**INTRODUCTION**

Data is the code word of the computer industry. Data refers to a collection of facts usually collected as a result of observation and experiment or processes within a computer system. This may consist of numbers, words or images or observations of a set of variables. Data are often viewed as a lowest level of abstraction from which information and knowledge are derived.

In this project we have given an insight to the development of a computer based database titled **"BLOOD BANK MANAGEMENT**" using Microsoft Visual Basic 6.0 as a front-end and Microsoft Access as a back-end.

Blood Bank automation or computerization of the blood bank is the application of computers to perform either in part or as a whole the activities that are carried out in the blood bank setting.

Blood bank automation requires the use of computers and associated programs that are suited for the purpose. These associated or group of programs that are use on the computer system can be describe as Pharmacy Management System.

Blood Bank management system is used to perform tasks such as registration of customers, blood and staffs. It helps the customer to purchase the required blood bases on their needs. Customer can make transaction through cash. Information related to blood, employees, customers, etc can be easily retrieved.

This software is required to eliminate tedious manual searching, cataloguing, and loaning out of medicines, Including registration and clearance of customers.

SYNOPSIS

This system is mainly based on collection, storage and usage of blood in needy situations. City life is turning hectic as we move in 21st century. Day by day all the newspapers that were filled with motivating articles and some important news is now filled with either Bollywood stuffs or some kinds of crime, bomb blasts becoming one of the main events. Hence the need of blood is the crucial subject of consideration. Also many diseases are seen to be creped in our day to day life. One of which can be instanced as Thallesmania. One that is affected by this disease also needs the transmission of the blood week by week. Hence blood bank is the most suitable option then.

**PROJECT**

BLOOD BANK MANAGEMENT is a software application to maintain day to day transactions in a blood bank. This software help to register all the donors, Blood collection details, blood issued details etc.,

**OBJECTIVE**

The main objective of this application is to automate the complete operations of the blood bank. They need maintain hundreds of thousands of records. Also searching should be very faster so they can find required details instantly.

**SCOPE**

This application is built such a way that it should suits for all type of blood banks in future. So every effort is taken to implement this project in this blood bank, on successful implementation in this blood bank, we can target other blood banks in the city.

**Main modules of the project:**

This project has the following modules, to manage all the requirements of the blood bank.

1. Blood bank details
2. Donor Details
3. Hospital Details
4. Outsider details
5. Equipment details
6. Blood collection details
7. Blood Issued Details
8. Camp details
9. Stock Details
10. Reports

To manage employees in the blood bank it had the following modules

1. Employee Details
2. Employee Salary Payment
3. Reports

**TOOLS INFORMATION**

* **Hardware Requirement**

Processor : Intel Core Duo 2.0 GHz or more

RAM : 1 GB or More

Hard disk : 80GB or more

Monitor : 15” CRT or LCD monitor

Keyboard : Normal or Multimedia

Mouse : Compatible mouse

* **Software Requirement**

Operating System : Windows XP, Windows 7

Front End : VISUAL BASIC 6.0

Back End : Microsoft Access 2007

TOOLS USED

**FRONT END: Visual Basic 6.0**

VISUAL BASIC is a high level programming language which evolved from the earlier Dos version called BASIC.BASIC means Beginner’s All – Purpose Symbolic Instruction Code. It is a very easy Programming language to learn.

The code looks a lot like English Language. Different software companies produced different versions of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASIC and so on. However, people prefer to use Microsoft Visual Basic today, as it is a well developed programming language and supporting resources are available anywhere. Now there are many versions of VB exist in the market, the most popular one and still widely used by many VB programmers is none other than Visual Basic 6.We also have VB.net, VB2005, VB2008 and the latest VB2010.Both Vb2008 and VB2010 are fully object oriented programming (OOP) language.

VISUAL BASIC is a VISUAL and events driven programming Language. These are the main divergence from the old BASIC. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In VB, programming is done in a graphical environment. In the old BASIC, you have to write program code

For each graphical object you wish to display it on screen, including its position and its color. However, In VB, you just need to drag and drop any graphical object anywhere on the form, and you can change its color any time using the properties window.

BACK END: Microsoft Access

Microsoft Access is a “Relational Database Management System.” The description that follows applies to Microsoft Access 2000, Microsoft Access 2002, Microsoft Access 2007 Microsoft Access 2007 & 2010, even Microsoft Access 97. In fact what follows applies to just about every Windows database out there regardless of who makes it.

**What it Does**

Access can store data in specific format for sorting, querying, and reporting. Sorting is pretty straightforward; data is simply presented to you in particular orders. An example might be presenting your customer data (customer number, name, address, city, state, zip, and total purchases) in last name order.

Querying means that as a user of this database, you can ask Access for a collection of information relating to location such as state or country, price as it might relate to how much a customer spent, and date as it might relat4 to when items were purchased.

Querying can include sorting as well. For example if you wanted to see the top spending customers in the state of Florida querying would be a way to do that. A Query on data typically returns a sub-set of the collection of data, but can return all of it in a different order as well.

Reporting is simply query results in printable or viewable form.

**How the Data Are Stored**

In order for Access to perform these functions data has to be stored in the smallest possible units. These units are called fields. A field might contain a first name, a last name, a middle name, a street address, and so on. Notice that I do not propose that the entire name be placed in one field. If that were done sorting one could perform would end up being presented by the first name. But if a separate field is used for the last name, another foe first, and so on, much more useful sorting can be accomplished.

Fields are also defined as a type of data (number, text, date, date-time, dollar, etc.). By storing data in its own specific field type, Access (or any RDBMS for that matter) can sort that data in very tightly controlled as long as Access knows what type of sort to apply to that data.

SYSTEM ANALYSIS

This project has been developed to automate data handling in Blood Bank Management. The different forms have been used to input data, make changes to the existing data, delete some records according to the requirements. The different reports give the user a Firth and knowledge about various details.

This project opens with a splash screen clicking on which takes the user to the screen having login form by clicking on it, the respective password entry form opens. On entering the correct password the main MDI form is invoked, where the user can select the desired form or report from the menu.

* The Cancel button associated with every form cancels the currently performed operation and not the form.
* To exit from the form click the “x” button available on the top corner of every Form.
* It has not been possible to cater to all the possible quires and the project can be upgraded according to the requirements.

There are some important modules in the purpose system, which are:

* **Login for admin:** This module will allow the administrator to enter their respective forms.
* **Donor details:** This module will enable admin to add new donor and to change their details.
* **Hospital details:** This module will enable the Administrator to add new Hospital details, modify their details. The Hospital ID is generated automatically. He can also generate Hospital transaction and allows printing.
* **Employee details:**  This is maintained by the administrator. He can modify the details of the employee, salary of the employees.
* **Outsider details:** This module enables the administrator to add new outsider details, modify their details. The Outsider ID is generated automatically. He can also generate Outsider transaction and allows printing.

EXISTING SYSTEM

In the manual system, firstly the Blood bank and its staff have to manage information regarding the donor details of all the donors manually. Doing this manual transaction was really tedious job. Secondly information regarding Hospital transactions, employee details was to be maintained. This process is time consuming and it requires a great manual effort.

**Disadvantages:**

* More time is consumed.
* More hard work to maintain all records.
* Bulk of paper is to be searched for a single search.

PROPOSED SYSTEM

Today one cannot afford to rely on the fallible human beings to stand against the

Merciless competition where it is not wise to say “to err is human” no longer valid.

It is outdated to rationalize your mistake. So, to keep pace with time, to bring about

The best result without malfunctioning and greater efficiency we have to replace the unending heaps of files with a much sophisticated hard disk of the computer. One has to use the data management software. Software has been an ascent in automation of various organizations. Many software products working are now in markets, which have helped in making the organizations work easier and efficient. Data management initially had to maintain a lot of ledgers and a lot of paperwork has to be done but now software production, this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done. This prevents a time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it’s age of computers and automating such an organization gives the better look.

The advantages of the proposed system as follows:

* To reduce the workload.
* To reduce the processing time.
* To view the details of student and staff and to maintain reliable and update information.
* Easy accessibility to computerized online report.

FEASIBILITY STUDY

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a feasibility study. Once it has been determined that the project is feasible keeping the benefit of the organization in mind, the analyst can go ahead and prepare the project specification, which finalizes the project requirements. Different tests of feasibility are studied during the investigation.

The main of them are-

* **Technical feasibility**
* **Economic feasibility**
* **Operational feasibility**

TECHNICAL FEASIBILITY

This is considered with specifying equipment and software that will successfully satisfy the user requirement. It involves determining whether or not a system can actually be constructed or upgraded to solve the problem at hand. The technical needs of a system may vary considerably, but might include and following:

* The necessary technology of both hardware and software existed and also could be acquired for the new system.
* As the improvement in storage technology has developed over the recent years, data storage is becoming easier and safer.
* The new system with powerful database technology has the capability to hold the proposed data.
* Technically, the system is designed in such a way that is provides accuracy, reliability, easy access, data security and integrity.

ECONOMICAL FEASIBILITY

Economic feasibility involves estimating benefits and costs. These benefits and costs may be tangible or intangible. It is seen whether the expenditure incurred for developing the new system will be cost effective or not. Because of the confusion between the types of costs, it is sometimes very difficult to decide if the benefits outweigh the costs. The basically involves top-level management of the company who are the decision makers. Some key fin dings from the study are listed below.

* There was no extra cost burden to conduct full systems investigations.
* Basic hardware and software would ensure the smooth run of the application, as the necessary tools were easily available.
* The benefits are in the form of reduced costs with merely any errors, thus reducing the manual work.
* If the system were used without any major changes to it, no extra costs would be incurred.

**OPERATIONAL FEASIBILITY**

Operational feasibility deals with the human aspect of the organization, proposed projects are beneficial only if they can be turned into information systems that will meet the organization’s operating requirements. This feasibility test asks whether the system will work when developed and installed, the users need to be convinced about the advantages of the new system.

Unless this done effectively, the system would not be implemented even after its development and the old system would continue to be used.

**PROBLEM SPECIFICATION**

For the present fashion in the shopping system premises there is no solution as such to provide the convenience of supporting multiple platforms and technologies as there are no three tier based solution. The present solution is using web technologies for the user. Any changes in the business logic requires lot of system studies in terms coding and in terms of new hardware and software specifications and the possibilities of integrating the web services is zero. For any small changes requires the understanding of the business logic of the entire organization. There is no job classification such as:

* System analyst and business logic developer
* Hardware programmer
* Application assembler
* Application developer

In the present scenario of the most of the responsibilities like coding and integration of business logic is done by the software engineers. There is no segregation of tasks. So it is very difficult to manage the enhancements of the solutions.

**SYSTEM DESIGN**

A computer procedure is a series of operations designed to manipulated data to produce outputs from a computer system. This procedure may be a single program or series of program. This details design of the computer procedure allows the acceptance lower levels of details, which will define the detailed steps to be taken to produce a specified computer output. When complete, these procedures definitions together with data specifications are organized by the programmers for which required programs can be written.

DESIGN TOOLS

Various tools are being used by a system analysis to specify computer procedures. Not all of them are used here to design this project.

* E-R diagram
* Input design
* Output design

E-R DIAGRAM

Donor

Blood

Donates

Stored

Hospital

Registers

Orders

Works

Receptionist

Blood Bank

Issues

Blood Bag

Data Base Design

**Login**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Username | Text |
| Password | Text |

**Blood Bag Details**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| BloodBag\_ID | Text |
| Donor\_ID | Text |
| BloodBank\_Name | Text |
| BloodGroup | Text |
| Status | Text |
| Collection\_Date | Date/Time |
| Expiry\_Date | Date/Time |
| EnteredBy | Text |
| Amount | Number |

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Donor\_ID | Text |
| Name | Text |
| Address | Text |
| Gender | Text |
| Age | Number |
| MaritalStatus | Text |
| Phone\_No | Number |
| BloodGroup | Text |
| Occupation | Text |
| Email\_ID | Text |
| LastDonateDate | Date/Time |

**Donor Details**

**Donor Health Information**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Donor\_ID | Text |
| Body\_Weight | Number |
| Temperature | Number |
| Pulse\_Rate | Number |
| Blood\_Pressure | Number |
| Haemoglobin | Number |
| Bag\_Weight | Number |
| TestVDRL | Text |
| TestHBSAG | Text |
| TestMP | Text |
| TestHCV | Text |
| TestHIV | Text |

**Hospital Details**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Hospital\_ID | Text |
| Hospital\_Name | Text |
| Address | Text |
| City | Text |
| Pin\_Code | Number |
| Phone\_No | Number |

**Other Blood Bank Details**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Name | Text |
| Address | Text |
| City | Text |
| PinCode | Number |
| Phone\_No | Number |

**Outsider Details**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Outsider\_ID | Text |
| Outsider\_Name | Text |
| Address | Text |
| Age | Number |
| Gender | Text |
| Phone\_No | Number |
| Hospital\_Name | Text |
| BloodGroup | Text |

**Camp Schedule**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Code | Text |
| Name | Text |
| Address1 | Text |
| Address2 | Text |
| Address3 | Text |
| PinCode | Number |
| Contact\_Person | Date/Time |
| Phone\_No | Number |
| Scheduled\_Date | Date/Time |
| No\_Of\_Beds | Number |

**Stock Details**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| BloodGroup | Text |
| NoOfBags\_Available | Number |

**Employee Details**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Employee\_ID | Text |
| Employee\_Name | Text |
| Gender | Text |
| DOB | Number |
| Age | Number |
| Qualification | Text |
| Designation | Text |
| Salary | Number |
| Address1 | Text |
| Address2 | Text |
| Address3 | Text |
| Pincode | Number |
| City | Text |
| Phone\_No | Number |
| Blood\_Group | Text |
| Email\_ID | Text |
| Joining\_Date | Date/Time |

**Employee Salary**

|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Salary\_Month | Date/Time |
| Employee\_ID | Text |
| Employee\_Name | Text |
| Attended\_Days | Number |
| Basic\_Pay | Number |
| DA | Number |
| HRA | Number |
| Deductions | Number |
| Gross\_Salary | Number |
| Net\_Salary | Number |
| Pay\_Date | Date/Time |

**Equipment Details**

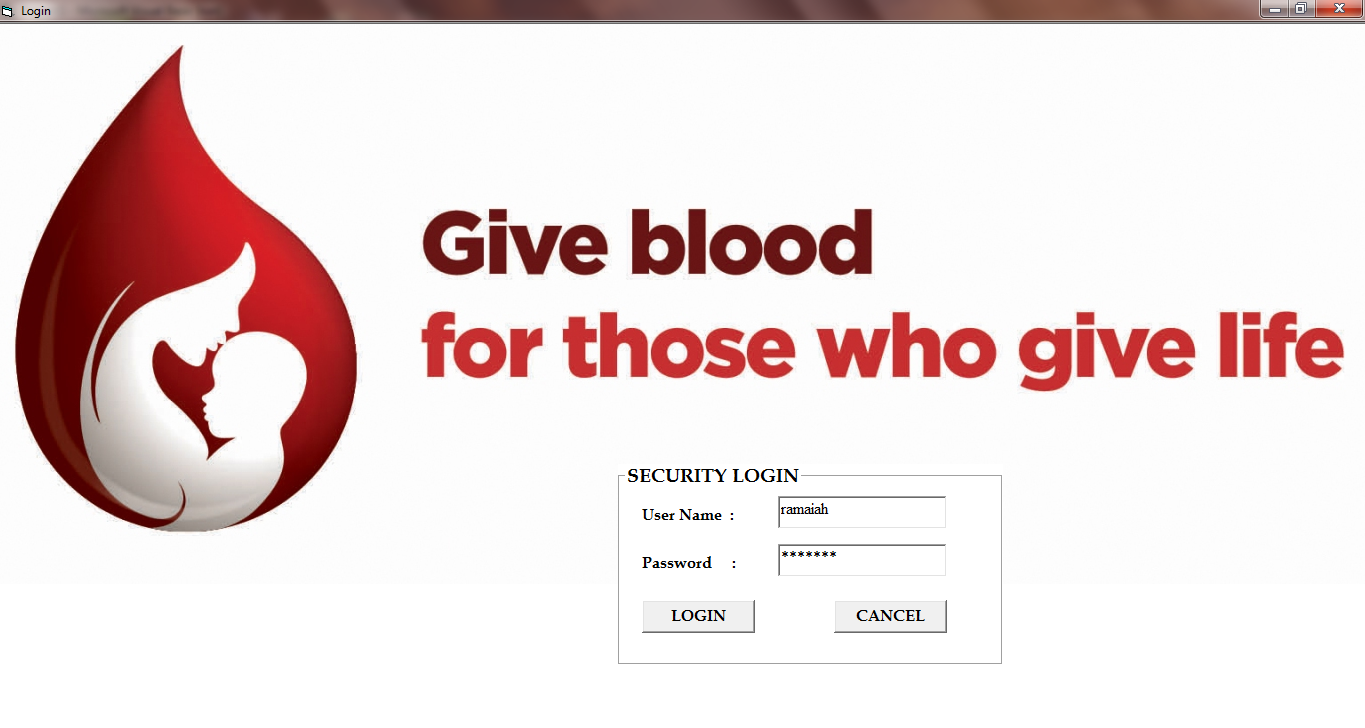
|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Equipment\_Code | Text |
| Equipment\_Name | Text |
| Model\_No | Text |
| Capacity | Text |
| Purchased\_On | Date/Time |
| PurchasedFrom | Text |
| Bill\_No | Text |
| Cost | Number |

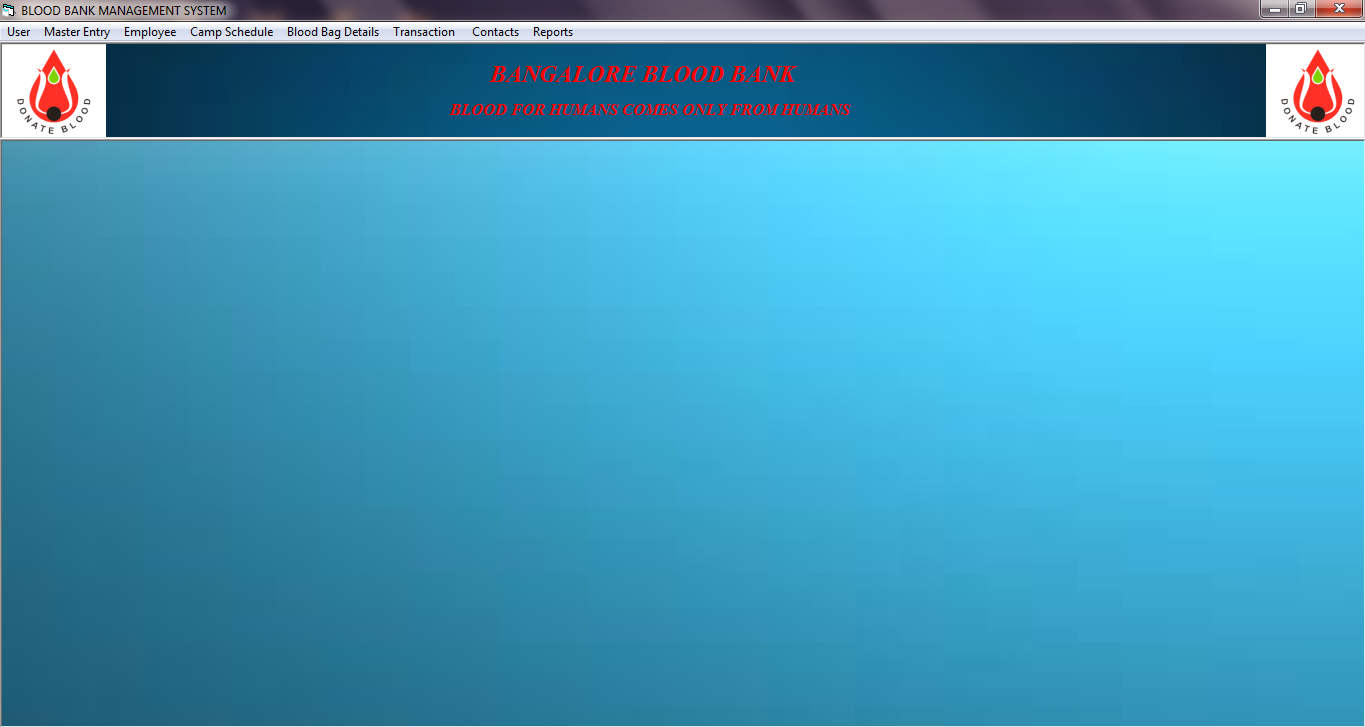
**Transaction**

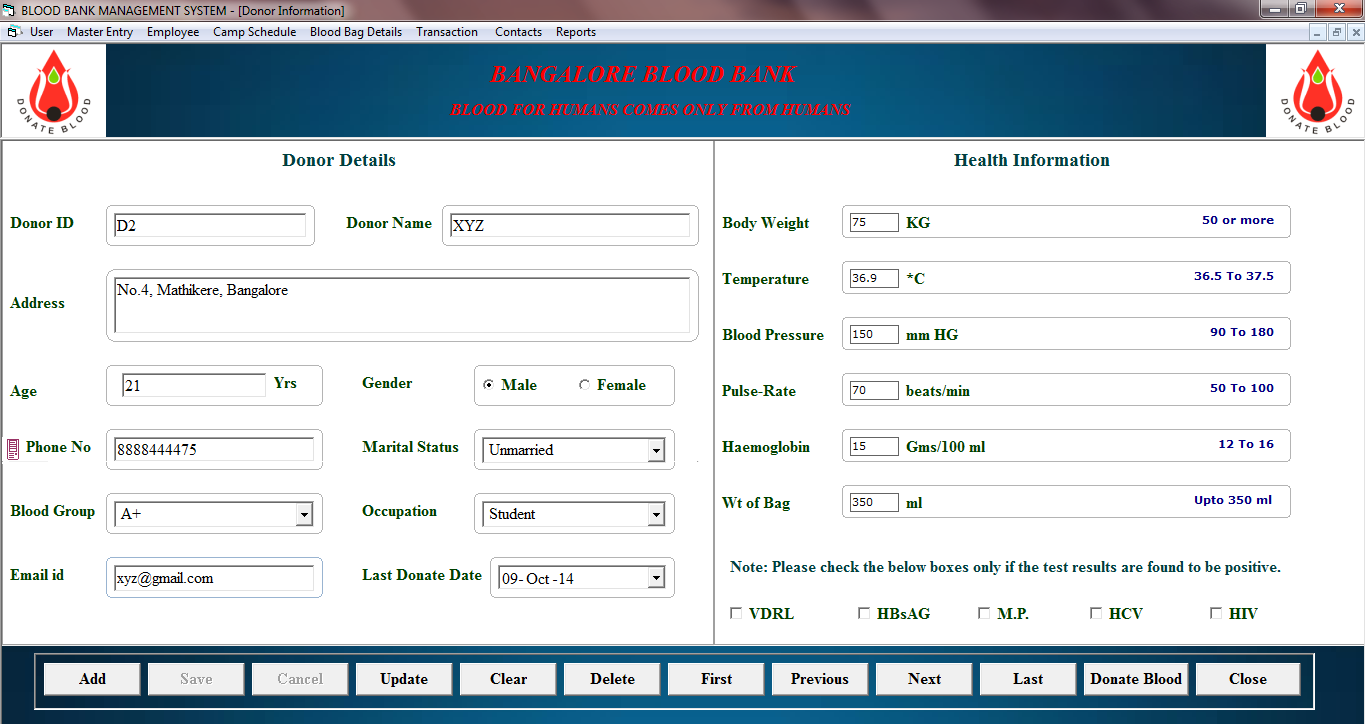
|  |  |
| --- | --- |
| **Field Name** | **Data Type** |
| Transaction\_ID | Text |
| Recipient\_Type | Text |
| Recipient\_ID | Text |
| Recipient\_Name | Text |
| Date | Date/Time |
| BloodGroup | Text |
| Stock | Number |
| NoOfBags | Number |
| BloodBag\_ID | Text |
| Rate | Number |
| Amount | Number |

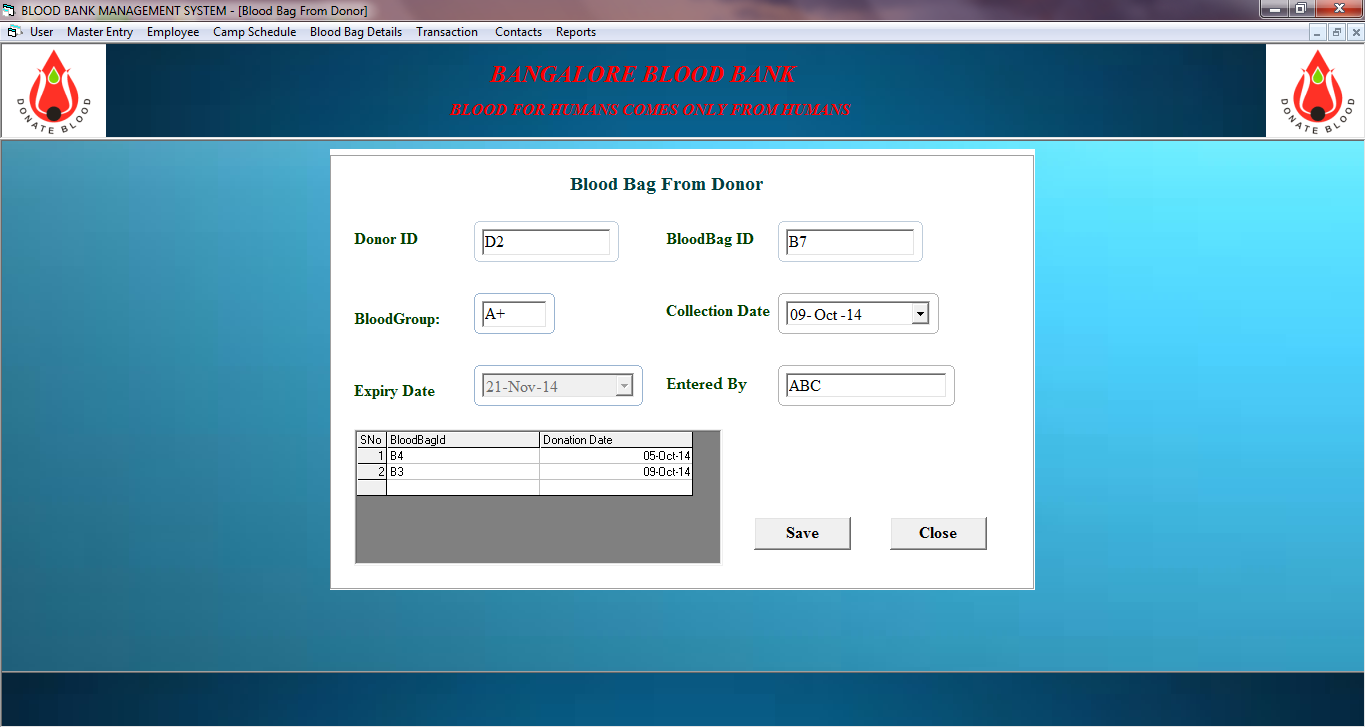
SCREEN SHOTS

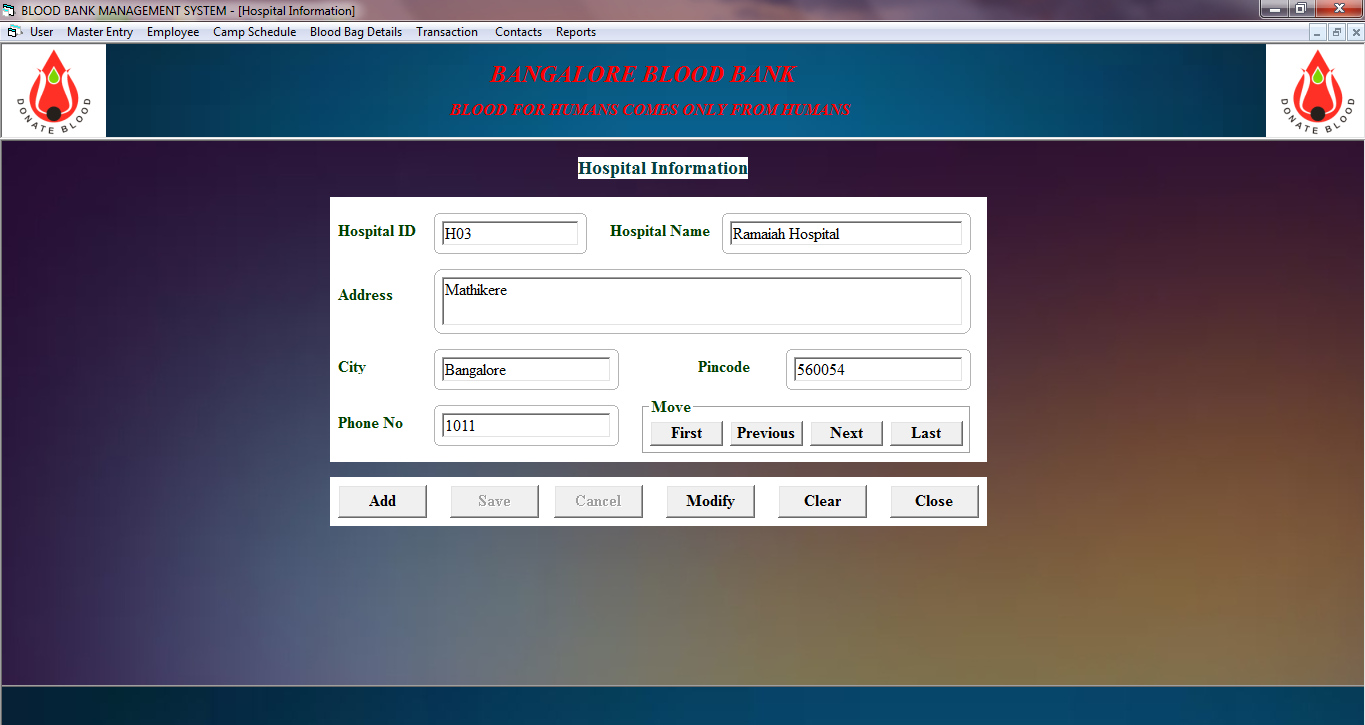
**Login Form:**



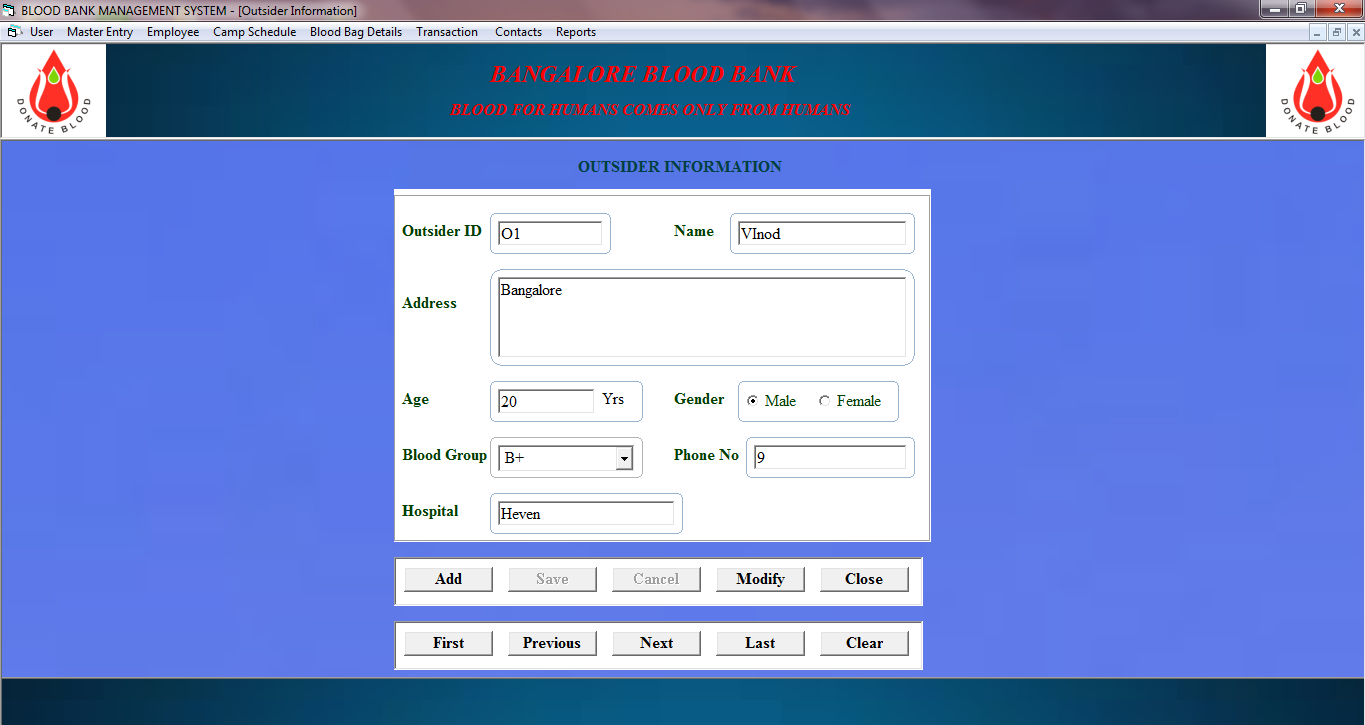
**MDI Form:**

**Donor Details**

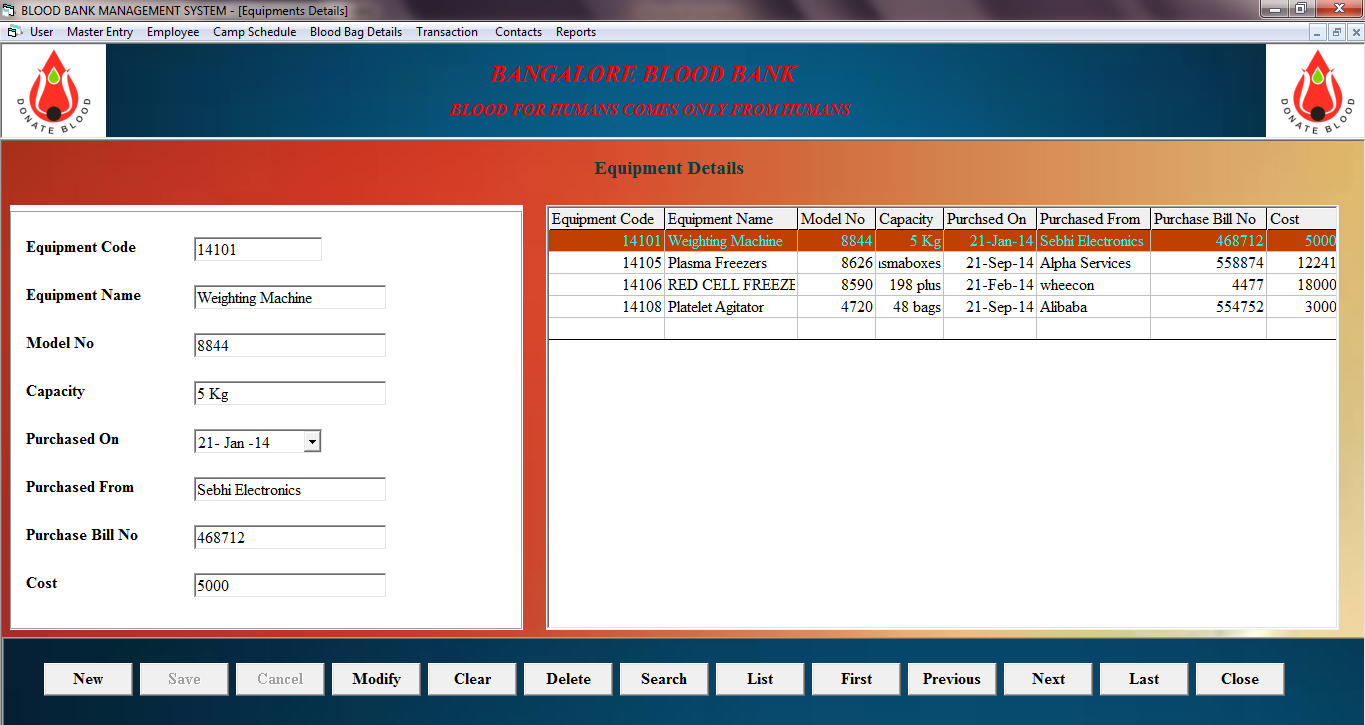
**Blood Bag from Donor**

**Hospital Details**

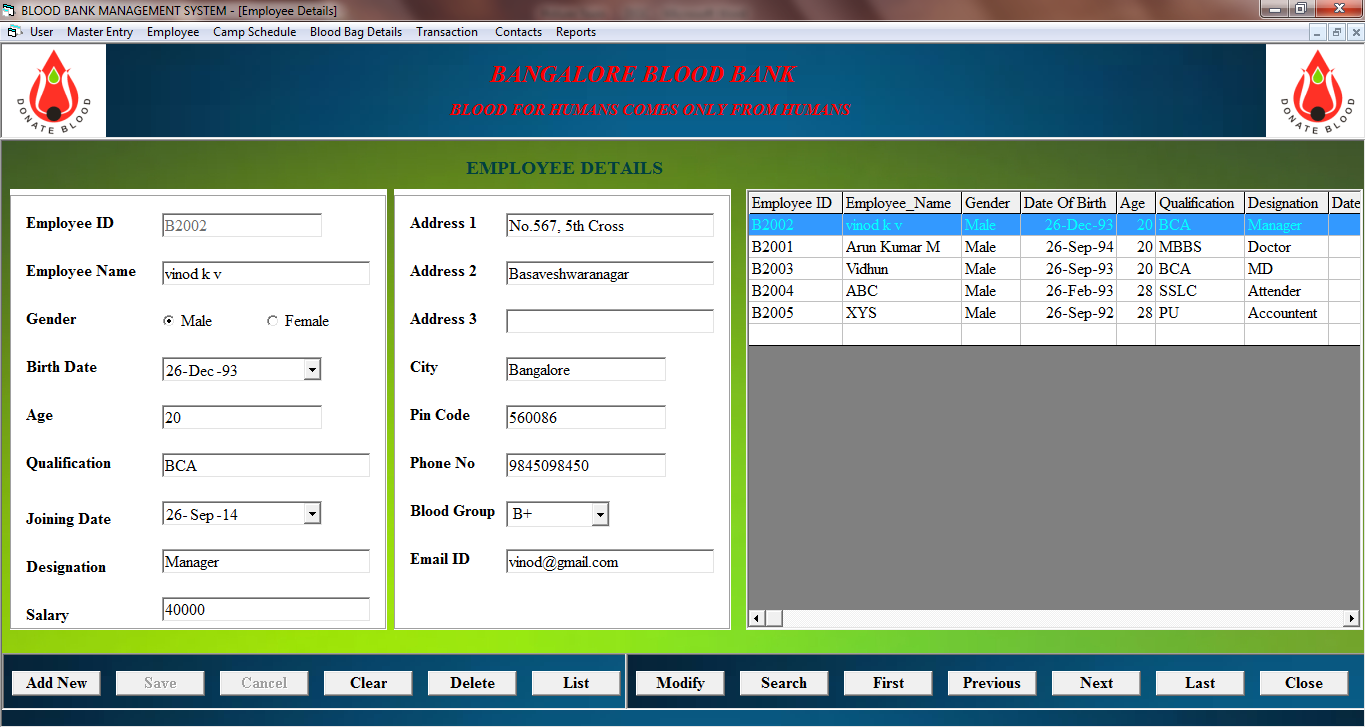
**Outsider Information**

****

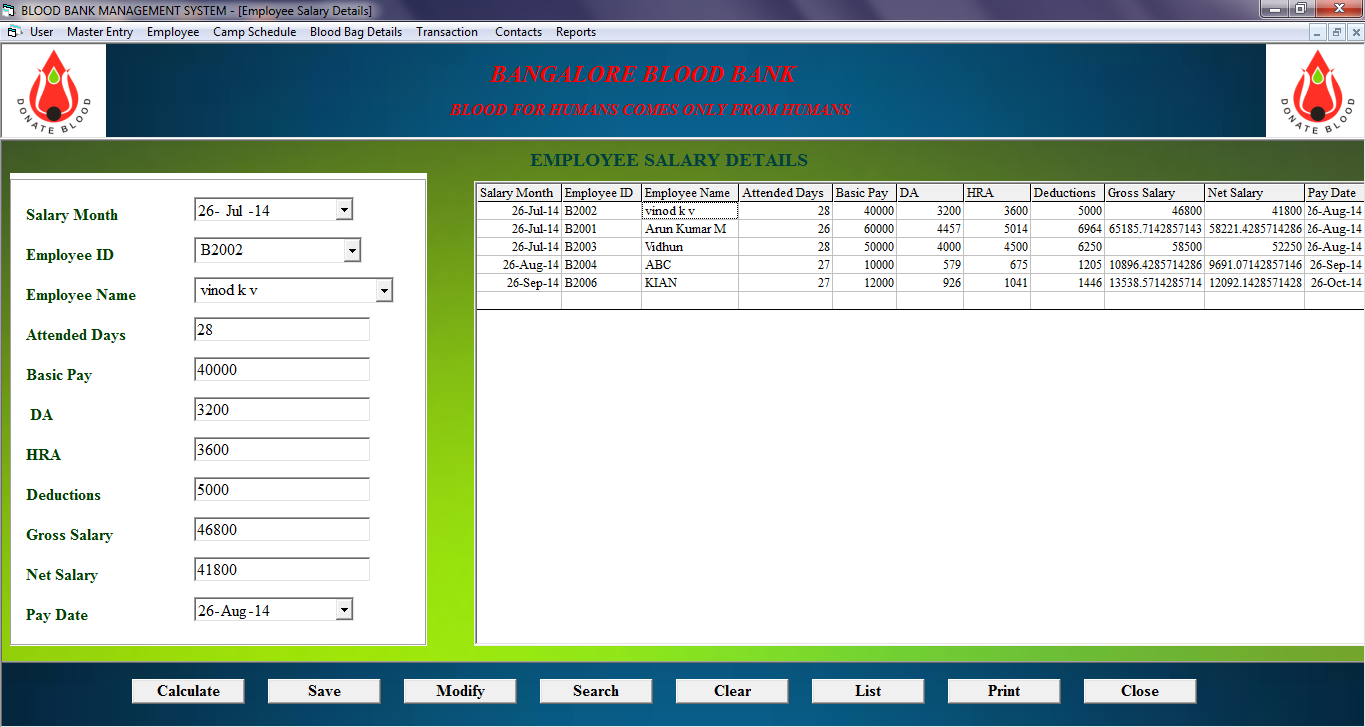
**Equipment Details**

****

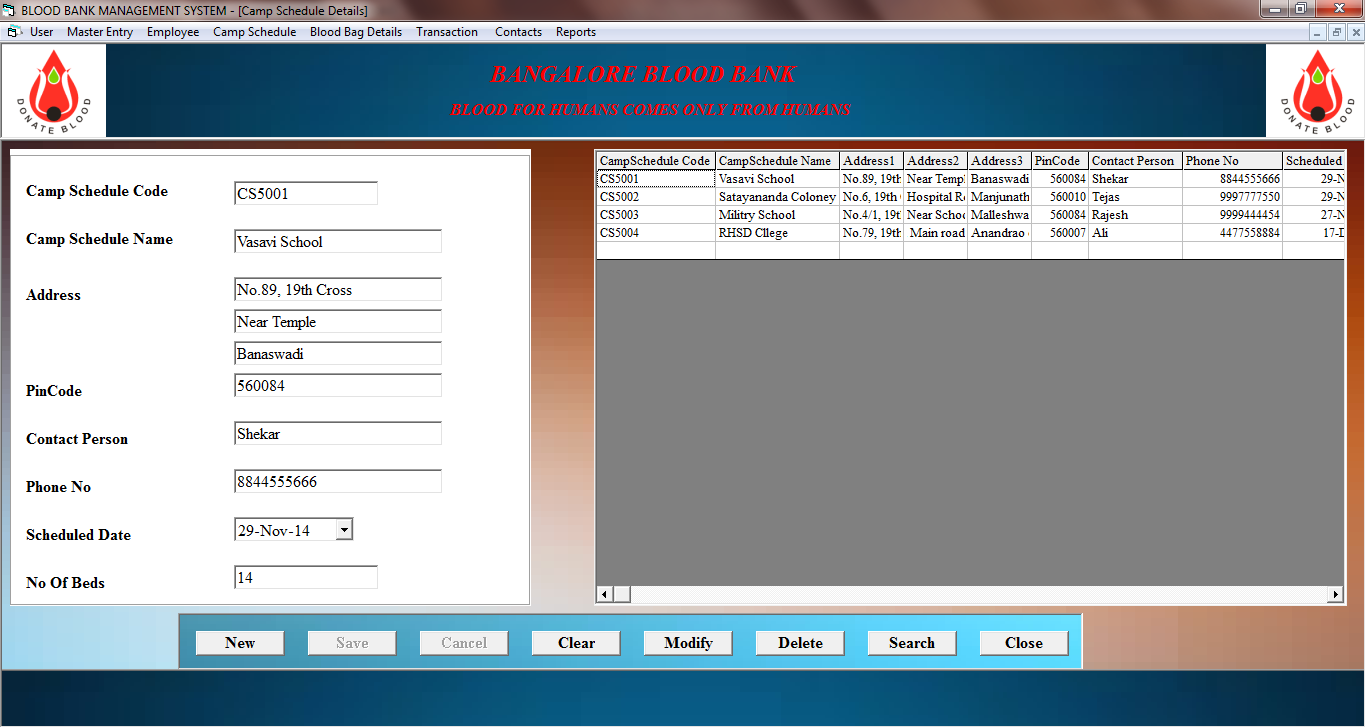
**Employee Details**

****

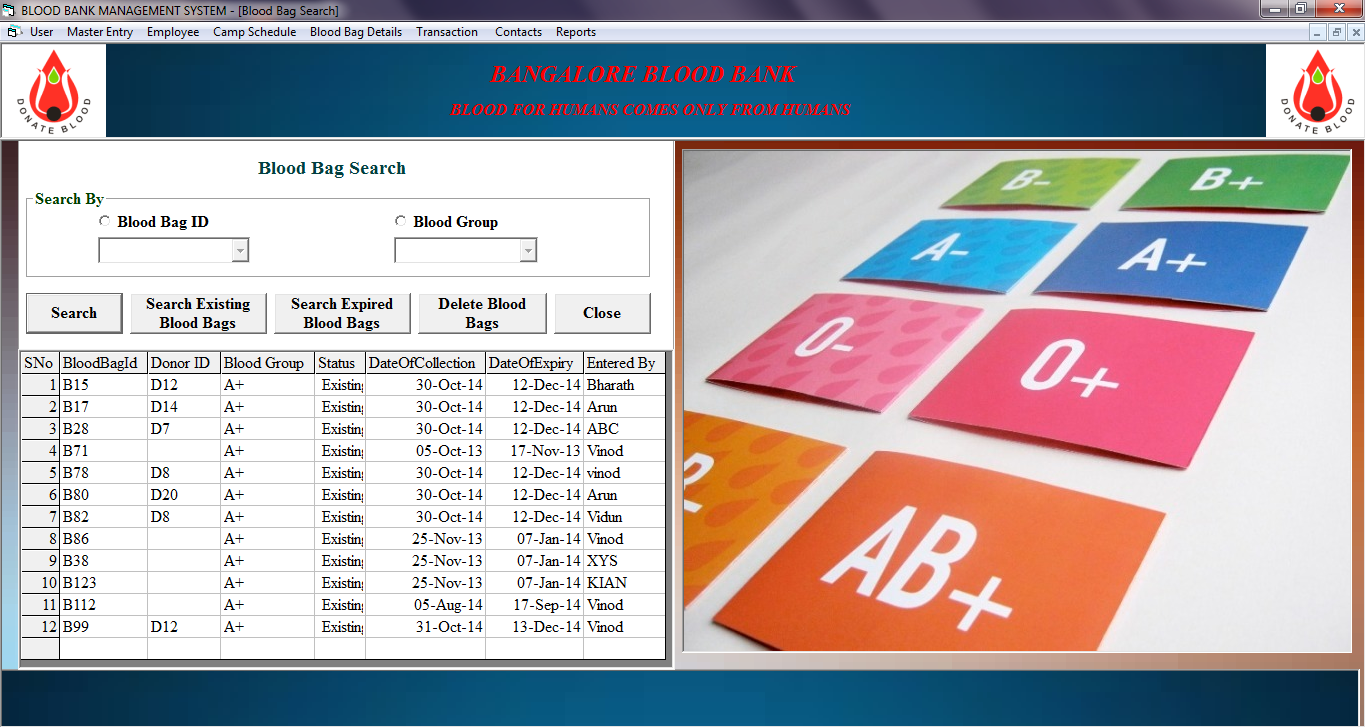
**Employee Salary Details**

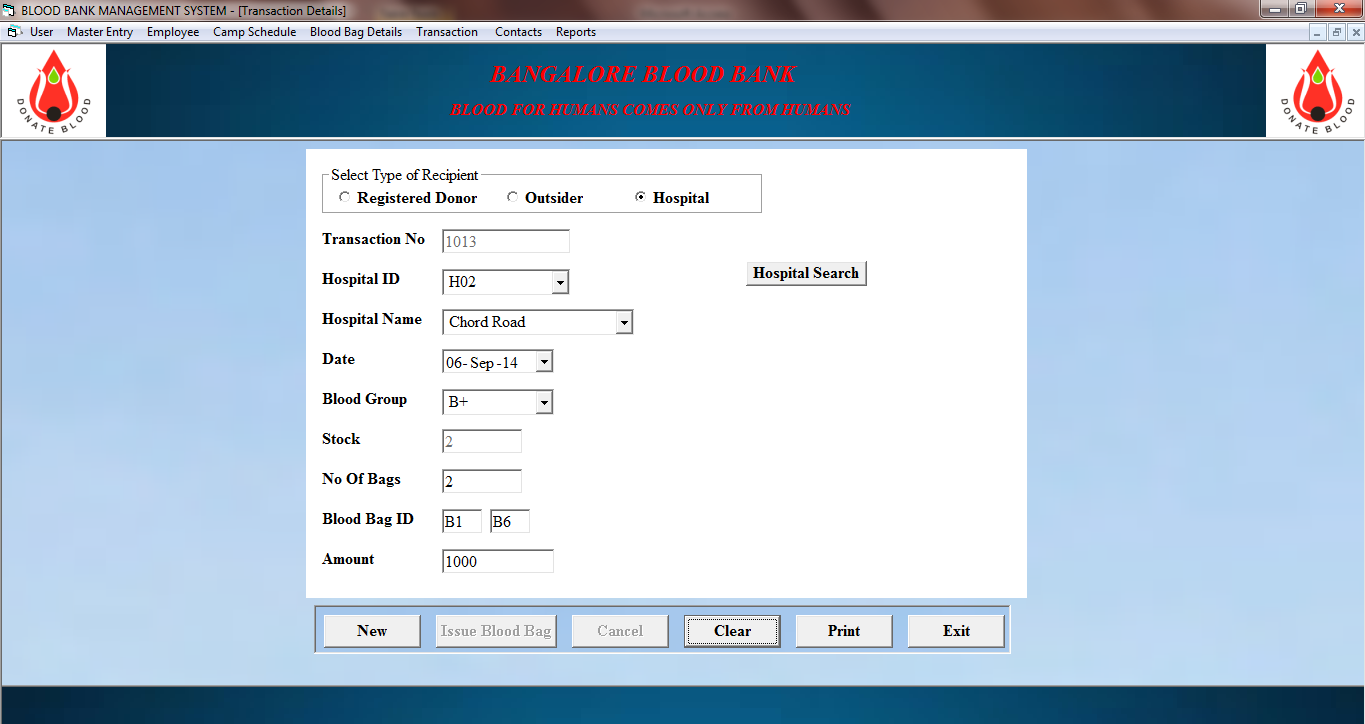


**Camp Details**

****

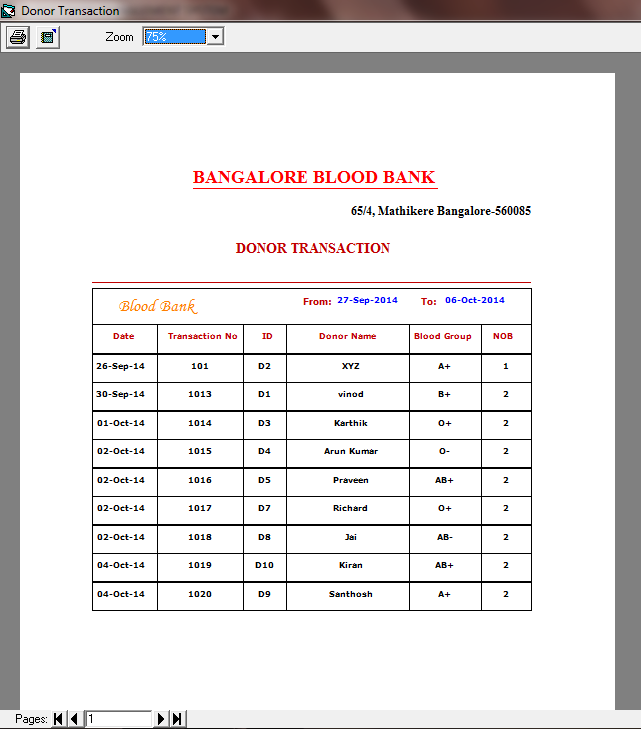
**Blood Bag Search**

****

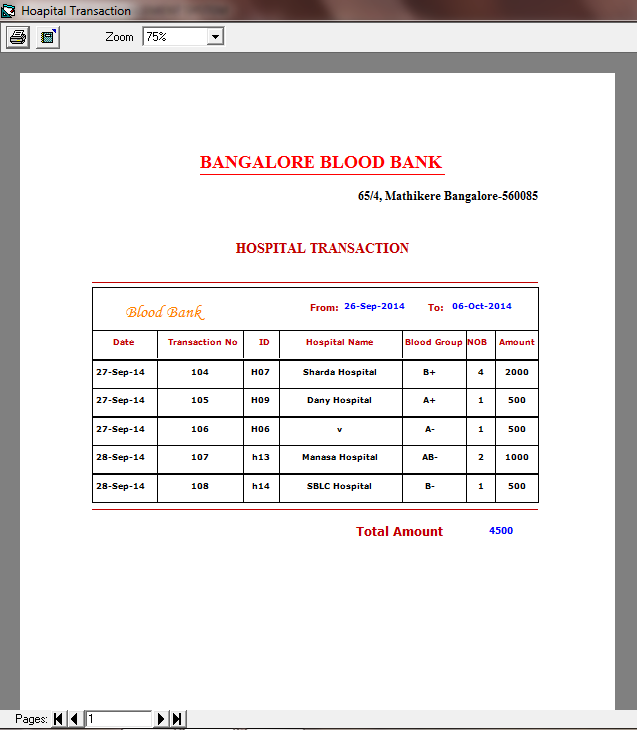
**Transaction Details**

REPORTS

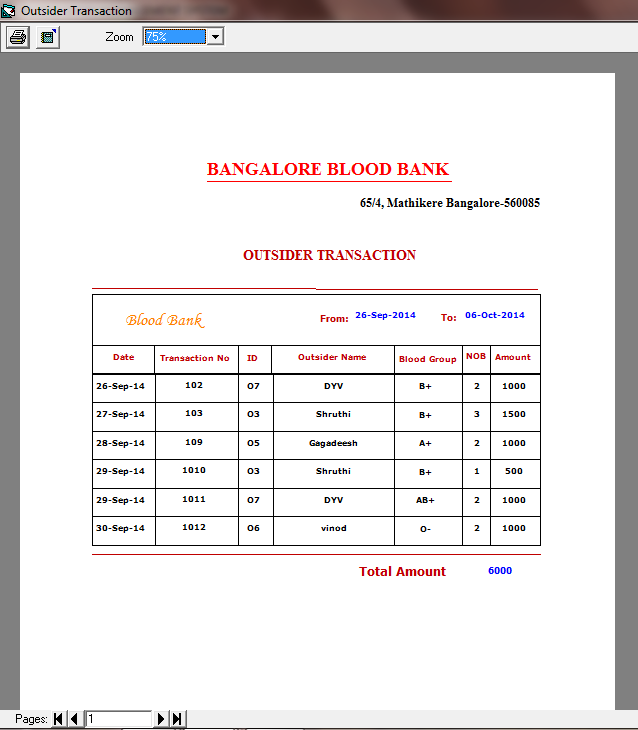
**Donor Transaction Report:**



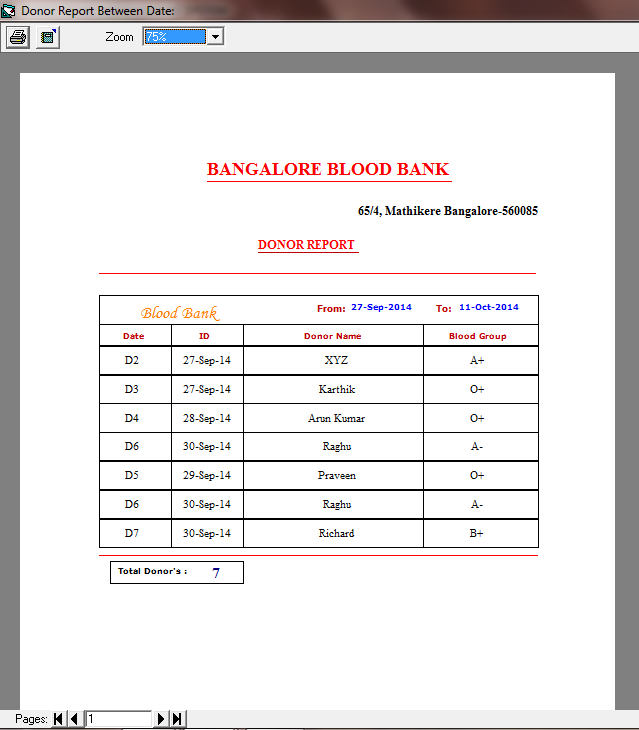
**Hospital Transaction Report:**



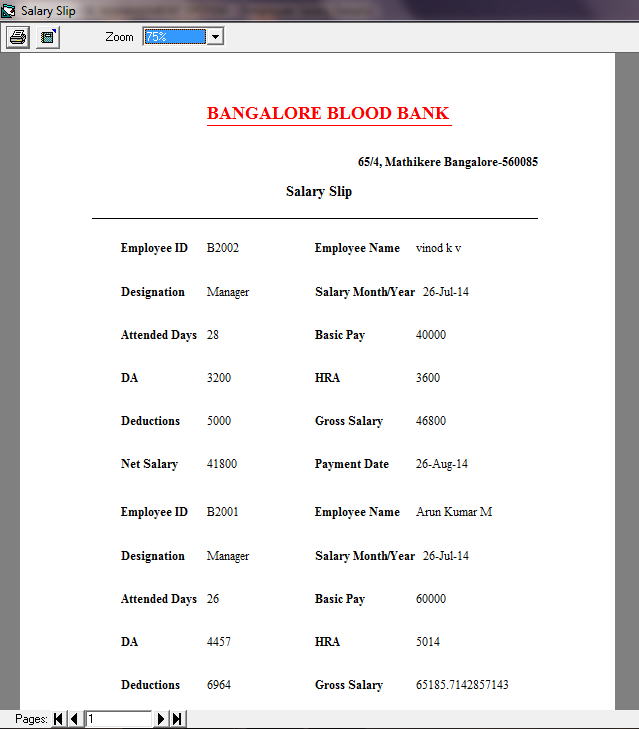
**Outsider Transaction Report:**



**Donor Report**



Employee Salary Report



CODING

Login Form:

Private Sub cmdlogin\_Click()

Dim count

Dim rs As New ADODB.Recordset

Dim sql As String

Dim quit

If txtUsername.Text = "" Then

MsgBox "Fill The User Name!!!", vbInformation, "User Name Missing "

txtUsername.Text = ""

txtUsername.SetFocus

SendKeys "{home}" & "+{end}"

Exit Sub

End If

If txtPassword.Text = "" Then

MsgBox "Please Enter Password!!!", vbInformation, "Password Missing"

txtPassword.Text = ""

txtPassword.SetFocus

Exit Sub

End If

If IsNumeric(Trim(txtUsername.Text)) Then

MsgBox "Please Enter Characters Only", vbInformation

txtUsername.Text = ""

txtUsername.SetFocus

Exit Sub

End If

If Trim(txtUsername.Text) = "" Then

MsgBox "Please Enter Correct Information", vbInformation + vbOKOnly

txtUsername.Text = ""

txtUsername.SetFocus

End If

On Error GoTo e1

Dim strsql As String

strsql = "select \* from Login where Username='" & txtUsername.Text & "' and Password='" & txtPassword.Text & "'"

If dbrec.State = adStateOpen Then

dbrec.Close

End If

Module1.main

dbrec.Open strsql, dbcon, adOpenDynamic, adLockOptimistic

e1:

If err.Number = 3709 Then

ElseIf Not dbrec.EOF Then

USER\_NAME = txtUsername.Text

MsgBox "!!! Login Done Successfully !!! " & vbCrLf & vbCrLf & "User Name:- " & USER\_NAME & vbCrLf & vbCrLf & "Time :-" & Time, vbOKOnly + vbInformation, "Login"

MDIForm1.Show

Unload Me

Else

Call login

End If

dbrec.Close

dbcon.Close

End Sub

Public Function login()

Dim can

can = MsgBox("Wrong Username and Password!!", vbInformation, "login Failed")

If can = vbOK Then

txtUsername.Text = ""

txtPassword.Text = ""

txtUsername.SetFocus

End If

End Function

Private Sub Form\_Load()

Counter = 0

Me.Height = Screen.Height - 400

Me.Width = (Screen.Width)

Randomize

txtPassword.Text = ""

End Sub

Private Sub txtusername\_KeyPress(keyascii As Integer)

Select Case keyascii

Case 8

Case 32, 46

Case 65 To 90

Case 97 To 122

Case Else

keyascii = 0

End Select

End Sub

Private Sub txtpassword\_GotFocus()

If txtUsername.Text = "" Then

MsgBox "User Name cannot be Empty", vbOKOnly, "user missing"

txtUsername.SetFocus

End If

End Sub

Private Sub cmdcancel\_Click()

Dim can

can = MsgBox("ARE YOU SURE YOU WANT TO CANCEL", vbYesNo + vbQuestion, "Cancelation")

If can = vbYes Then

Unload Me

End

Else

txtUsername.SetFocus

End If

End Sub

MDI Form

Private Sub mnubloodbagsearch\_Click()

BloodBagSearch.Show

End Sub

Private Sub mnucampschedule\_Click()

CampScheduleDetails.Show

End Sub

Private Sub mnuchangepassword\_Click()

ChangePassword.Show

End Sub

Private Sub mnuclose\_Click()

Unload Me

End Sub

Private Sub mnucontacts\_Click()

Contacts.Show

End Sub

Private Sub mnudeleteuser\_Click()

Form1.Show

End Sub

Private Sub mnudonordetails\_Click()

donorinformation.Show

End Sub

Private Sub mnudonorreport\_Click()

DonorList.Show

End Sub

Private Sub mnudonorsearch\_Click()

DonorSearch.Show

End Sub

Private Sub mnudonortransaction\_Click()

DonorTrans.Show

End Sub

Private Sub mnuemployeedetails\_Click()

EmployeeDetails.Show

End Sub

Private Sub mnuequipmentdetails\_Click()

EquipmentDetails.Show

End Sub

Private Sub mnuhospitaldetails\_Click()

HospitalInfo.Show

End Sub

Private Sub mnuhospitalsearch\_Click()

HospitalSearch.Show

End Sub

Private Sub mnuhospitaltransaction\_Click()

HospitalTransaction.Show

End Sub

Private Sub mnulogout\_Click()

login.Show

Unload Me

End Sub

Private Sub mnunewuser\_Click()

NewUser.Show

End Sub

Private Sub mnuoutsiderdetails\_Click()

OutsiderInfo.Show

End Sub

Private Sub mnuoutsidersearch\_Click()

OutsiderSearch.Show

End Sub

Private Sub mnuoutsidertransaction\_Click()

Outsidertransaction.Show

End Sub

Private Sub mnusalarydetails\_Click()

EmployeeSalary.Show

End Sub

Private Sub mnustockdetails\_Click()

StockDetails.Show

End Sub

Private Sub mnutransactiondetails\_Click()

TransactionDetails.Show

End Sub

Private Sub mnutransactionlist\_Click()

TransactionList.Show

End Sub

Donor Transaction

Private Sub cmdPrint\_Click()

If dtpFrom.Value > dtpTo.Value Then

MsgBox "Please check Selected Dates", vbInformation + vbOKOnly

Exit Sub

End If

If dtpTo.Value > Date Then

MsgBox "select 2nd Date as Todays Date"

Exit Sub

End If

DonorTransReport

End Sub

Function DonorTransReport()

Dim rs As New ADODB.Recordset

Dim strsql As String

Module1.Connect

rs.Open "select \* from Trans where Recipient\_Type=1", cnn, adOpenDynamic, adLockOptimistic

If rs.RecordCount < 1 Then

MsgBox "Record not present", vbInformation

Exit Function

Else

Set DRDonorTrans.DataSource = rs

With DRDonorTrans

.Sections("Section2").Controls.Item("lblFrom").Caption = Format(dtpFrom.Value, "dd-MMM-yyyy")

.Sections("section2").Controls.Item("lblTo").Caption = Format(dtpTo.Value, "dd-MMM-yyyy")

End With

With DRDonorTrans.Sections("Section1").Controls

.Item("txtDate").DataField = "Date"

.Item("txtTransNo").DataField = "Transaction\_ID"

.Item("txtID").DataField = "Recipient\_ID"

.Item("txtDonorName").DataField = "Recipient\_Name"

.Item("txtBloodGroup").DataField = "BloodGroup"

.Item("txtNOB").DataField = "NoOfBags"

End With

DRDonorTrans.Refresh

DRDonorTrans.Show

End If

rs.Close

End Function

Private Sub Form\_Load()

dtpTo.Value = Date

End Sub

Hospital Transaction

Private Sub cmdPrint\_Click()

If dtpFrom.Value > dtpTo.Value Then

MsgBox "Please check Selected Dates", vbInformation + vbOKOnly

Exit Sub

End If

If dtpTo.Value > Date Then

MsgBox "select 2nd Date as Todays Date"

Exit Sub

End If

HospitalTransReport

End Sub

Function HospitalTransReport()

Dim rs As New ADODB.Recordset

Dim strsql As String

Module1.Connect

rs.Open "select \* from Trans where Recipient\_Type=2", cnn, adOpenDynamic, adLockOptimistic

If rs.RecordCount < 1 Then

MsgBox "Record not present", vbInformation

Exit Function

Else

Set DRHospitalTrans.DataSource = rs

With DRHospitalTrans

.Sections("Section2").Controls.Item("lblFrom").Caption = Format(dtpFrom.Value, "dd-MMM-yyyy")

.Sections("section2").Controls.Item("lblTo").Caption = Format(dtpTo.Value, "dd-MMM-yyyy")

End With

With DRHospitalTrans.Sections("Section1").Controls

.Item("txtDate").DataField = "Date"

.Item("txtTransNo").DataField = "Transaction\_ID"

.Item("txtID").DataField = "Recipient\_ID"

.Item("txtHospitalName").DataField = "Recipient\_Name"

.Item("txtBloodGroup").DataField = "BloodGroup"

.Item("txtNOB").DataField = "NoOfBags"

.Item("txtAmount").DataField = "Amount"

End With

DRHospitalTrans.Sections("Section5").Controls.Item("funTotal").DataField = "Amount"

DRHospitalTrans.Refresh

DRHospitalTrans.Show

rs.Close

End If

End Function

Private Sub Form\_Load()

dtpTo.Value = Date

End Sub

Blood bag from Donor

Private Sub cmdClose\_Click()

Unload Me

End Sub

Private Sub cmdSave\_Click()

Dim dm As Integer

Dim m As Integer

Dim strsql As String

dm = Format(dtpcollectiondate.Value, "mm")

m = Format(donorinformation.dtplastdate.Value, "mm")

dtpexpirydate.Value = dtpcollectiondate.Value + 43 ' the Expiry date is 43 days after the Collection date

strsql = "update DonorDetails set LastDonateDate='" & dtpcollectiondate.Value & "' where Donor\_ID='" & txtDID.Text & "'"

cnn.Execute strsql

strsql = "insert into BloodBagDetails (BloodBag\_ID,Donor\_ID,BloodGroup,Status,Collection\_Date,Expiry\_Date,EnteredBy) values ('" & txtbloodbagid.Text & "','" & txtDID.Text & "','" & txtbloodgp.Text & "',' Existing ','" & dtpcollectiondate.Value & "','" & dtpexpirydate.Value & "','" & txtenteredby.Text & "')"

cnn.Execute strsql

MsgBox "Record Saved", vbInformation, "Donor Blood Donation"

strsql = "update StockDetails set NoOfBags\_Available=NoOfBags\_Available+" & 1 & " where BloodGroup='" & txtbloodgp.Text & "'"

cnn.Execute strsql

Set dbcon = Nothing

flexgrid

End Sub

Private Sub txtDID\_KeyPress(keyascii As Integer)

Select Case keyascii

Case 8

Case 65 To 90

Case 97 To 122

Case 48 To 57

Case Else

keyascii = 0

End Select

End Sub

Private Sub txtEnteredBy\_KeyPress(keyascii As Integer)

Select Case keyascii

Case 8

Case 32, 46

Case 65 To 90

Case 97 To 122

Case Else

keyascii = 0

End Select

End Sub

Private Sub flexgrid()

Dim strsql As String

Dim srno As Integer

Module1.Connect

'Checks for the Blood Bag ID

dbrec.Open "SELECT \* FROM BloodBagDetails WHERE Donor\_ID='" & txtDID.Text & "'", cnn, adOpenKeyset, adLockOptimistic

srno = 1

MSFGDonate.Clear

MSFGDonate.FormatString = "SNo | BloodBagId | Donation Date "

MSFGDonate.ColWidth(1) = 2300

MSFGDonate.ColWidth(2) = 2300

'Assigns the values from the database to the corresponding controls.

While Not dbrec.EOF

MSFGDonate.TextMatrix(srno, 0) = srno

MSFGDonate.TextMatrix(srno, 1) = dbrec("BloodBag\_ID") & ""

MSFGDonate.TextMatrix(srno, 2) = dbrec("Collection\_Date") & ""

dbrec.MoveNext

srno = srno + 1

If MSFGDonate.Rows = srno Then MSFGDonate.Rows = MSFGDonate.Rows + 1

Wend

dbrec.Close

End Sub

Private Sub cmdcanceldonate\_Click()

Call Clear(Me)

End Sub

**Camp Schedule**

Private Sub cmdClose\_Click()

Dim response As String

response = MsgBox("Do you want to Quit ?", vbQuestion + vbYesNo, "DONOR DETIALS")

Select Case response ' Provides the MsgBox response as the expression to match for the Case Structure.

Case vbYes:

Unload Me ' Unload the current form

Case vbNo:

Exit Sub

End Select

End Sub

Private Sub cmdCSearch\_Click()

Dim rs As New ADODB.Recordset

Module1.Connect

If optCode.Value = True Then

If cmbCode.Text = "" Then

MsgBox "Select Camp Code", vbInformation, "Camp Code"

Exit Sub

Else

rs.Open "select \* from CampSchedule where Code='" & cmbCode.Text & "'", cnn, adOpenStatic, adLockOptimistic

End If

ElseIf optName.Value = True Then

If cmbName.Text = "" Then

MsgBox "Select Camp Name", vbInformation, "Camp Name"

Exit Sub

Else

rs.Open "select \* from CampSchedule where Name='" & cmbName.Text & "'", cnn, adOpenStatic, adLockOptimistic

End If

End If

If rs.BOF And rs.EOF Then

MsgBox "No Records Found for this Criteria", vbInformation, "Camp Search"

Exit Sub

End If

txtCode.Text = rs("Code")

txtName.Text = rs("Name")

txtAdd1.Text = rs("Address1")

txtAdd2.Text = rs("Address2")

txtAdd3.Text = rs("Address3")

txtPinCode.Text = rs("PinCode")

txtPerson.Text = rs("Contact\_Person")

txtPhNo.Text = rs("Phone\_No")

dtpDate.Value = rs("Scheduled\_Date")

txtBeds.Text = rs("No\_Of\_Beds")

rs.Close

FrameSearch.Visible = False

End Sub

Private Sub cmdDelete\_Click()

Dim sql As String

Dim response As String

If txtCode.Text = "" Then

MsgBox "Enter Camp Schedule Code to be Deleted", vbInformation, "Delete Camp Schedule"

Exit Sub

End If

response = MsgBox("Do you want delete Camp Schehule Code '" & txtCode.Text & "'", vbQuestion + vbYesNo, "Delete Camp Schedule")

Select Case response

Case vbYes: sql = "delete from CampSchedule where Code='" & txtCode.Text & "'"

cnn.Execute sql

Call Clear(Me)

ListCampSchedule

MsgBox "Camp Schehule Code '" & txtCode.Text & "' Deleted from Record", vbInformation, "Delete Camp Schedule"

Case vbNo: Exit Sub

End Select

End Sub

Private Sub cmdModify\_Click()

If ValidateIndividual = False Then Exit Sub

Dim sql As String

sql = "update CampSchedule set Name='" & txtName.Text & "',Address1='" & txtAdd1.Text & "',Address2='" & txtAdd2.Text & "',Address3='" & txtAdd3.Text & "',PinCode='" & txtPinCode.Text \_

& "',Contact\_Person='" & txtPerson.Text & "',Phone\_No='" & txtPhNo.Text & "',Scheduled\_Date='" & dtpDate.Value & "',No\_Of\_Beds='" & txtBeds.Text & "' where Code='" & txtCode.Text & "'"

cnn.Execute sql

ListCampSchedule

MsgBox "Camp schedule Record Modified", vbInformation, "Camp Schedule"

End Sub

Private Sub cmdNew\_Click()

Call Clear(Me)

Dim n As Integer

Dim nn As String

Dim rs As New ADODB.Recordset

Module1.Connect

rs.Open "select \* from CampSchedule order by Code", cnn, adOpenStatic, adLockOptimistic

n = rs.RecordCount + 1

nn = "CS500" + Trim(n)

For i = 1 To rs.RecordCount

If rs("Code") <> "CS500" + Trim(i) Then

n = i

nn = "CS500" + Trim(n)

Exit For

End If

rs.MoveNext

Next

txtCode.Text = nn

rs.Close

cmdNew.Enabled = False

cmdSave.Enabled = True

cmdScancel.Enabled = True

cmdModify.Enabled = False

cmdSearch.Enabled = False

cmdDelete.Enabled = False

cmdClear.Enabled = False

End Sub

Private Sub cmdSave\_Click()

If ValidateIndividual = False Then Exit Sub

Dim sql As String

sql = "insert into CampSchedule(Code,Name,Address1,Address2,Address3,PinCode,Contact\_Person,Phone\_No,Scheduled\_Date,No\_Of\_Beds) values ('" & txtCode.Text & "','" & txtName.Text \_

& "','" & txtAdd1.Text & "','" & txtAdd2.Text & "','" & txtAdd3.Text & "','" & txtPinCode.Text & "','" & txtPerson.Text & "','" & txtPhNo.Text & "','" & dtpDate.Value & "','" & txtBeds.Text & "')"

cnn.Execute sql

ListCampSchedule

MsgBox "Camp schedule Record Saved", vbInformation, "Camp Schedule"

cmdNew.Enabled = True

cmdSave.Enabled = False

cmdScancel.Enabled = False

cmdModify.Enabled = True

cmdSearch.Enabled = True

cmdDelete.Enabled = True

cmdClear.Enabled = True

End Sub

Private Function ListCampSchedule()

Dim rs As New ADODB.Recordset

Dim slno As Integer

Module1.Connect

rs.Open "select \* from CampSchedule", cnn, adOpenStatic, adLockOptimistic

msfCamp.Clear

msfCamp.FormatString = "CampSchedule Code | CampSchedule Name | Address1 | Address2 | Address3 | PinCode | Contact Person | Phone No | Scheduled Date | No Of Beds"

msfCamp.ColWidth(7) = 1500

slno = 1

For i = 1 To rs.RecordCount

msfCamp.TextMatrix(slno, 0) = rs("Code")

msfCamp.TextMatrix(slno, 1) = rs("Name")

msfCamp.TextMatrix(slno, 2) = rs("Address1")

msfCamp.TextMatrix(slno, 3) = rs("Address2")

msfCamp.TextMatrix(slno, 4) = rs("Address3")

msfCamp.TextMatrix(slno, 5) = rs("PinCode")

msfCamp.TextMatrix(slno, 6) = rs("Contact\_Person")

msfCamp.TextMatrix(slno, 7) = rs("Phone\_No")

msfCamp.TextMatrix(slno, 8) = rs("Scheduled\_Date")

msfCamp.TextMatrix(slno, 9) = rs("No\_Of\_Beds")

slno = slno + 1

If msfCamp.Rows = slno Then

msfCamp.Rows = msfCamp.Rows + 1

End If

rs.MoveNext

Next

rs.Close

End Function

Private Sub cmdScancel\_Click()

Call Clear(Me)

cmdNew.Enabled = True

cmdSave.Enabled = False

cmdScancel.Enabled = False

cmdModify.Enabled = True

cmdSearch.Enabled = True

cmdDelete.Enabled = True

cmdClear.Enabled = True

End Sub

Private Sub cmdSearch\_Click()

FrameSearch.Visible = True

End Sub

Private Sub cmdClear\_Click()

Call Clear(Me)

End Sub

Private Sub Form\_Load()

Call Clear(Me)

ListCampSchedule

FillCombo

End Sub

Private Sub msfCamp\_DblClick()

Dim rs As New ADODB.Recordset

Module1.Connect

rs.Open "select \* from CampSchedule where Code='" & msfCamp.TextMatrix(msfCamp.Row, 0) & "'", cnn, adOpenStatic, adLockOptimistic

txtCode.Text = rs("Code")

txtName.Text = rs("Name")

txtAdd1.Text = rs("Address1")

txtAdd2.Text = rs("Address2")

txtAdd3.Text = rs("Address3")

txtPinCode.Text = rs("PinCode")

txtPerson.Text = rs("Contact\_Person")

txtPhNo.Text = rs("Phone\_No")

dtpDate.Value = rs("Scheduled\_Date")

txtBeds.Text = rs("No\_Of\_Beds")

rs.Close

End Sub

Private Sub optCode\_Click()

optName.Value = False

cmbCode.Text = ""

cmbName.Text = ""

cmbCode.Enabled = True

cmbName.Enabled = False

End Sub

Private Sub optName\_Click()

optCode.Value = False

cmbCode.Text = ""

cmbName.Text = ""

cmbCode.Enabled = False

cmbName.Enabled = True

End Sub

Private Function FillCombo()

Dim rs As New ADODB.Recordset

Module1.Connect

rs.Open "select \* from CampSchedule", cnn, adOpenStatic, adLockOptimistic

For i = 1 To rs.RecordCount

cmbCode.AddItem rs("Code")

cmbName.AddItem rs("Name")

rs.MoveNext

Next

rs.Close

End Function

Private Function ValidateIndividual() As Boolean

If txtCode.Text = "" Then

MsgBox "Enter Camp Schedule Code", vbInformation, "Camp Schedule Code"

txtCode.SetFocus

ValidateIndividual = False

Exit Function

End If

If txtName.Text = "" Then

MsgBox "Enter Camp Schedule Name", vbInformation, "Camp Schedule Name"

txtName.SetFocus

ValidateIndividual = False

Exit Function

End If

If txtAdd1.Text = "" Then

MsgBox "Enter Adress", vbInformation, "Address"

txtAdd1.SetFocus

ValidateIndividual = False

Exit Function

End If

If txtPinCode.Text = "" Then

MsgBox "Enter PinCode", vbInformation, "PinCode"

txtPinCode.SetFocus

ValidateIndividual = False

Exit Function

End If

If txtPerson.Text = "" Then

MsgBox "Enter Contact Person Name", vbInformation, "Contact Person"

txtPerson.SetFocus

ValidateIndividual = False

Exit Function

End If

If txtPhNo.Text = "" Then

MsgBox "Enter Phone Number", vbInformation, "Phone Number"

txtPhNo.SetFocus

ValidateIndividual = False

Exit Function

End If

If txtBeds.Text = "" Then

MsgBox "Enter No of Beds Required", vbInformation, "No Of Beds"

txtBeds.SetFocus

ValidateIndividual = False

Exit Function

End If

ValidateIndividual = True

End Function

Private Sub txtAdd1\_KeyPress(keyascii As Integer)

Select Case keyascii

Case 8

Case 32

Case 65 To 90, 97 To 122, 44 To 57

Case Else

keyascii = 0

End Select

End Sub

Private Sub txtAdd2\_KeyPress(keyascii As Integer)

Select Case keyascii

Case 8

Case 32

Case 65 To 90, 97 To 122, 44 To 57

Case Else

keyascii = 0

End Select

End Sub

**SOFTWARE TESTING**

As the coding is completed according to the requirements we have to test the quality of the software. Software testing is a critical element of the software quality assurance and represents the ultimate review of specification, design and coding. Although testing is to uncover the errors in the software but is also demonstrates the software functions appear to be working as per the specifications, those performance requirements appear to have been met, in addition, data collected as testing is conducted provide a good indication of software reliability and some indications of software quality as a whole. To assure the software quality we conduct both white box testing and black box testing.

**White box Testing:**

White box testing is a test case design method that uses the control structure of the procedural designs to derive test cases. As we are using a non procedural language, there is very small scope for the white box testing. Whenever it is necessary, there the control structures are tested and successfully passed all the control structures with a very minimum error.

**Black Box Testing:**

Black box testing focuses on the functional requirements of the software. It enables to derive sets of input conditions that will fully exercise all functional requirements for a program. The black box testing finds almost all errors, if finds some interface errors and errors in accessing the database and some performance errors in b lack box testing we use mainly two techniques: equivalence partitioning the boundary volume analysis technique.

* Equivalence partitioning:

In this method we divide input domain of program into classes of data from which test cases are derived. An Equivalence class represents a set of valid or invalid or a set of related values or a Boolean conditions. The equivalence for these is:

Input condition requires specific, value-specific or non-specific two classes.

* Input condition requires a range – in the range or out of range two classes
* Input condition specifies a member of set-belongs to a set or not belongs to the set of two classes.
* Input condition id Boolean- valid or invalid Boolean condition two classes.

By these types of equivalent classes, we can test for many cases.

* Boundary Values Analysis:

Number of errors usually occurs at the boundaries of the input domain generally. In this technique a selection of test cases is exercised using boundary values i.e. around boundaries.

In our project **Blood bank Management System**, by the above two techniques, we eliminated almost all errors from the software and checked for numerous test values for each and every input value. The results were satisfactory.

**System Testing:**  
 System testing is designated to uncover weakness that was not detected in the earlier tests. The total system is tested for recovery and fallback after various major failures to ensure that no data are lost. An acceptance test is done for the validity and reliability of the system. The philosophy behind the testing is to find error in project. There are many test cases designed with this in mind.  
  
In our project **BLOOD BANK MANAGEMENT SYSTEM**, we have tested in many ways possible. The whole system has been tested according to different types of testing such as unit testing, combined module testing, stress testing, volume testing, user acceptance test, integration testing.

The flow of testing is as follows:

**Code Testing:**

Specification testing is done to check if the program does with it should do and how it should behave under various conditions or combinations and submitted for processing in the system and it is checked if any overlaps occur during the processing.

This strategy examines the logic of the program. Here only syntax of the code is tested. In code testing syntax error are corrected, to ensure that the code is perfect.

In our project, logic is complex but a very interesting one. The syntax and the statements are all tested line by line by line for error correction. During compilation each line has to be checked for errors. This is done by code testing.

**Unit Testing:**

The first level of testing is called unit testing. Here different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test oracles.

Unit testing comprises a set of tests performed by an individual programmer prior to integration of the units into a large system. A system unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first the modules to locate errors. These errors are verified and corrected so that the unit perfectly fits to the project.

In our project **BLOOD BANK MANAGEMENT SYSTEM**, the code is divided into many units. We have used divide and conquer technique. Each unit is separated and tested according to the user comfort. There is a line separating after each part of the coding. It is easier to correct the errors.

System Testing:  
 The next level of testing is system testing and acceptance testing. This testing is done to check if the system has met its requirements and to find the external behavior of the system.

System testing involves two kinds of activities:

* Integration testing
* Acceptance testing

Integration Testing:  
 The next level of testing is called the Integration Testing. In this many tested modules are combined into subsystems, which were then tested.

Test case data is prepared to check the control flow of all the modules and to exhaust all possible inputs to the program. Situations like treating the modules when there is no data entered in the test box is also tested.

This testing strategy dictates the order in which modules must be available, and exerts strong influence on the order in which the modules must be written. Debugged and unit tested. In Integration testing, all the modules/units on which unit testing is performed are integrated together and tested.

In any project, there are different modules or subsystems. These subsystems are tested and combined together or integrated together to form a bigger module or sometimes a whole system. This is done by integration testing.

In our project, even though the coding is divided into many parts. At the end, it is all integrated into one big module and tested.

**Acceptance Testing:**  
 This testing is performed finally by user to demonstrate that the implemented system satisfies its requirements. The user gives various inputs to get required outputs.  
 In our project, we have designed it in a way where any form of people can understand and use it.

* It is user friendly
* It can be implemented everywhere
* It can hold any number of records
* The flow of the project is easy and understanding

Specification Testing:  
 Specification testing is done to check if the program does what it should do and how it should behave under various conditions or combination and submitted for processing in the system and it is checked if any overlaps occur during the processing.

In our project, the main aim is to specify the needs of maintaining the data base correctly and efficiently. This specification is done by testing it according to various conditions.

Performance Time Testing:

Performance time testing is done to determine how long it takes to accept and respond i.e., the total time of processing when it has to handle quite a large number of records. It is essential to check the exception speed of the system, which runs well with only a handful of test transactions. Such system might be slow when a fully loaded. So testing is done by providing large number of data for processing. A system testing is designed to uncover weakness that was not detected in the earlier tests.

The total system is tested for recovery and fallback after various major failures to ensure that no data are lost during an emergency. An acceptance test is done to the user about the validity and reliability of the system.

In our project, we have assigned specific time limit to certain parts, which is very much important. The time taken to load and unload the forms and progress bar and time interval everything must be fast and within the time limit. Performance time testing takes care of all this.

IMPLEMENTATION

The final and important phase in system life cycle is the installation of the new system. The term installation has different meaning, ranging from the conversion of basic application a complete replacement of a computer system. The procedure, however are virtually the same. Installation is the process of converting a new system into an operational one. It involves user training and successful running of the developed system.

In the installation phase the user and the system manuals are prepared handed over to the user to operate the developed system. The objective of each manual is the same but the target customers may be different.

The user manual is aimed at those who would be using the system but not interested in technical information about the insides of the system, or implementation of login within the system.

**The information that they are interested are-**

* Scope of the system
* Flow and links across the programs and modules
* Validation done within the program
* The actual usage of the system

**The contents of the user manual are the following:**

* Menu map and Disk usage
* Usage description
* Back up procedures
* Security measures
* Input’s and output’s formats
* Do’s and don’ts

The system manual is aimed at those persons who would ne maintain, changing and enhancing system and also the people who are interested in the implementation of the logic within the system.

FUTURE ENHANCEMENTS

The system has been developed under the given conditions and is found to work effectively and efficiently. We should make the possible use of the techniques. The basic purpose of all the innovations in the technology is for our betterment.

The system has been tested with sample covering .All possible options and their performances are good. Since this system is flexible and modular, further modifications of this package can be easily incorporated. The system can be easily placed on the office so that best work efficiently.

CONCLUSION

In this project, **“BLOOD BANK MANAGEMENT SYSTEM”** we have tried to computerize various processes of Blood Bank.

Blood Bank Management System is very flexible software and can be used in any branch of BLOOD BANK for keeping record. In this software we have tried to provide all the Blood bank management system related record keeping facilities which helps to keep record and employees who belongs to it.

The main focus of this project is to less in human efforts. The maintenance of the record is made efficient, as all the records are stored in the SQL database.

It is user interactive and effective than the existing system. The flexibility of visual basic helps to maintain the **“BLOOD BANK MANAGEMENT SYSTEM”** more efficiently.

Finally, we are thankful to all the people who have given us their hearty support in this endeavour.

BIBILOGRAPHY

The following are the books that have been referred for the successful completion of our project work:

**Complete Reference** – 5th Edition by Herbert Schildt, published by

McGraw-Hill Publishing Company Limited.

**Software Engineering** – A Practitioner’s Approach, Fifth Edition by

Roger S Pressman, Published by McGraw – Hill International Edition.

**Visual Basic 6 Programming** – By Steven Holzner, published by dream tech press.

**Database** – By Elmasri Navathe