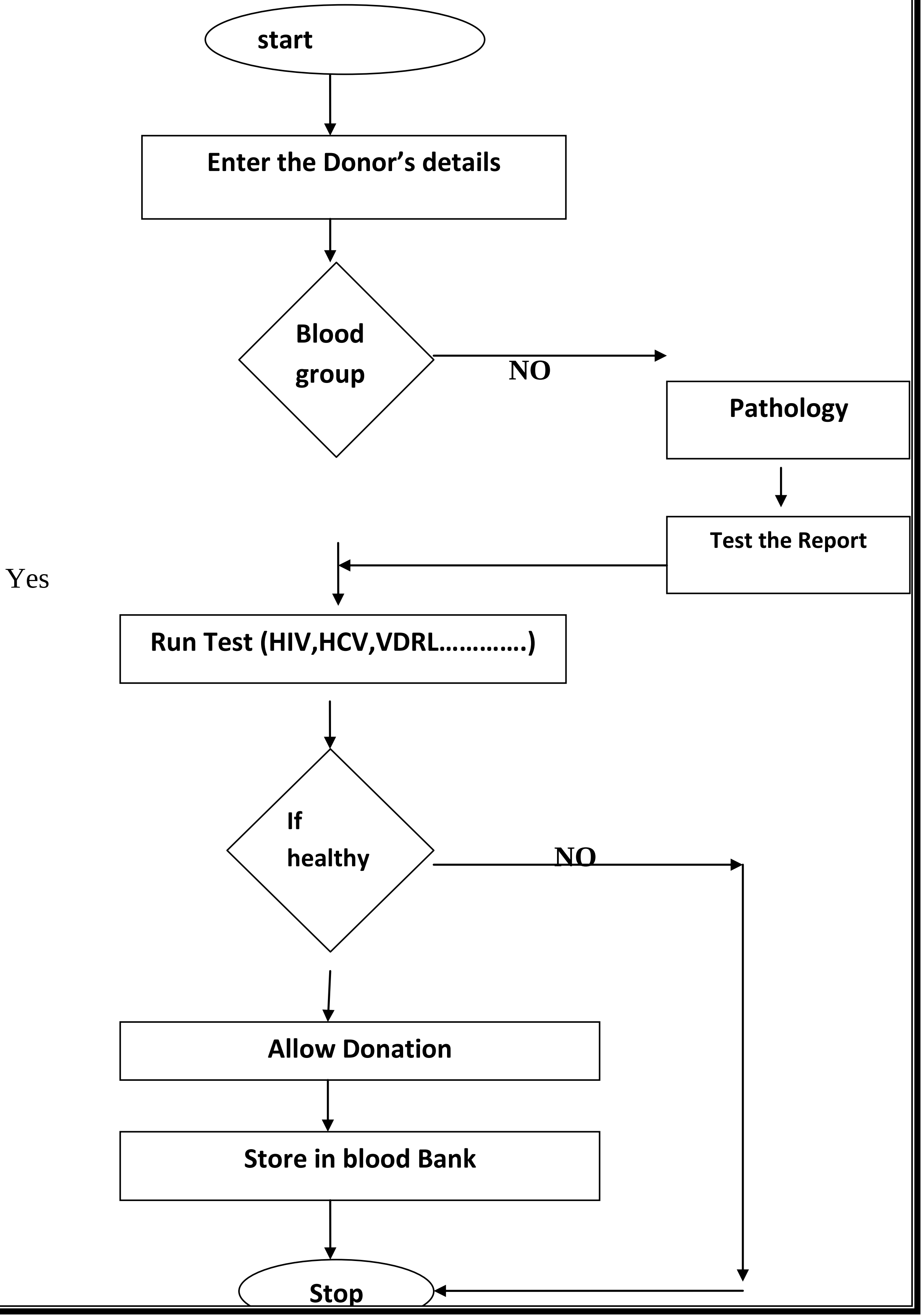
```
        Me.radbuy.Enabled = True
        Me.cbobg.Enabled = True
        Me.cbobbno.Enabled = True
        Me.txtcost.ReadOnly = False
        Me.btncancel.Enabled = True
        Me.btnsave.Enabled = True
    End Sub
    Public Sub showrecords()

        txtid.Text = ds.Tables(0).Rows(i)("issue_id").ToString()
        txtdatepic.Value = ds.Tables(0).Rows(i)("date").ToString()
        cborecid.Text = ds.Tables(0).Rows(i)("recid").ToString()
        cbobg.Text = ds.Tables(0).Rows(i)("bloodgroup").ToString()
        If (ds.Tables(0).Rows(i)("mode").ToString.Equals("Buy")) Then
            radbuy.Checked = True
        Else
            radrep.Checked = True
        End If
        cbodonorid.Text = ds.Tables(0).Rows(i)("donorid").ToString()
        cbobbno.Text = ds.Tables(0).Rows(i)("bloodbagno").ToString()
        txtcost.Text = ds.Tables(0).Rows(i)("tcost").ToString()
    End Sub

    Private Sub blood_issue_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        disControl()
        Call conecDB()
        Call initCMD()
        da = New SqlDataAdapter("select * from tblblood_issue", connDB)
        da.Fill(ds, "tblblood_issue")
        showrecords()
    End Sub
    Public Sub checkvalidate()
        'For validation

        If txtbg.Text = " " Or cbobbno.SelectedIndex = -1 Or cbobg.SelectedIndex = -1 Or txtcost.Text = " " Then
            flag123 = False
        End If

    End Sub


    Private Sub btnsave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnsave.Click

        'ds.Clear()
        flag123 = True
        If radrep.Checked = True Then  'for male female radiobutton
            str1 = radrep.Text
        Else
            str1 = radbuy.Text
        End If
        dated = txtdatepic.Value
        Call conecDB()
        Call initCMD()
        Call checkvalidate() 'For validation

        If flag123 = True Then

            issuedate = txtdatepic.Value
            SQL = "Select exp_date from tblaccepted where bloodbag_no='" & cbobbno.Text & "'"
            Call execComDB(SQL)
            dr = comDB.ExecuteReader
            dr.Read()
            expirydate = dr.Item("exp_date").ToString
            dr.Close()

            'Call disControl()
            If (expirydate > issuedate) Then     'to check the expiry date
                SQL = "Insert into tblblood_issue(issue_id,date,recid, bloodgroup, bgissued,mode, donorid,
bloodbagno,tcost)values('" & txtid.Text & "','" & dated & "','" & cborecid.Text & "','" & txtbg.Text & "','" &
cbobg.Text & "','" & str1 & "','" & cbodonorid.Text & "','" & cbobbno.Text & "','" & txtcost.Text & "')"
                SQL1 = "delete  from tblaccepted where bloodbag_no='" & cbobbno.Text & "'"
                Call execComDB(SQL)     'Execute the insert query
                Call execComDB(SQL1)
                'Me.Close()
                'MDIParent1.Show()
                MsgBox("Saved successfully")
                MsgBox("Print receipt?", MsgBoxStyle.YesNo, ) 'to print the receipt
                If (MsgBoxResult.Yes) Then
                    Stock_rep.Show()
                End If

            Else

                MsgBox("The blood bag has expired")
                'ds.Clear()
            End If
            ds.Clear()
            da = New SqlDataAdapter("select * from tblblood_issue", connDB)
            da.Fill(ds, "tblblood_issue")
            If (ds.Tables(0).Rows.Count > 0) Then
                i = 0
                showrecords()
            End If
        Else
            MsgBox("All fields required")
        End If

        closeDB()
    End Sub

    Private Sub btncancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btncancel.Click
        disControl()
        Call conecDB()
        Call initCMD()
        da = New SqlDataAdapter("select * from tblblood_issue", connDB)
        da.Fill(ds, "tblblood_issue")
        showrecords()
        btnnew.Enabled = True
    End Sub

    Private Sub btnedit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        enaControl()
        btnnew.Enabled = False
        btnsave.Enabled = False
        'btnupdate.Enabled = True
    End Sub


    Private Sub btnupdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        dated = txtdatepic.Value

        If radrep.Checked = True Then  'for male female radiobutton
            str1 = radrep.Text
        Else
            str1 = radbuy.Text
        End If
        Call conecDB()
        Call initCMD()
        Call disControl()
        If MsgBox("Are you sure you want to update the selected record?", CType(MsgBoxStyle.YesNo +
MsgBoxStyle.DefaultButton2 + MsgBoxStyle.Question, MsgBoxStyle), "Update") = MsgBoxResult.Yes Then

            SQL = "update tblblood_issue set date='" & dated & "',recid='" & cborecid.Text & "', bloodgroup='" &
txtbg.Text & "', bgissued='" & cbobg.Text & "',mode='" & str1 & "', donorid='" & cbodonorid.Text & "',
bloodbagno='" & cbobbno.Text & "', tcost='" & txtcost.Text & "' where issue_id='" & txtid.Text & "'"
            Call execComDB(SQL)     'Execute the insert query
        End If
        ds.Clear()
        da = New SqlDataAdapter("select * from tblblood_issue", connDB)
        da.Fill(ds, "tblblood_issue")
        showrecords()
        btnnew.Enabled = True
    End Sub
    Private Sub btndel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndel.Click
        Call conecDB()
        Call initCMD()
        Call disControl()
        If MsgBox("Are you sure you want to delete the selected record?", CType(MsgBoxStyle.YesNo +
MsgBoxStyle.DefaultButton2 + MsgBoxStyle.Question, MsgBoxStyle), "Update") = MsgBoxResult.Yes Then

            SQL = "delete  from tbldonor where donor_id='" & txtid.Text & "'"
            Call execComDB(SQL)     'Execute the insert query
            MsgBox("Record deleted successfully")
        End If

        ds.Clear()
        da = New SqlDataAdapter("select * from tblblood_issue", connDB)
        da.Fill(ds, "tblblood_issue")
        showrecords()
        btnnew.Enabled = True
    End Sub

    Private Sub btnbg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnbg.Click
        Call conecDB()
        Call initCMD()

        SQL = "Select bloodgrp from tblrecipient where rec_id='" & cborecid.Text & "'"
        Call execComDB(SQL)
        dr = comDB.ExecuteReader
        dr.Read()
        txtbg.Text = dr.Item("bloodgrp")
        dr.Close()
    End Sub

    Private Sub btnnew_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnnew.Click
        txtdatepic.Value = Now()
        btnbg.Enabled = True
        txtbg.Enabled = True
        btnqty.Enabled = True
        Call conecDB()
        Call initCMD()


        'ds.Clear()             ' for retrieving the recipient from tblrecipient
        da3 = New SqlDataAdapter("select rec_id from tblrecipient", connDB)
        da3.Fill(ds3, "tblrecipient")
        cborecid.DataSource = ds3.Tables("tblrecipient")
        cborecid.ValueMember = "rec_id"
        cborecid.DisplayMember = "rec_id"

        'ds.Clear()     'for retrieving the donor id from tbldonor
        da1 = New SqlDataAdapter("select donor_id from tbldonor", connDB)
        da1.Fill(ds1, "tbldonor")
        cbodonorid.DataSource = ds1.Tables("tbldonor")
        cbodonorid.ValueMember = "donor_id"
        cbodonorid.DisplayMember = "donor_id"


        Me.btncancel.Enabled = True
        Me.btnsave.Enabled = True
        btnnew.Enabled = False
        btndel.Enabled = False
        'Me.btnedit.Enabled = False
        'Me.btnupdate.Enabled = False
        txtdatepic.Value = Now()

        da = New SqlDataAdapter("select * from tblblood_issue", connDB)

        da.Fill(ds, "tblblood_issue")
        i = ds.Tables(0).Rows.Count - 1 'go to last record
        txtid.Text = ds.Tables(0).Rows(i)("issue_id").ToString + 1
        cleartxtbox()
        enaControl()
    End Sub
    Private Sub btnprev_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnprev.Click
        If (i.Equals(ds.Tables(0).Rows.Count - 1) Or (i > 0)) Then

            i = i - 1
            showrecords()
        Else
            MsgBox("You have reached the first record", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub btnfirst_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnfirst.Click
        If (ds.Tables(0).Rows.Count > 0) Then

            i = 0
            showrecords()

        End If
    End Sub

    Private Sub btnnext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnnext.Click
        If (i < ds.Tables(0).Rows.Count - 1) Then
            i = i + 1
            showrecords()
        Else
            MsgBox("You have reached the last record", MsgBoxStyle.Critical)

        End If
    End Sub

    Private Sub btnlast_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnlast.Click
        da = New SqlDataAdapter("select * from tblblood_issue", connDB)
        da.Fill(ds, "tblblood_issue")
        i = ds.Tables(0).Rows.Count - 1
        showrecords()
    End Sub

    Private Sub btnqty_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnqty.Click
        ds.Clear()
        strbg = cbobg.Text
        da = New SqlDataAdapter("select * from tblaccepted where blood_group like '" + strbg + "'", connDB)
        da.Fill(ds, "tblaccepted")
        count = ds.Tables("tblaccepted").Rows.Count.ToString()
        txtqty.Text = count

        ds2.Clear()
        da2 = New SqlDataAdapter("select * from tblaccepted where blood_group='" & cbobg.Text & "'", connDB)
        da2.Fill(ds2, "tblaccepted")
        cbobbno.DataSource = ds2.Tables("tblaccepted")
        cbobbno.ValueMember = "bloodbag_no"
        cbobbno.DisplayMember = "bloodbag_no"
    End Sub
End Class
```

Recipient’s Perspective

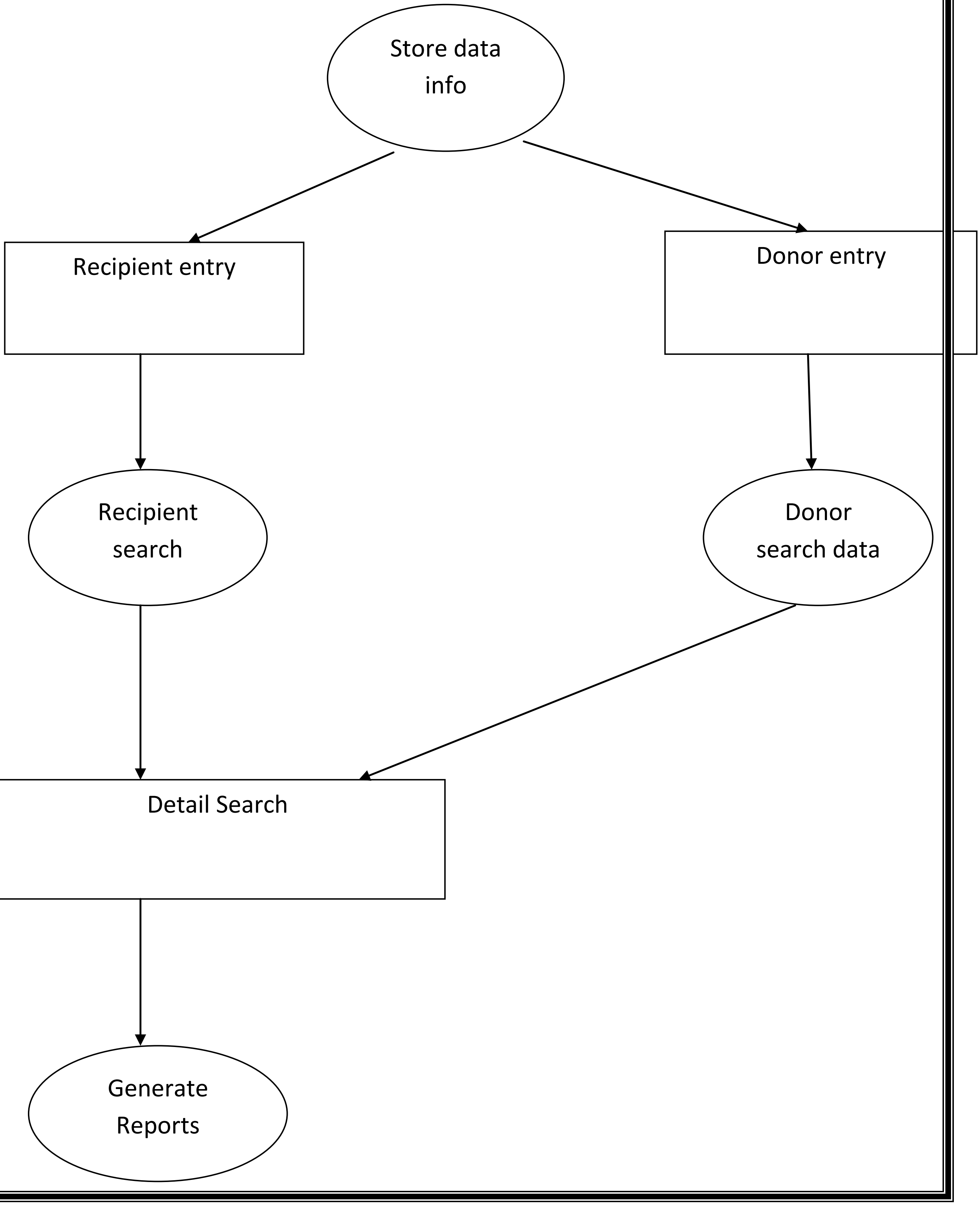


Yes

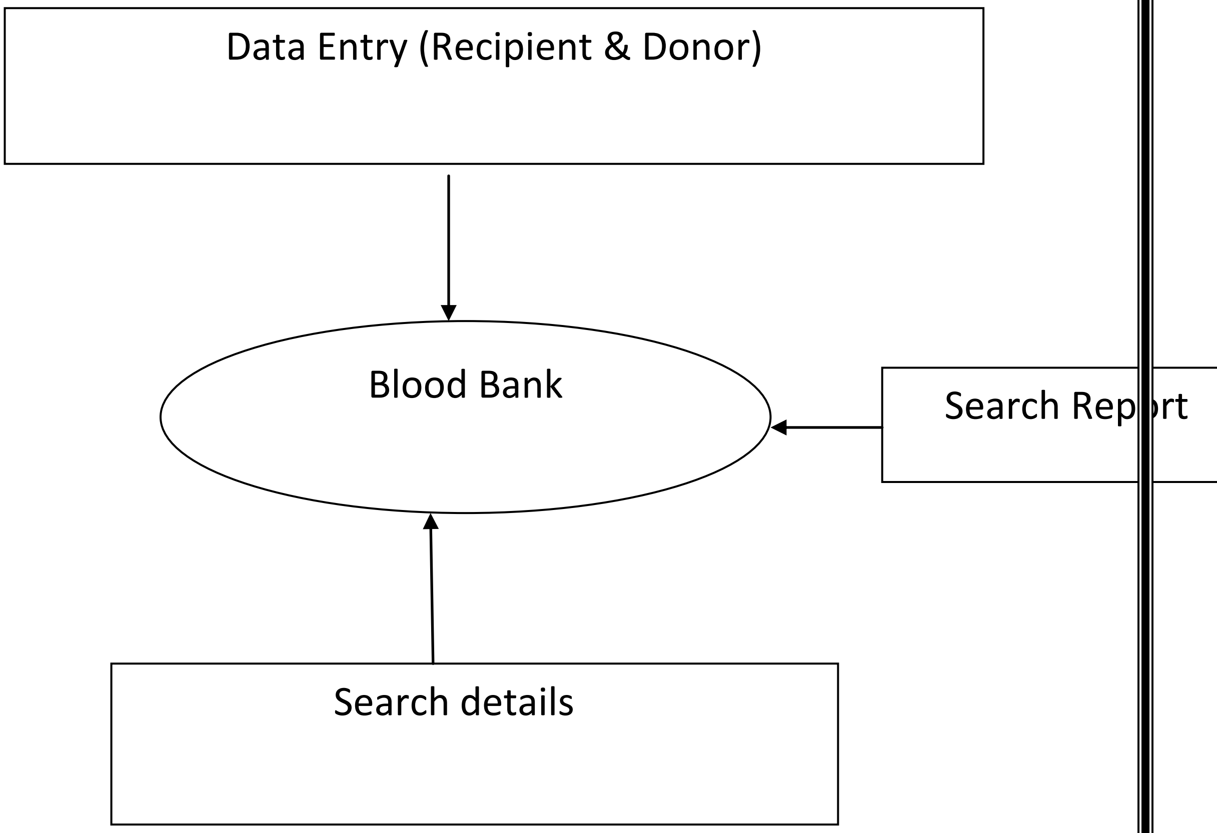
Donor's Perspective

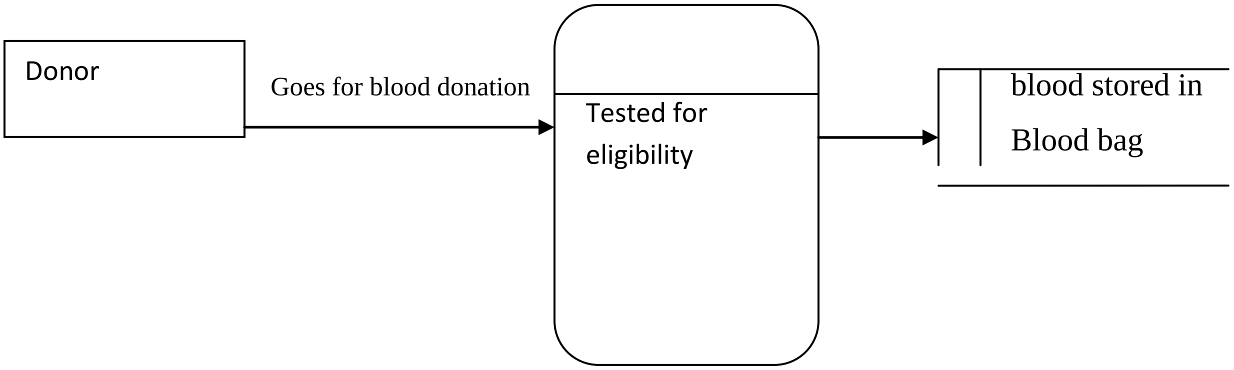


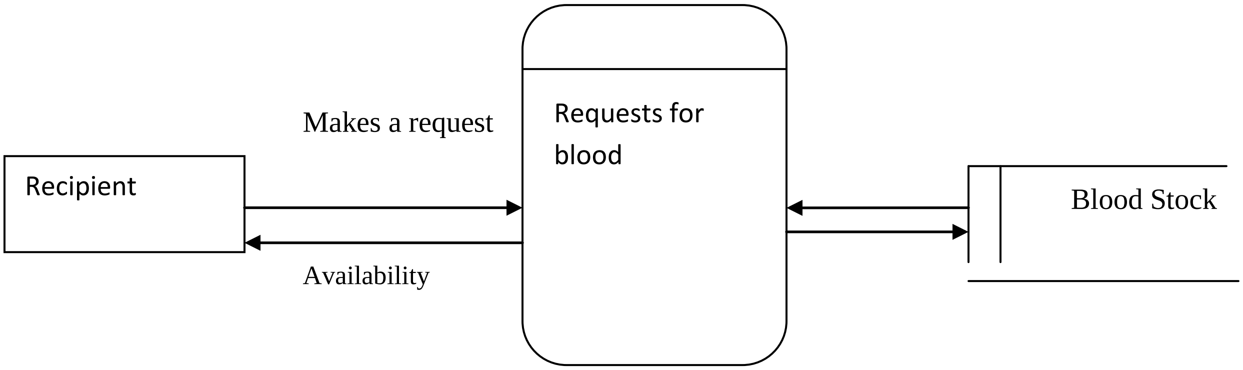
DFD (LEVEL 1)


Store data info
Recipient entry
Donor entry
Recipient search
Donor search data
Detail Search
Generate Reports

DFD (Context free)




Donor
Goes for blood donation
Tested for eligibility
blood stored in Blood bag


Makes a request
Recipient
Availability
Requests for blood
Blood Stock