CHRIST ACADEMY
INSTITUTE FOR ADVANCED STUDIES

Christ Nagar, Hullahalli, Begur - Koppa Road, Sakkalwara Post, Bengaluru-560083

**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

A PROJECT REPORT ON

# "BLOOD BANK MANAGEMENT SYSTEM"

*Submitted in the partial fulfilment for the award of degree in*

**BACHELOR OF COMPUTER APPLICATIONS**

Submitted by

**VIGNESH S PILLAI**
**(199USB7048)**

Under the guidance of

**DR. SANGEETHA GOVINDA**
**HEAD OF THE DEPARTMENT**
**DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS**

**MARCH - 2022**

# BLOOD BANK MANAGEMENT SYSTEM

# 1. INTRODUCTION

## 1.1 Project Description

The “Blood Bank Management System” has been developed to overcome the problems of maintaining manual records. This application helps the staff to manage records of donors and recipient and also update blood stock digitally.

The project is totally built at administrative end and thus only the admin or Blood bank staff is guaranteed the access to the application.

The application works as much as possible to avoid errors while entering data. It also shows error message while entering invalid data. User interface is designed in such a way that it is easily understandable to the user. “Blood Bank Management System” is an error free, secure, reliable and fast management system.

## 1.2 Problem Statement

Traditional Blood Banks are still using the old method of hand-written records. This system would help in automating the system of Blood banks and to overcome the load of the manual system by digitally maintaining records of Donor, Recipient and stock.

## 1.3 Existing System

The existing system which is currently being used in blood banks follows a manual way of maintaining records which is paper-based where the staff manually maintains details regarding Donor, Stock and Recipients.

## 1.4 Drawbacks of Existing System

- Takes a lot of time
- Lack of Security
- Lack of Accuracy

## 1.5 Proposed System

To overcome the drawbacks of existing system, the proposed system is developed. This application helps us to maintain Donor, Stock and Recipient details in the forms of Digital records. “Blood Bank Management System” is an error free, secure, reliable and fast management system.

## 1.6 Objectives

The main objectives of Blood Bank management System are:

- To manage details of Donor, Recipient, Blood Stock and Staff details.
- To Reduce time consumption and increase efficiency.
- Makes it easier to keep track of the blood records and stock.
- Minimal possibility for errors.
- All the data related to blood bank will be recorded and can be verified at any time.

## 1.7 Benefits of Proposed System

- User-friendly and easy to handle.
- Chances of errors are decreased.
- Proper information is delivered and in a faster way.
- Stored data can be easily edited or updated.
- Data is more secure.

## 1.8 Modules

- ADMIN MODULE: Admin has a username and password. The admin module gives an admin or a staff, the access to the application and gives the rights to add, update or delete records.

- DONOR MODULE: In this module, the donor's details like Donor ID, Name, Blood Group, Contact can be added, updated or deleted.

- STOCK MODULE: This module provides the fundamental concepts required for managing blood stock. This includes keeping track of how much blood stock is available in each blood bank.

- RECIPIENT MODULE: In this module, the recipient's details like Recipient ID, Name, Blood Group required, Contact can be added.

# 2. SYSTEM REQUIREMENTS

## 2.1 Hardware Specifications

- PROCESSOR : AMD Ryzen 5 5600h with Radeon Graphics
- SPEED : 3.00 Ghz
- RAM : 8 GB
- HDD : 500 GB
- MONITOR : HP Pavillion

## 2.2 Software Specifications

- OPERATING SYSTEM : Windows 11 64bit
- FRONT END : Visual Basic 6.0
- BACK END : MySQL
- LANGUAGE USED : Visual Basic
- DOCUMENTATION : Microsoft Word

# 3. TOOLS

## 3.1 Back End - SQL

SQL (Structured Query Language) is a standardized programming language that's used to manage relational databases and perform various operations on the data in them. It is also known as SQL databases; relational systems comprise a set of tables containing data in rows and columns. Initially created in the 1970s, SQL is regularly used not only by database administrators, but also by developers writing data integration scripts and data analysts looking to set up and run analytical queries.

The use of SQL includes modifying database table and index structures; adding, updating and deleting rows of data; and retrieving subsets of information from within a database for transaction processing and analytics applications. Queries and other SQL operations take the form of commands written as statements -- commonly used SQL statements include select, add, insert, update, delete, create, alter and truncate.

### 3.1.1 Need of SQL

- It is widely used in the Business Intelligence tool.
- Data Manipulation and data testing are done through SQL.
- Data Science tools depend highly on SQL. Big data tools such as Spark, Impala are dependent on SQL.
- It is one of the demanding industrial skills.

### 3.1.2 Advantages of SQL

SQL has many advantages which makes it popular and highly demanded. It is a reliable and efficient language used for communicating with the database. Some advantages of SQL are:

- Faster Query Processing

- Large amount of data is retrieved quickly and efficiently. Operations like Insertion, deletion, manipulation of data is also done in almost no time.
- No Coding Skills
- For data retrieval, large number of lines of code is not required. All basic keywords such as SELECT, INSERT INTO, UPDATE, etc are used and also the syntactical rules are not complex in SQL, which makes it a user-friendly language.
- Standardized Language
- Due to documentation and long establishment over years, it provides a uniform platform worldwide to all its users.
- Portable
- It can be used in programs in PCs, server, laptops independent of any platform (Operating System, etc). Also, it can be embedded with other applications as per need/requirement/use.
- Interactive Language
- Easy to learn and understand, answers to complex queries can be received in seconds.
- Multiple data views

### 3.1.3 Disadvantages of SQL

Although SQL has many advantages, still there are a few disadvantages. Various Disadvantages of SQL are as follows:

- Complex Interface
- SQL has a difficult interface that makes few users uncomfortable while dealing with the database.
- Cost
- Some versions are costly and hence, programmers cannot access it.
- Partial Control
- Due to hidden business rules, complete control is not given to the database.

### 3.1.4 Applications of SQL

- SQL is used by developers and DBAs in writing Data Integration Scripts.
- It is used to deal with analytical queries to analyse the data and get instincts from it.
- Retrieving Information
- Modification/Manipulation of data and database table operation such as Insertion, Deletion and Updation.

## 3.2 Front End- Visual Basic

All the controls in the Toolbox except the Pointer are objects in Visual Basic. These objects have associated properties, methods and events.

Real world objects are loaded with properties. For example, a flower is loaded certain colour, shape and fragrance. Similarly programming objects are loaded with properties. A property is a named attribute of a programming object. Properties define the characteristics of an object such as Size, Colour etc. or sometimes the way in which it behaves. For example, a Textbox accepts properties such as Enabled, Font, Multiline, Text, Visible, Width, etc.

- Enables property allows the Textbox to be enabled or disabled at run time depending on the condition set to True or False.
- Font property sets a particular font in the Textbox.
- Multiline property allows the Textbox to accept and display multiple lines at run time.
- Text property of the Textbox control sets a particular text in the control.
- Visible property is used to hide the object at run time.
- Width property sets the Textbox to the desired width at design time.

A method is an action that can be performed on objects. For example, a cat is an object. Its properties might include long white hair, blue eyes, 3 pounds weight etc. A complete definition of cat must only encompass on its looks, but should also include a complete itemization of its ctivities. Therefore, a cat's methods might be move, jump, play, breath etc.

Similarly, in object-oriented programming, a method is a connected or built-in procedure, a block of code that can be invoked to impart some action on a particular object. A method requires an object to provide them with a context. For example, the word Move has no meaning in Visual Basic, but the statement.

### 3.2.1 Event Driven Programming

Visual Basic programs are built around events. Events are various things that can happen in a program. This will become clearer when studied in contrast to procedural programming. In procedural languages, an application is written is executed by checking for the program logically through the program statements, one after another. For a temporary phase, the control may be transferred to some other point in a program. While in an event driven application, the program statements are executed only when a particular event calls a specific part of the code that is assigned to the event.

Let us consider a Textbox control and a few of its associated events to understand the concept of event driven programming. The Textbox control supports various events such as Change, Click, Mouse Move and many more that will be listed in the Properties dropdown list in the code window for the Textbox control. We will look into a few of them as given below.

- The code entered in the Change event fires when there is a change in the contents of the Textbox.
- The Click event fires when the Textbox control is clicked.

# 4. DATA DICTIONARY

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects and other data.

The data dictionary is a crucial component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary

## 4.1 Staff Login Table

| COLUMN NAME | DATA TYPE | LENGTH | KEY |
|---|---|---|---|
| staffID | Varchar | 50 | Primary |
| Sname | Text | 50 | |
| location | Varchar | 50 | |
| contact | Numeric | 10 | |
| username | Varchar | 50 | |
| password | Varchar | 50 | |

Table 1: stafflogin

## 4.2 Donor Table

| COLUMN NAME | DATA TYPE | LENGTH | KEY |
|---|---|---|---|
| donorID | Varchar | 50 | Primary |
| Dname | Text | 50 | |
| Dblood | Varchar | 50 | |
| dgender | Text | 20 | |
| donorage | Numeric | 10 | |
| dcontact | Numeric | 10 | |
| daddress | Varchar | 50 | |
| dondate | Datetime | 10 | |
| lastdondate | Text | 20 | |
| healthdesc | Varchar | 50 | |

Table 2: donor

## 4.3 Blood Stock Table

| COLUMN NAME | DATA TYPE | LENGTH | KEY |
|---|---|---|---|
| Bno | Varchar | 50 | Primary |
| bloodgroup | Varchar | 50 | |
| donorID | Varchar | 50 | |
| Units | Numeric | 10 | |
| bloodbank | Text | 20 | |

Table 3: bloodstock

## 4.4 Recipient Table

| COLUMN NAME | DATA TYPE | LENGTH | KEY |
|---|---|---|---|
| recipientID | Varchar | 50 | Primary |
| rname | Varchar | 50 | |
| rblood | Varchar | 50 | |
| rgender | Text | 20 | |
| rno | Varchar | 50 | |
| rcontact | Numeric | 10 | |
| raddress | Varchar | 50 | |
| issuedate | Datetime | 10 | |

Table 4: recipient

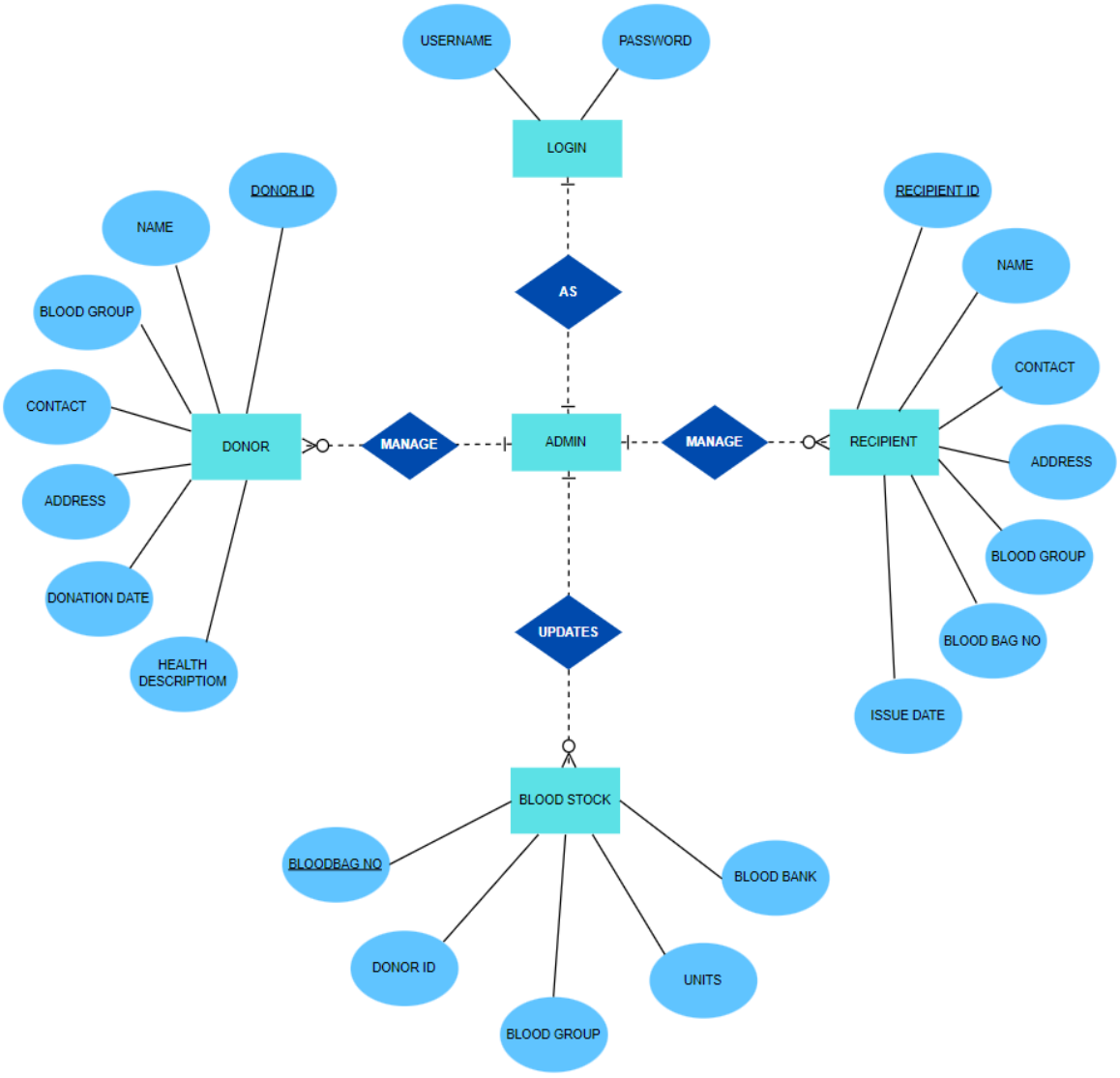
# 5. ENTITY RELATIONSHIP DIAGRAM

A graphical representation of the entities and the relationships between them. Entity relationship diagrams are a useful medium to achieve a common understanding of data among users and application developers.

In data modelling, an entity-relationship model (ERM) is a representation of structured data; entity-relationship modelling is the process of generating these models. The end-product of the modelling process is an entity-relationship diagram (ERD), a type of Conceptual Data Model or Semantic Data Model.

## Components of ER Diagram

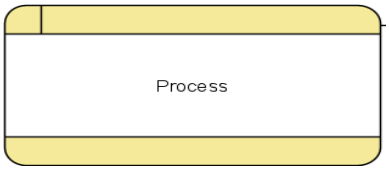
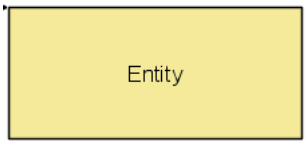
| SYMBOL | NAME | DESCRIPTION |
|---|---|---|
|   | Attribute | An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. |
|   | Derived Attribute | A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram. |
|   | Entity | An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties |
|   | Multivalued Attribute | An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram. |
|   | Relationship | A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities. |


USERNAME
PASSWORD
LOGIN
AS
ADMIN
DONOR ID
NAME
BLOOD GROUP
CONTACT
ADDRESS
DONATION DATE
HEALTH DESCRIPTIOM
DONOR
MANAGE
MANAGE
RECIPIENT ID
NAME
CONTACT
ADDRESS
BLOOD GROUP
BLOOD BAG NO
ISSUE DATE
RECIPIENT
UPDATES
BLOOD STOCK
BLOODBAG NO
DONOR ID
BLOOD GROUP
UNITS
BLOOD BANK

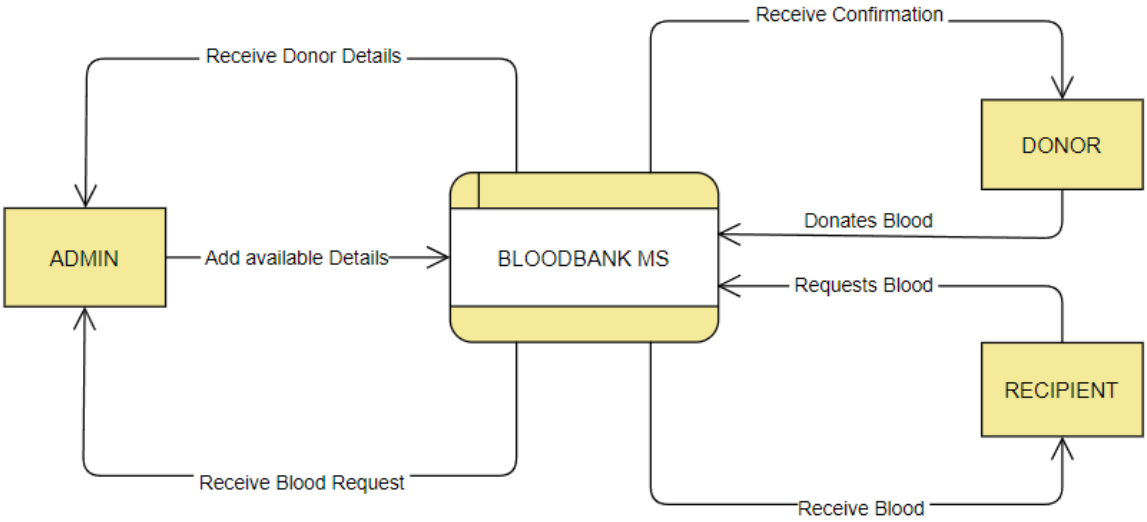
# 6. DATA FLOW DIAGRAM

The data flow diagrams are pictorial or graphical representation of the outline of the system study. The data flow diagram covers all the processes and data storage area which takes place during any transaction in the system. The data flow diagrams are functionally divided into context level, Zero level and First level data flow diagrams.

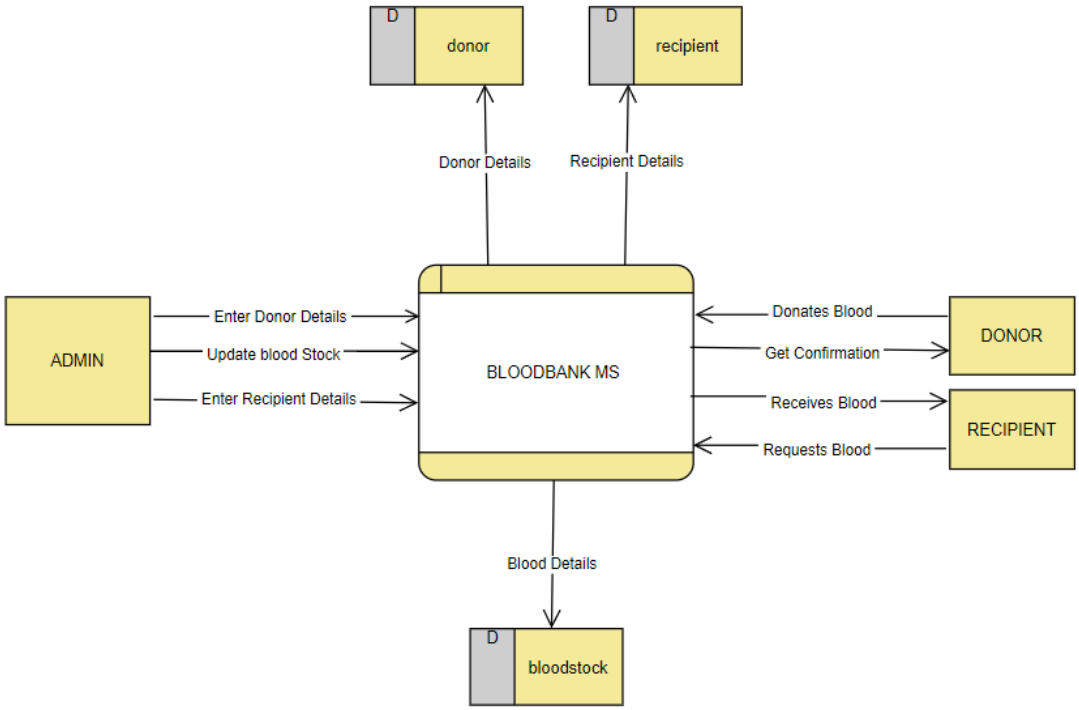
## Symbols Used in DFD

| SYMBOL | NAME | DESCRIPTION |
|---|---|---|
|  | Process | A process shows a transformation or manipulation of data flows within the system. |
|  | External Entity | An external entity is a source or destination of a data flow which is outside the area of study. Only those entities which originate or receive data are represented on a business process diagram. |
|  | Data Store | A data store is a holding place for information within the system. It is represented by an open-ended narrow rectangle. |
|  | Data Flow | A data flow shows the flow of information from its source to its destination. A data flow is represented by line, with arrowheads showing direction of flow. |

## LEVEL-0 DFD




## LEVEL-1 DFD



# 7. SOURCE CODE

## WELCOME PAGE

```
Private Sub Form_Load()

Timer1.Enabled = True

End Sub


Private Sub Timer1_Timer()

ProgressBar1.Value = ProgressBar1.Value + 5

If (ProgressBar1.Value = ProgressBar1.Max) Then

Timer1.Enabled = False

Unload Me

Form2.Show

End If

End Sub
```

## STAFF LOGIN FORM

```
Private Sub cmdexit_Click()

End

End Sub


Private Sub cmdlogin_Click()
```

```
Adodc1.Refresh
On Error GoTo errmsg
With Adodc1.Recordset
.MoveFirst
.Find "username='" & txtuname.Text & "'"
If .EOF = True Then
MsgBox "INCORRECT USERNAME!!!"
ElseIf (txtpwd.Text = Adodc1.Recordset.Fields("password")) Then
MsgBox "YOU ARE SUCCESSFULLY LOGGED IN!!!"
Form4.Show
Else
MsgBox "INCORRECT PASSWORD"
txtuname.Text = ""
txtpwd.Text = ""
End If
End With
Exit Sub
errmsg:
MsgBox Err.Description
End Sub


Private Sub cmdregister_Click()
Form3.Show
```

```
Unload Me

End Sub
```

## NEW USER REGISTRATION FORM

```
Private Sub cmdregister_Click()

Dim id As Integer

Dim id1 As String

txtsid.Enabled = True

txtname.Enabled = True

txtloc.Enabled = True

txtcont.Enabled = True

txtuname.Enabled = True

txtpwd.Enabled = True

txtsid = ""

txtname = ""

txtloc = ""

txtcont = ""

txtuname = ""

txtpwd = ""

On Error GoTo errmsg

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

txtsid = "S101"
```

```
Else
Adodc1.Recordset.MoveLast
id1 = Adodc1.Recordset("staffID")
id = Mid(id1, 2, 4) + 1
txtsid = "S" & id
End If
txtname.SetFocus
Adodc1.Recordset.AddNew
Exit Sub
errmsg:
MsgBox Err.Description
End Sub


Private Sub cmdsave_Click()
On Error GoTo errmsg
If Not Len(txtcont) = 10 Then
MsgBox "ENTER VALID CONTACT NUMBER!!"
txtcont = ""
End If


If (txtsid.Text = "" Or txtname.Text = "" Or txtloc.Text = "" Or txtcont.Text = "" Or
txtuname.Text = "" Or txtpwd.Text = "") Then
MsgBox "PLEASE FILL EMPTY FIELDS!!!"
```

```
Else
Adodc1.Recordset.Fields("staffID") = txtsid.Text
Adodc1.Recordset.Fields("sname") = txtname.Text
Adodc1.Recordset.Fields("location") = txtloc.Text
Adodc1.Recordset.Fields("contact") = txtcont.Text
Adodc1.Recordset.Fields("username") = txtuname.Text
Adodc1.Recordset.Fields("password") = txtpwd.Text
Adodc1.Recordset.Update
MsgBox "NEW STAFF ADDED SUCCESSFULLY!!!"
txtsid = ""
txtname = ""
txtloc = ""
txtcont = ""
txtuname = ""
txtpwd = ""
End If
Exit Sub
errmsg:
MsgBox Err.Description
End Sub
```

## STAFF DASHBOARD FORM

```
Private Sub cmdadd_Click()
Form5.Show
```

```
Unload Me

End Sub


Private Sub cmdaddblood_Click()

Form9.Show

Unload Me

End Sub


Private Sub cmdaddpatient_Click()

Form7.Show

Unload Me

End Sub


Private Sub cmdgoto_Click()

Form2.Show

Unload Me

End Sub


Private Sub cmdsearch_Click()

Form8.Show

Unload Me

End Sub
```

```
Private Sub cmdstock_Click()

Form10.Show

Unload Me

End Sub


Private Sub cmdupdate_Click()

Form6.Show

Unload Me

End Sub
```

## DONOR REGISTRATION FORM

```
Private Sub cmdadd_Click()

Dim id As Integer

Dim id1 As String

txtdid.Enabled = True

txtdname.Enabled = True

Combo1.Enabled = True

Combo2.Enabled = True

txtdage.Enabled = True

txtcont.Enabled = True

txtdadd.Enabled = True

txtdondate.Enabled = True

Combo3.Enabled = True
```

```
txtdesc.Enabled = True

txtdid = ""

txtdname = ""

Combo1 = ""

Combo2 = ""

txtdage = ""

txtcont = ""

txtdadd = ""

txtdondate = ""

Combo3 = ""

txtdesc = ""

On Error GoTo errmsg

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

txtdid = "D101"

Else

Adodc1.Recordset.MoveLast

id1 = Adodc1.Recordset("donorID")

id = Mid(id1, 2, 4) + 1

txtdid = "D" & id

End If

txtdname.SetFocus

Adodc1.Recordset.AddNew
```

```
Exit Sub

errmsg:

MsgBox Err.Description

End Sub


Private Sub cmdback_Click()

Form4.Show

Unload Me

End Sub

Private Sub cmdsave_Click()

On Error GoTo errmsg

If Not Len(txtcont) = 10 Then

MsgBox "ENTER VALID CONTACT NUMBER!!"

txtcont = ""

End If

If Not txtdage > 17 Then

MsgBox "NOT ELIGIBLE TO DONATE!!"

txtdage = ""

End If

If (txtdid.Text = "" Or txtdname.Text = "" Or Combo1.Text = "" Or Combo2.Text = "" Or
txtdage.Text = "" Or txtcont.Text = "" Or txtdadd.Text = "" Or txtdondate.Text = "" Or
Combo3.Text = "" Or txtdesc.Text = "") Then

MsgBox "PLEASE FILL EMPTY FIELDS!!!"
```

```
Else
Adodc1.Recordset.Fields("donorID") = txtdid.Text
Adodc1.Recordset.Fields("dname") = txtdname.Text
Adodc1.Recordset.Fields("dblood") = Combo1.Text
Adodc1.Recordset.Fields("dgender") = Combo2.Text
Adodc1.Recordset.Fields("donorage") = txtdage.Text
Adodc1.Recordset.Fields("dcontact") = txtcont.Text
Adodc1.Recordset.Fields("daddress") = txtdadd.Text
Adodc1.Recordset.Fields("dondate") = txtdondate.Text
Adodc1.Recordset.Fields("lastdondate") = Combo3.Text
Adodc1.Recordset.Fields("healthdesc") = txtdesc.Text
Adodc1.Recordset.Update
MsgBox "NEW DONOR ADDED SUCCESSFULLY!!!"
txtdid = ""
txtdname = ""
Combo1 = ""
Combo2 = ""
txtdage = ""
txtcont = ""
txtdadd = ""
txtdondate = ""
Combo3 = ""
txtdesc = ""
```

```
End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub


Private Sub Form_Load()

Combo1.AddItem "A+"

Combo1.AddItem "A-"

Combo1.AddItem "B+"

Combo1.AddItem "B-"

Combo1.AddItem "O+"

Combo1.AddItem "O-"

Combo1.AddItem "AB+"

Combo1.AddItem "AB-"

Combo2.AddItem "MALE"

Combo2.AddItem "FEMALE"

Combo2.AddItem "OTHERS"

Combo3.AddItem "WITHIN A MONTH"

Combo3.AddItem "WITHIN SIX MONTHS"

Combo3.AddItem "BEFORE ONE YEAR"

Combo3.AddItem "NEVER DONATED"

End Sub
```

## UPDATE/ DELETE DONOR FORM

```
Private Sub cmdfind_Click()

txtdid.Enabled = True

txtdname.Enabled = True

Combo1.Enabled = True

Combo2.Enabled = True

txtdage.Enabled = True

txtcont.Enabled = True

txtdadd.Enabled = True

txtdondate.Enabled = True

Combo3.Enabled = True

txtdesc.Enabled = True


Dim id1 As String

Adodc1.Refresh

id1 = txtdid

Adodc1.Recordset.Find "donorID='" & id1 & "'"

If Adodc1.Recordset.EOF Then

MsgBox " DONOR NOT FOUND !!"

Else

txtdid = Adodc1.Recordset.Fields("donorID")

txtdname = Adodc1.Recordset.Fields("dname")
```

```
Combo1 = Adodc1.Recordset.Fields("dblood")

Combo2 = Adodc1.Recordset.Fields("dgender")

txtdage = Adodc1.Recordset.Fields("donorage")

txtcont = Adodc1.Recordset.Fields("dcontact")

txtdadd = Adodc1.Recordset.Fields("daddress")

txtdondate = Adodc1.Recordset.Fields("dondate")

Combo3 = Adodc1.Recordset.Fields("lastdondate")

txtdesc = Adodc1.Recordset.Fields("healthdesc")

End If

Exit Sub

End Sub


Private Sub cmdupdate_Click()

On Error GoTo errmsg

If Not Len(txtcont) = 10 Then

MsgBox "ENTER VALID CONTACT NUMBER!!"

txtcont = ""

End If

If Not txtdage > 17 Then

MsgBox "NOT ELIGIBLE TO DONATE!!"

txtdage = ""

End If
```

```
If (txtdid.Text = "" Or txtdname.Text = "" Or Combo1.Text = "" Or Combo2.Text = "" Or txtdage.Text = "" Or txtcont.Text = "" Or txtdadd.Text = "" Or txtdondate.Text = "" Or Combo3.Text = "" Or txtdesc.Text = "") Then

MsgBox "PLEASE FILL EMPTY FIELDS!!!"

Else

Adodc1.Recordset.Fields("donorID") = txtdid.Text

Adodc1.Recordset.Fields("dname") = txtdname.Text

Adodc1.Recordset.Fields("dblood") = Combo1.Text

Adodc1.Recordset.Fields("dgender") = Combo2.Text

Adodc1.Recordset.Fields("donorage") = txtdage.Text

Adodc1.Recordset.Fields("dcontact") = txtcont.Text

Adodc1.Recordset.Fields("daddress") = txtdadd.Text

Adodc1.Recordset.Fields("dondate") = txtdondate.Text

Adodc1.Recordset.Fields("lastdondate") = Combo3.Text

Adodc1.Recordset.Fields("healthdesc") = txtdesc.Text

Adodc1.Recordset.Update

MsgBox "DONOR UPDATED SUCCESSFULLY!!!"

txtdid = ""

txtdname = ""

Combo1 = ""

Combo2 = ""

txtdage = ""

txtcont = ""
```

```
txtdadd = ""

txtdondate = ""

Combo3 = ""

txtdesc = ""

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub


Private Sub Form_Load()

Combo1.AddItem "A+"

Combo1.AddItem "A-"

Combo1.AddItem "B+"

Combo1.AddItem "B-"

Combo1.AddItem "O+"

Combo1.AddItem "O-"

Combo1.AddItem "AB+"

Combo1.AddItem "AB-"

Combo2.AddItem "MALE"

Combo2.AddItem "FEMALE"

Combo2.AddItem "OTHERS"

Combo3.AddItem "WITHIN A MONTH"
```

```
Combo3.AddItem "WITHIN SIX MONTHS"

Combo3.AddItem "BEFORE ONE YEAR"

Combo3.AddItem "NEVER DONATED"

End Sub


Private Sub cmdback_Click()

Form4.Show

Unload Me

End Sub


Private Sub cmddelete_Click()

On Error GoTo errmsg

Dim confirm As Integer

Adodc1.Refresh

Dim dno As String

dno = txtdid

confirm = MsgBox("DO YOU REALLY WANT TO DELETE THIS DONOR Y/N?",
vbYesNo + vbInformation)

If confirm = vbYes Then

Adodc1.Recordset.MoveFirst

Adodc1.Recordset.Find "donorID='" & dno & "'"

If Adodc1.Recordset.EOF Then

MsgBox "DONOR NOT FOUND!!"
```

```
Else

Adodc1.Recordset.Delete

MsgBox "DONOR DELETED SUCCESSFULLY!!"

txtdid = ""

txtdname = ""

Combo1 = ""

Combo2 = ""

txtdage = ""

txtcont = ""

txtdadd = ""

txtdondate = ""

Combo3 = ""

txtdesc = ""

End If

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub
```

## ADD BLOOD STOCK FORM

```
Private Sub cmdadd_Click()

Dim id As Integer
```

```
Dim id1 As String

txtbno.Enabled = True

txtbgroup.Enabled = True

txtdid.Enabled = True

txtunits.Enabled = True

txtbank.Enabled = True

txtbno = ""

txtbgroup = ""

txtdid = ""

txtunits = ""

txtbank = ""

On Error GoTo errmsg

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

txtbno = "B101"

Else

Adodc1.Recordset.MoveLast

id1 = Adodc1.Recordset("bno")

id = Mid(id1, 2, 4) + 1

txtbno = "B" & id

End If

txtdid.SetFocus

Adodc1.Recordset.AddNew
```

```
Exit Sub

errmsg:

MsgBox Err.Description

End Sub


Private Sub cmdback_Click()

Form4.Show

Unload Me

End Sub

Private Sub cmdfind_Click()

txtdid.Enabled = True

txtbgroup.Enabled = True


Dim id1 As String

Adodc2.Refresh

id1 = txtdid

Adodc2.Recordset.Find "donorID='" & id1 & "'"

If Adodc2.Recordset.EOF Then

MsgBox "DONOR NOT FOUND"

Else

txtdid = Adodc2.Recordset.Fields("donorID")

txtbgroup = Adodc2.Recordset.Fields("dblood")

End If
```

```
Exit Sub

End Sub


Private Sub cmdsave_Click()

On Error GoTo errmsg

If (txtbno.Text = "" Or txtbgroup.Text = "" Or txtdid.Text = "" Or txtunits.Text = "" Or
txtbank.Text = "") Then

MsgBox "PLEASE FILL EMPTY FIELDS!!!"

Else

Adodc1.Recordset.Fields("bno") = txtbno.Text

Adodc1.Recordset.Fields("bloodgroup") = txtbgroup.Text

Adodc1.Recordset.Fields("donorID") = txtdid.Text

Adodc1.Recordset.Fields("units") = txtunits.Text

Adodc1.Recordset.Fields("bloodbank") = txtbank.Text

Adodc1.Recordset.Update

MsgBox "BLOOD DETAILS ADDED SUCCESSFULLY!!!"

txtbno = ""

txtbgroup = ""

txtdid = ""

txtunits = ""

txtbank = ""

End If

Exit Sub
```

```
errmsg:

MsgBox Err.Description

End Sub
```

## VIEW/ DELETE STOCK FORM

```
Private Sub cmdback_Click()

Form4.Show

Unload Me

End Sub


Private Sub cmddelete_Click()

On Error GoTo errmsg

Dim confirm As Integer

Adodc1.Refresh

Dim bno As String

bno = txtbno

confirm = MsgBox("DO YOU REALLY WANT TO DELETE THIS BLOOD RECORD
Y/N?", vbYesNo + vbInformation)

If confirm = vbYes Then

Adodc1.Recordset.MoveFirst

Adodc1.Recordset.Find "bno='" & bno & "'"

If Adodc1.Recordset.EOF Then

MsgBox "BLOOD RECORD NOT FOUND!!"
```

```
Else

Adodc1.Recordset.Delete

MsgBox "BLOOD RECORD DELETED SUCCESSFULLY!!"

End If

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub


Private Sub cmdfind_Click()

Adodc1.RecordSource = " select * from bloodstock where bno like '" & txtbno.Text & "' OR bloodgroup like '" & Combo1.Text & "' AND bloodbank like '" & txtloc.Text & "'"

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

MsgBox " BLOOD RECORDS NOT FOUND !!"

Else

Adodc1.Caption = Adodc1.RecordSource

End If

End Sub


Private Sub Form_Load()

Combo1.AddItem "A+"
```

```
Combo1.AddItem "A-"

Combo1.AddItem "B+"

Combo1.AddItem "B-"

Combo1.AddItem "O+"

Combo1.AddItem "O-"

Combo1.AddItem "AB+"

Combo1.AddItem "AB-"

End Sub
```

## RECIPIENT REGISTRATION FORM

```
Private Sub cmdaddrec_Click()

Dim id As Integer

Dim id1 As String

txtrid.Enabled = True

txtrname.Enabled = True

Combo1.Enabled = True

txtradd.Enabled = True

txtbgroup.Enabled = True

txtbno.Enabled = True

txtrcont.Enabled = True

txtissuedate.Enabled = True

txtrid = ""

txtrname = ""
```

```
Combo1 = ""

txtradd = ""

txtbgroup = ""

txtbno = ""

txtrcont = ""

txtissuedate = ""

On Error GoTo errmsg

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

txtrid = "R101"

Else

Adodc1.Recordset.MoveLast

id1 = Adodc1.Recordset("recipientID")

id = Mid(id1, 2, 4) + 1

txtrid = "R" & id

End If

txtrname.SetFocus

Adodc1.Recordset.AddNew

Exit Sub

errmsg:

MsgBox Err.Description

End Sub
```

```
Private Sub cmdback_Click()

Form4.Show

Unload Me

End Sub


Private Sub cmdfind_Click()

txtbno.Enabled = True

txtbgroup.Enabled = True

Dim id1 As String

Adodc2.Refresh

id1 = txtbno

Adodc2.Recordset.Find "bno='" & id1 & "'"

If Adodc2.Recordset.EOF Then

MsgBox "BLOOD RECORD NOT FOUND"

Else

txtbno = Adodc2.Recordset.Fields("bno")

txtbgroup = Adodc2.Recordset.Fields("bloodgroup")

End If

Exit Sub

End Sub


Private Sub cmdsaverec_Click()

On Error GoTo errmsg
```

```
If Not Len(txtrcont) = 10 Then

MsgBox "ENTER VALID CONTACT NUMBER!!"

txtrcont = ""

End If


If (txtrid.Text = "" Or txtrname.Text = "" Or Combo1.Text = "" Or txtbno.Text = "" Or
txtradd.Text = "" Or txtbgroup.Text = "" Or txtrcont.Text = "" Or txtissuedate.Text = "")
Then

MsgBox "PLEASE FILL EMPTY FIELDS!!!"

Else

Adodc1.Recordset.Fields("recipientID") = txtrid.Text

Adodc1.Recordset.Fields("rname") = txtrname.Text

Adodc1.Recordset.Fields("rgender") = Combo1.Text

Adodc1.Recordset.Fields("rblood") = txtbgroup.Text

Adodc1.Recordset.Fields("rcontact") = txtrcont.Text

Adodc1.Recordset.Fields("raddress") = txtradd.Text

Adodc1.Recordset.Fields("issuedate") = txtissuedate.Text

Adodc1.Recordset.Fields("bno") = txtbno.Text

Adodc1.Recordset.Update

MsgBox "NEW RECIPIENT ADDED SUCCESSFULLY!!!"

txtrid = ""

txtrname = ""

Combo1 = ""
```

```
txtradd = ""

txtbgroup = ""

txtbno = ""

txtrcont = ""

txtissuedate = ""

End If

Exit Sub

errmsg:

MsgBox Err.Description

End Sub


Private Sub Form_Load()

Combo1.AddItem "MALE"

Combo1.AddItem "FEMALE"

Combo1.AddItem "OTHERS"

End Sub
```

## SEARCH FOR A DONOR FORM

```
Private Sub cmdback_Click()

Form4.Show

Unload Me

End Sub

Private Sub cmdsearch_Click()
```

```
Adodc1.RecordSource = " select * from donor where dblood like '" & Combo1.Text & "'
AND daddress like '" & txtloc.Text & "'"

Adodc1.Refresh

If Adodc1.Recordset.EOF Then

MsgBox " DONOR UNAVAILABLE !!"

Else

Adodc1.Caption = Adodc1.RecordSource

End If

End Sub


Private Sub Form_Load()

Combo1.AddItem "A+"

Combo1.AddItem "A-"

Combo1.AddItem "B+"

Combo1.AddItem "B-"

Combo1.AddItem "O+"

Combo1.AddItem "O-"

Combo1.AddItem "AB+"

Combo1.AddItem "AB-"

End Sub
```

# 8. IMPLEMENTATION

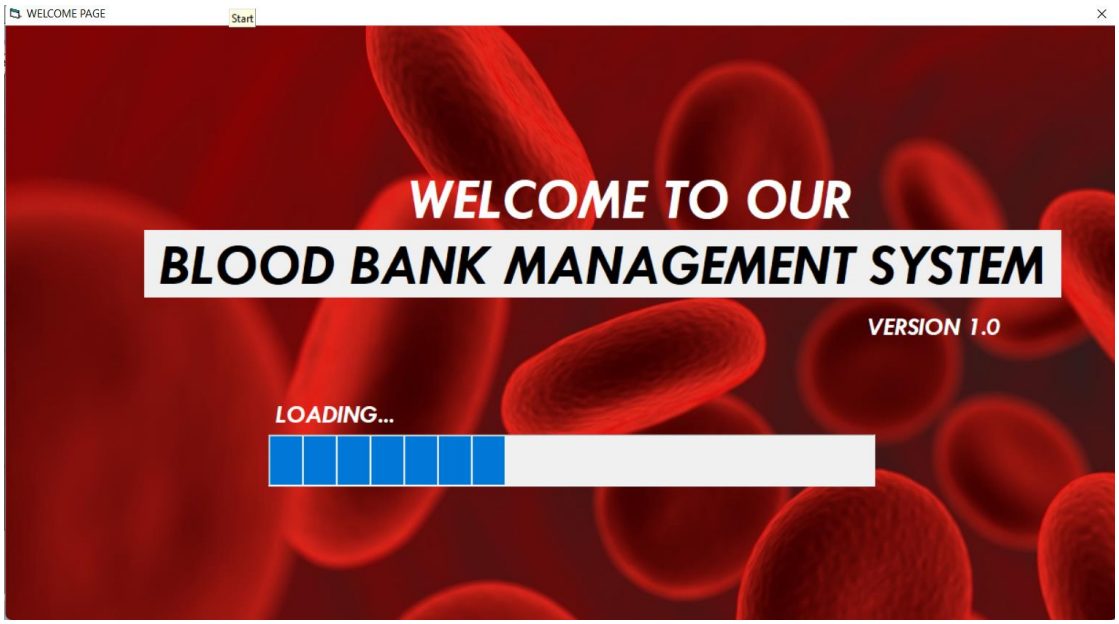



Fig 8.1 WELCOME PAGE

This page will redirect the user to the login screen

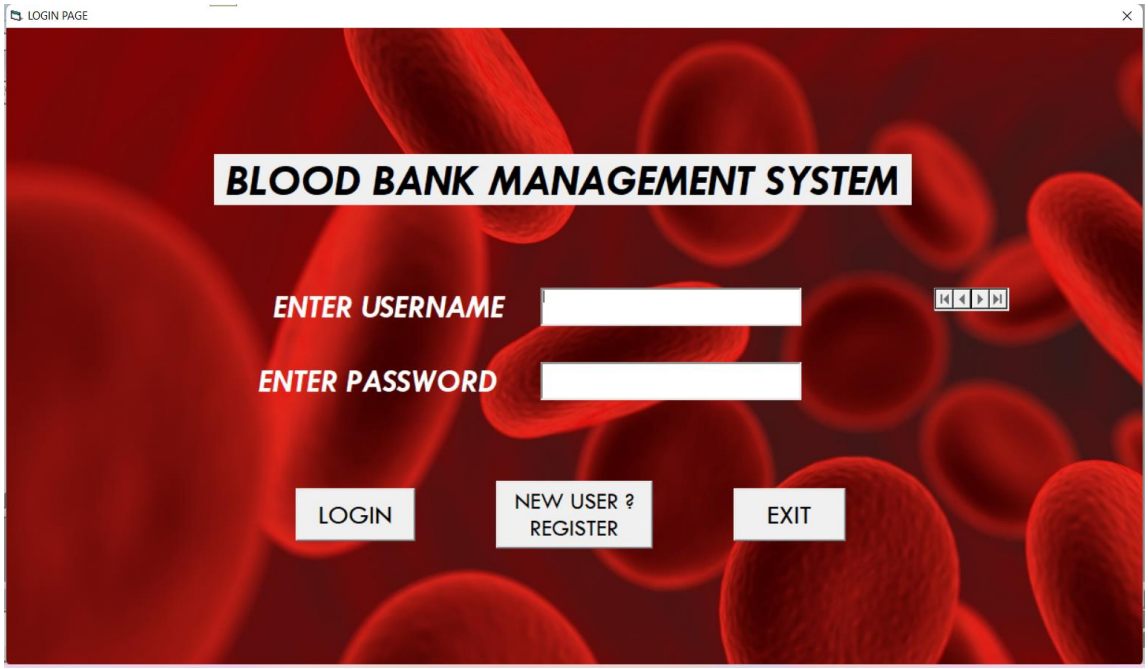



Fig 8.2 STAFF LOGIN

Here admin must provide username and password to Log In into the application.

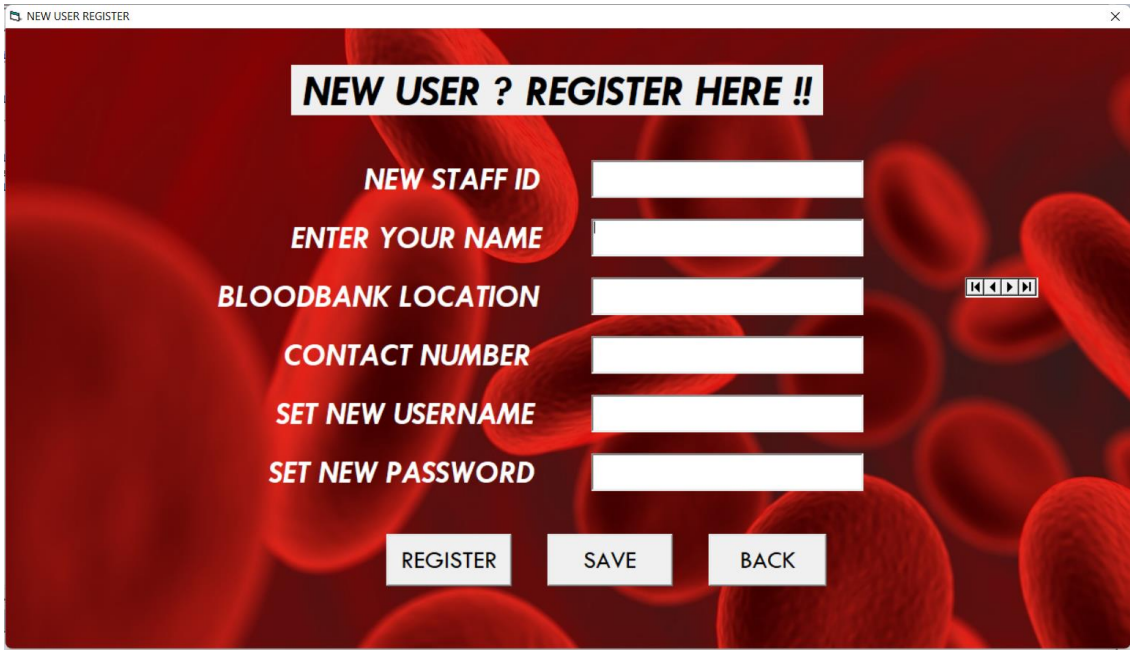



Fig 8.3 NEW USER REGISTRATION

A new user/staff can register here along with new a username and password

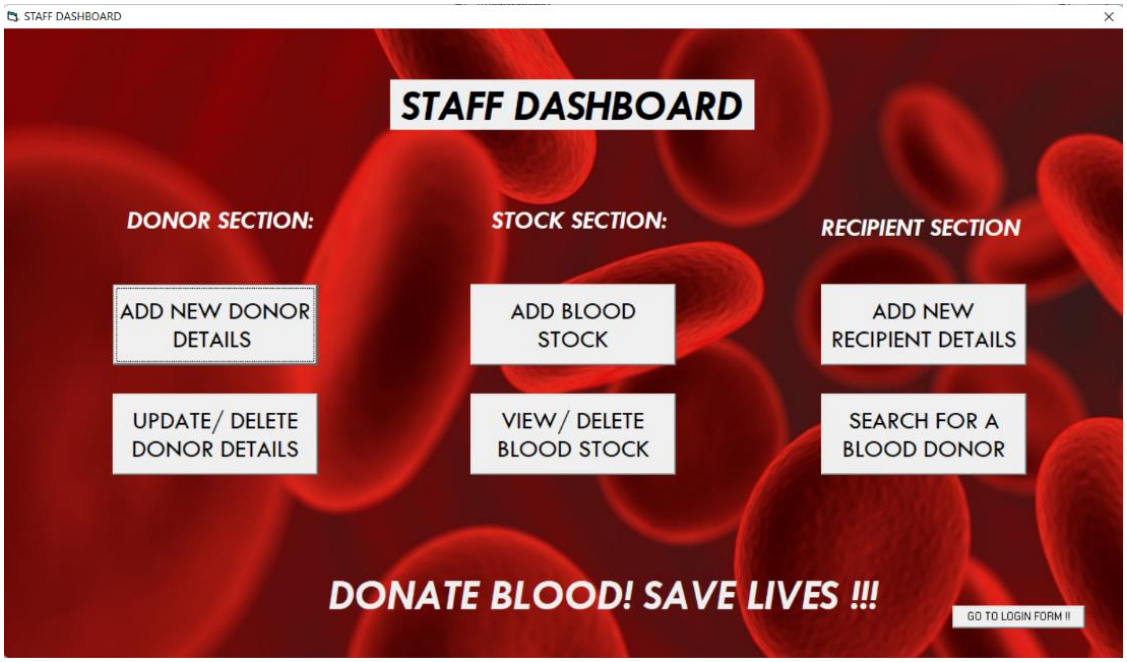



Fig 8.4 STAFF DASHBOARD

This form will display all the operations which an admin can perform using this application.

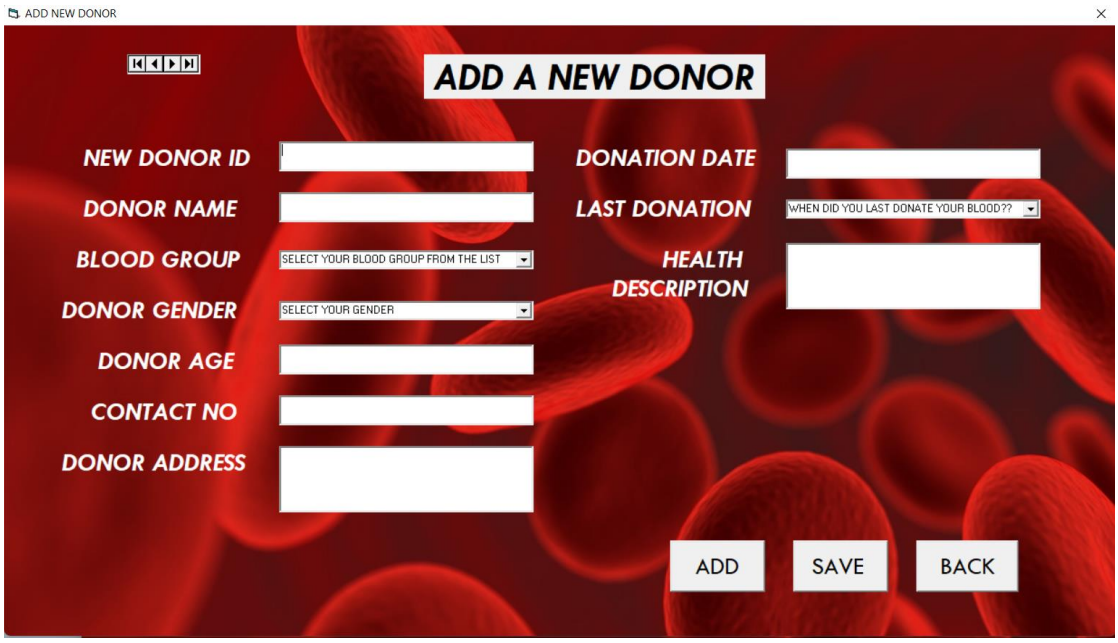



Fig 8.5 DONOR REGISTRATION

Here, Admin can register details of a new donor.

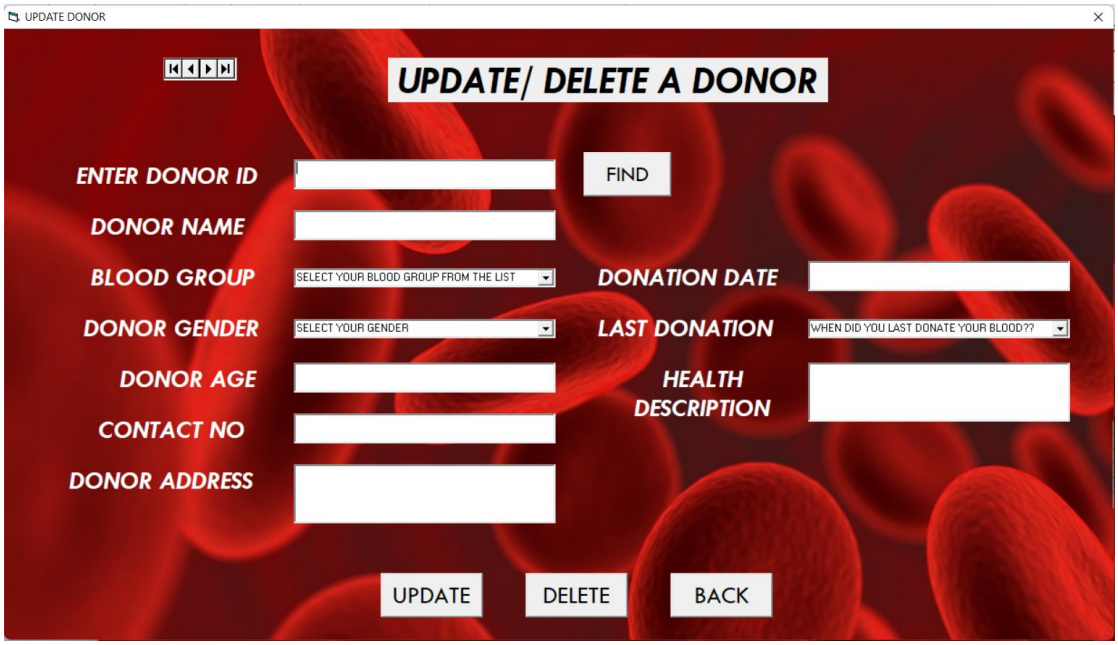



Fig 8.6 UPADTE EXISTING DONOR

Here, Admin can update the details of an existing donor.

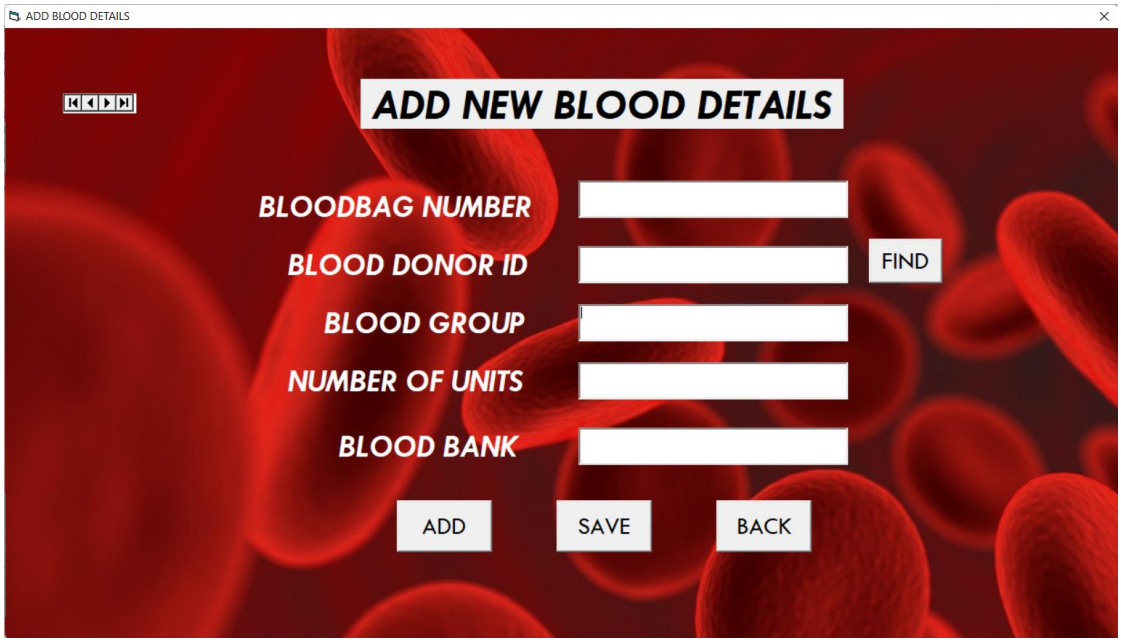



Fig 8.7 ADD BLOOD STOCK

Here, Admin can add new blood details into stock.

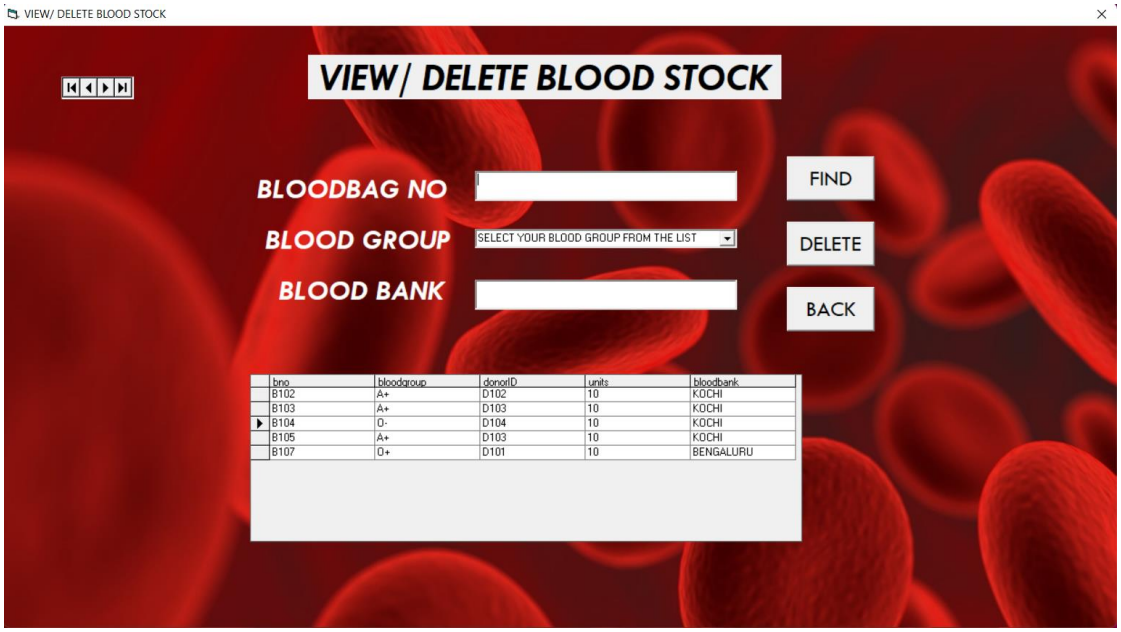



Fig 8.8 VIEW/ DELETE BLOOD STOCK

Here, Admin can view or delete a particular record from the blood stock.

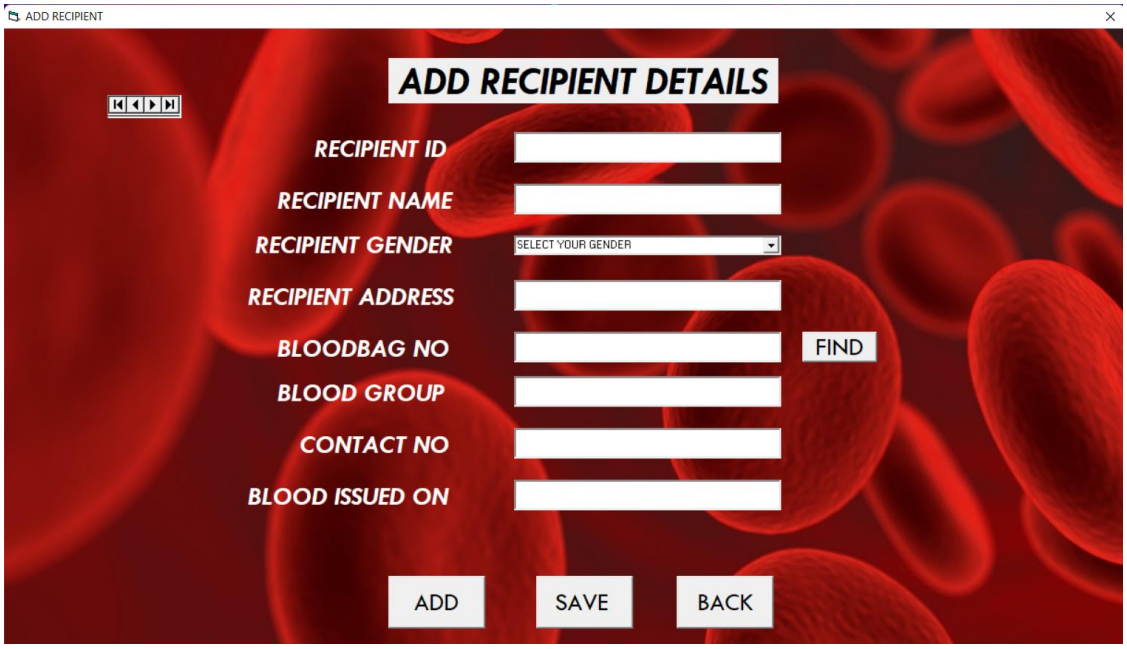



Fig 8.9 RECIPIENT REGISTRATION

Here, Admin can register details of a new recipient.

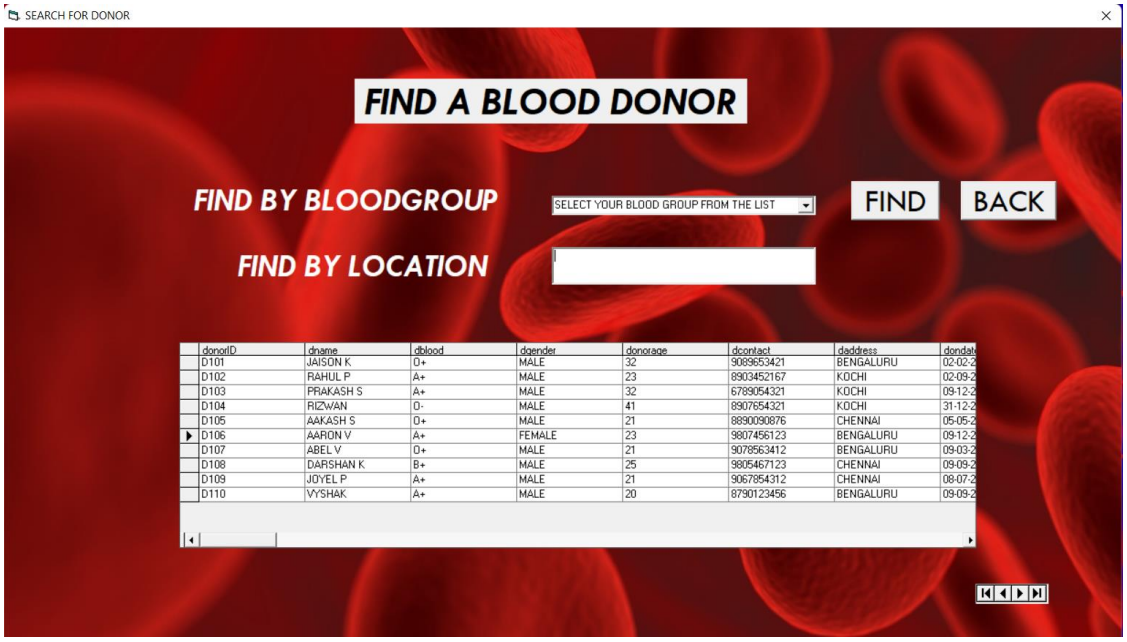



Fig 8.10 SEARCH FOR A BLOOD DONOR

Here, Admin can search for a blood donor by giving a specific blood group and location.

# 9. TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

This tutorial will give you a basic understanding on software testing, its types, methods, levels, and other related terminologies.

## System Testing

System testing is testing conducted on a complete, integrated system to evaluate its compliance with the specified requirements. After the completion of the integration testing, the product is passed for system testing. System testing is undertaken by independent testers who haven't played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements by the stakeholder.

## Integration Testing

Integration testing is performed to test individual components to check how they function together. In other words, it is performed to test the modules which are working fine individually and do not show bugs when integrated. It is the most common functional testing type and performed as automated testing.

## Unit Testing

In computer_programming, unit testing is a software_testing method by which individual units of source_code, sets of one or more computer program modules together with associated control data and operating procedures, are tested to determine whether they are fit for.

## 9.1 TEST CASE 1

## LOGIN FORM

| SL.NO | Description | Expected Result | Actual Result | Status |
|---|---|---|---|---|
| 1. | Staff Login Form | Valid Username and Password | Valid Username and Password | Pass |
| | | | Invalid Username | Fail |
| | | | Invalid Password | Fail |
| | | | Blank Input values | Not Applicable |

### 9.1.2 VALID USERNAME AND PASSWORD

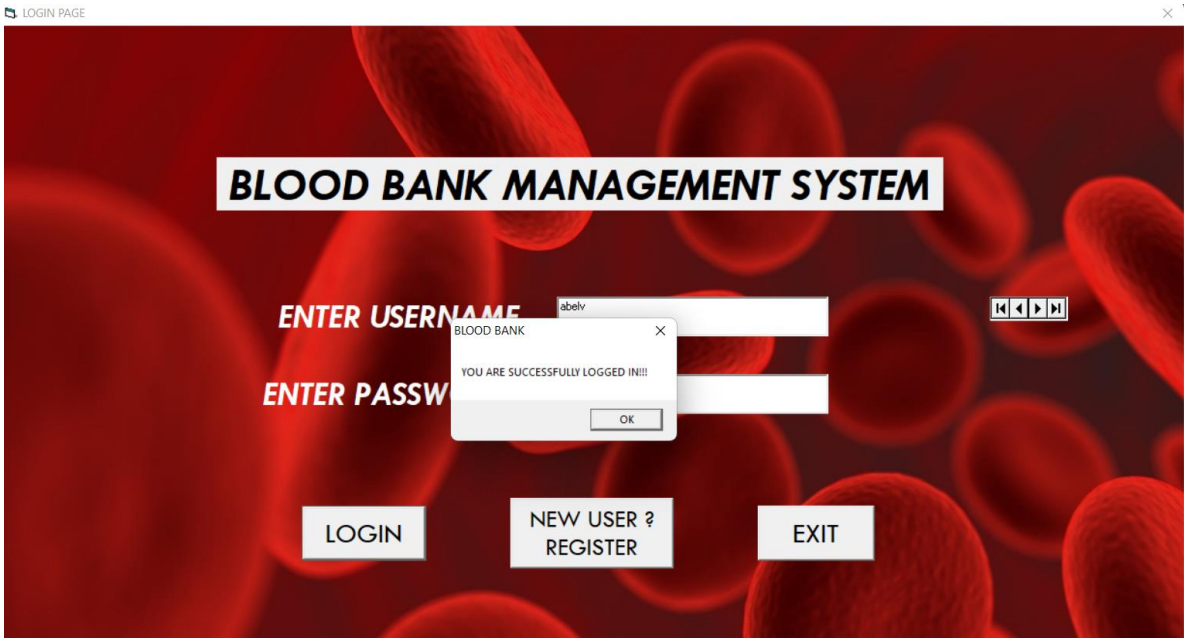



Fig 1.1

User entered valid login credentials. Login Successful!!!

## 9.1.2 INVALID USERNAME

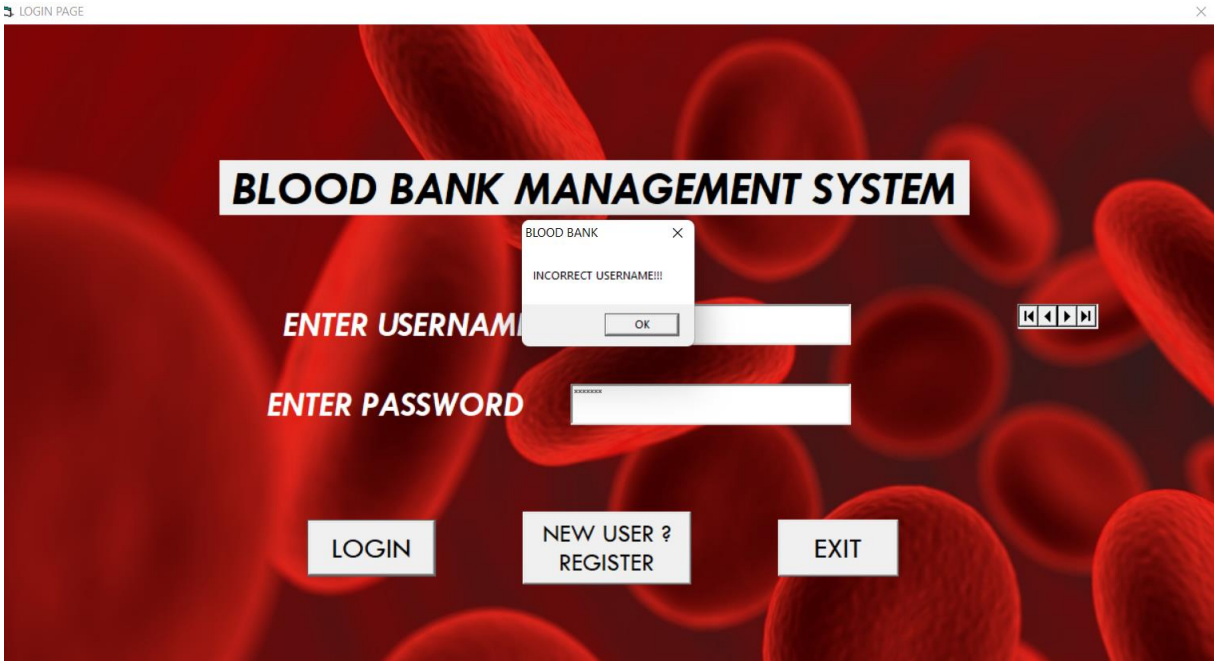



Fig 1.1

User entered wrong username. Login restricted!!!

## 9.1.3 INVALID PASSWORD

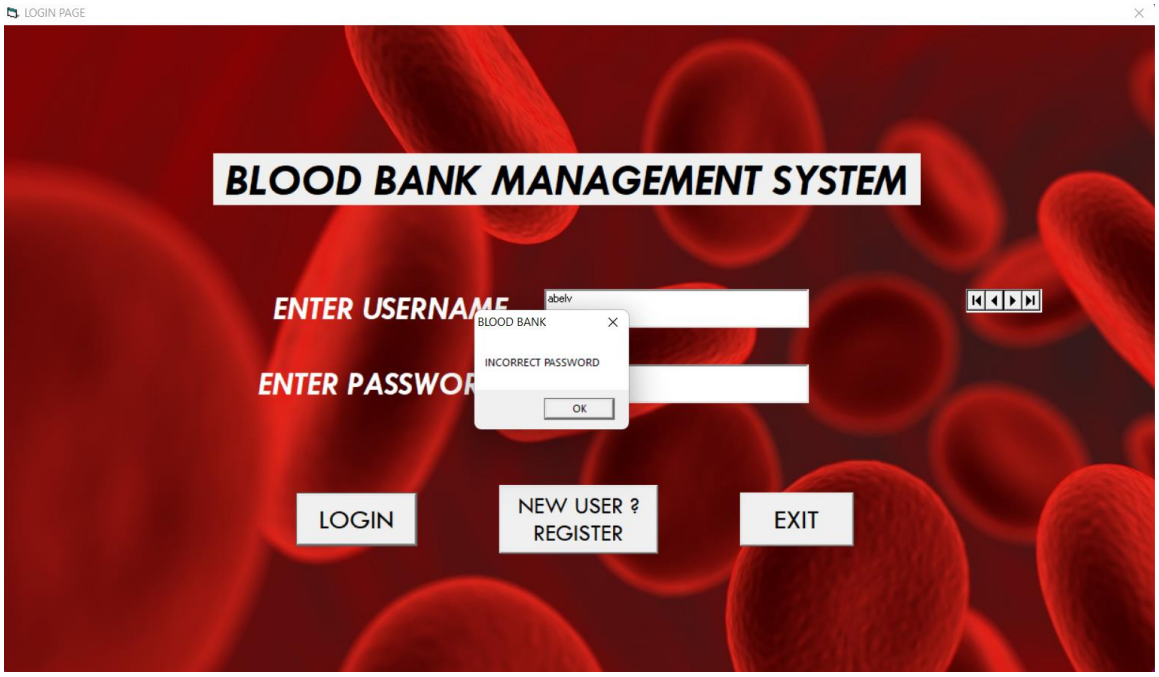



Fig 1.3

User entered wrong password. Login restricted!!!

## 9.1.4 BLANK INPUT VALUES

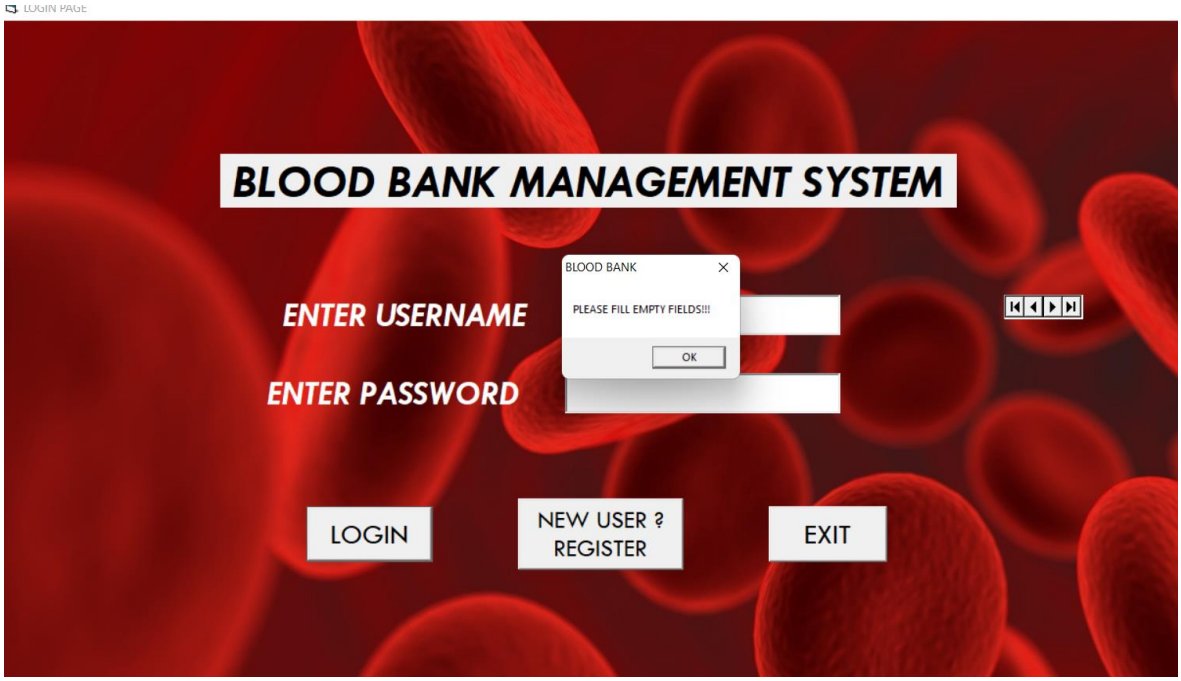



Fig 1.4

Login Restricted if any fields are left empty.

## 9.2 TEST CASE 2

## DONOR REGISTRATION FORM

| SL.NO | Description | Expected Result | Actual Result | Status |
|---|---|---|---|---|
| 1. | Donor Registration Form | All details must be valid-<br>Valid Name<br>Valid Age<br>Valid Contact | All details must be valid-<br>Valid Name<br>Valid Age<br>Valid Contact | Pass |
| | | | Invalid Name<br>(Special Symbols)<br>Valid Age<br>Valid Contact | Fail |
| | | | Valid Name<br>Invalid Age<br>(Less than 18 Years)<br>Valid Contact | Fail |
| | | | Valid Name<br>Valid Age<br>Invalid Contact<br>(less than 10 digits) | Fail |

## 9.2.1 REGISTRATION SUCCESSFUL

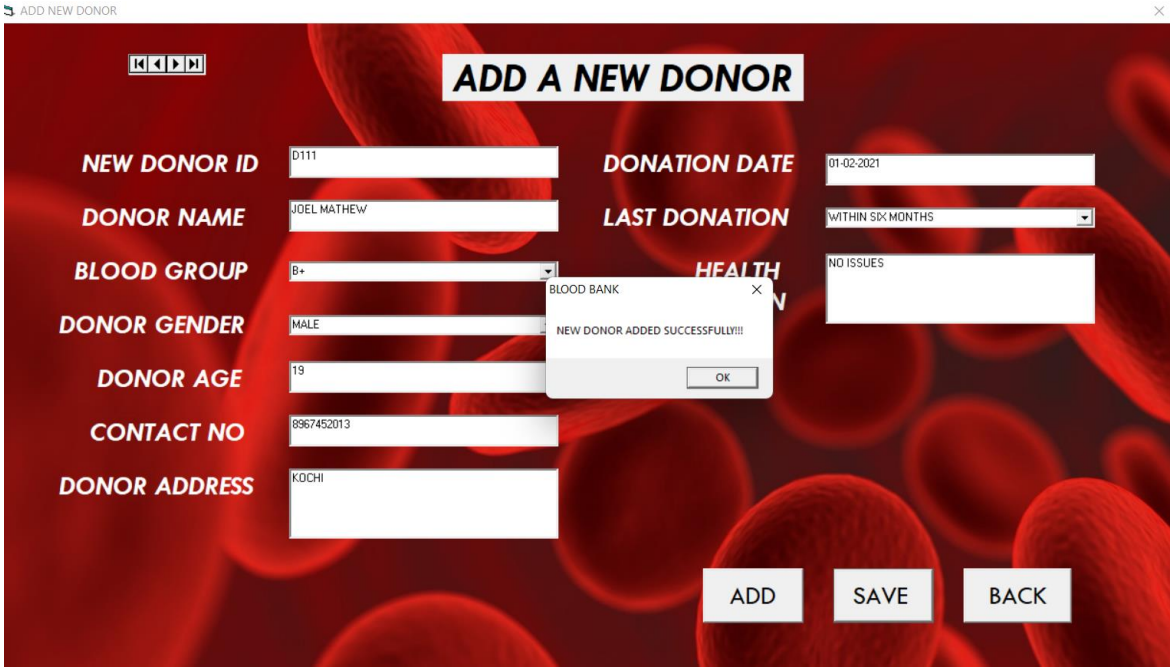



Fig 2.1

Donor is added successfully if all the details are correct and Valid.

## 9.2.2 INVALID CONTACT NUMBER

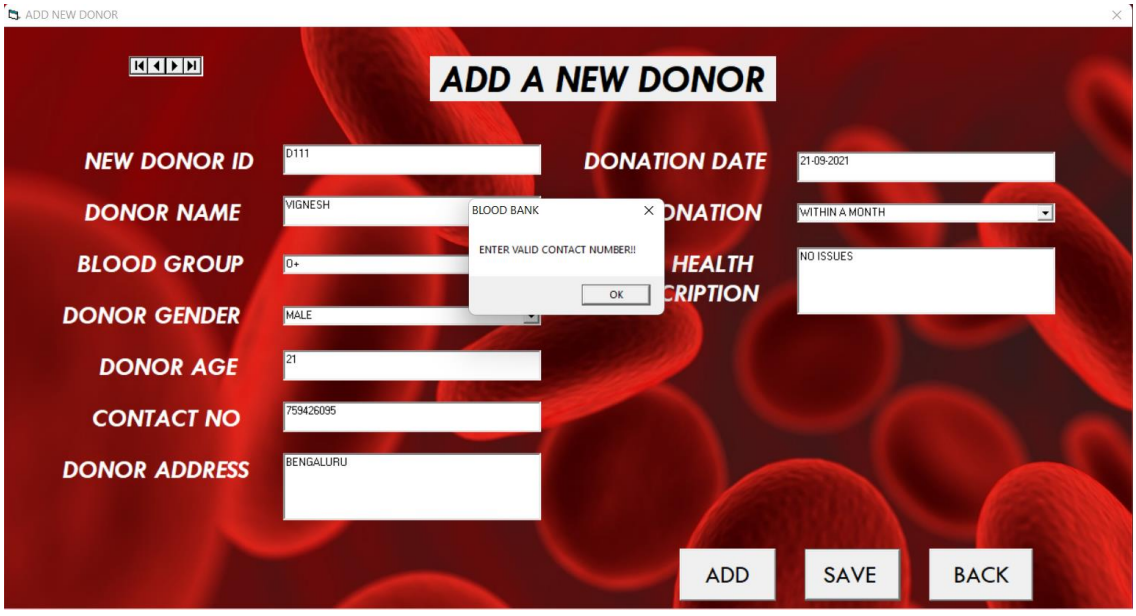



Fig 2.2

Error Message pops up if Contact No. entered is more than 10 digits

## 9.2.3 AGE BELOW 18

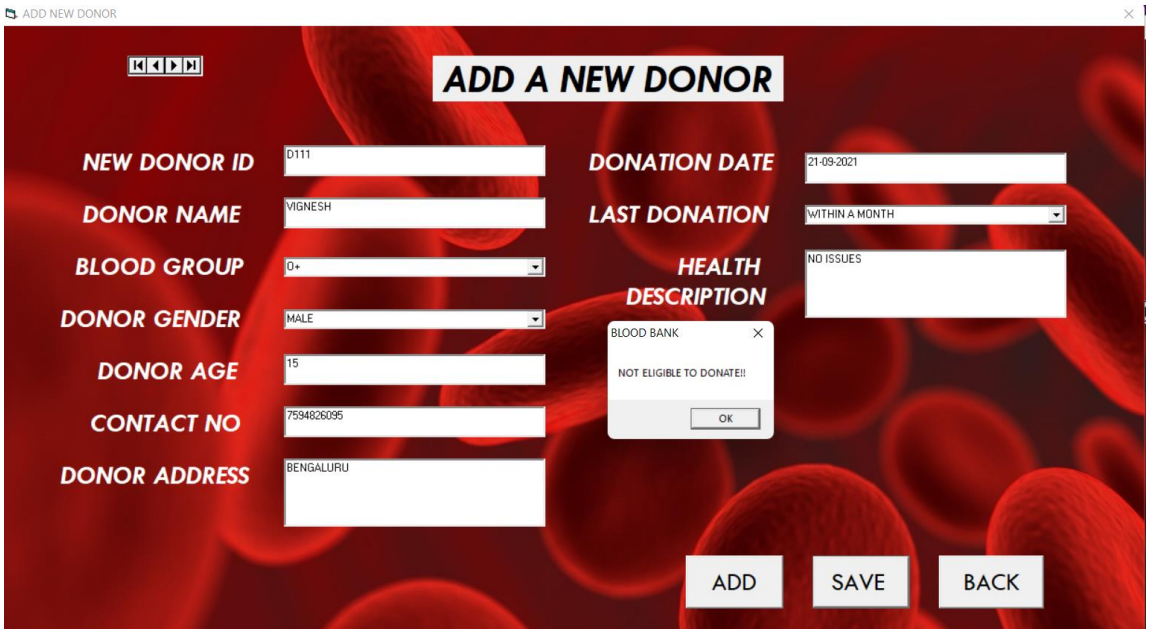



Fig 2.3

Donor cannot register if his age is below 18.

## 9.4 TEST CASE 3

## RECIPIENT REGISTRATION FORM

| SL.NO | Description | Expected Result | Actual Result | Status |
|---|---|---|---|---|
| 1. | Recipient Registration Form | All details must be valid-<br>Valid Name<br>Valid Contact<br>Valid Date | All details must be valid-<br>Valid Name<br>Valid Contact<br>Valid Date | Pass |
| | | | Valid Name<br>Invalid Contact<br>(less than 10 digits)<br>Valid Date | Fail |
| | | | Valid Name<br>Valid Contact<br>Invalid Date<br>(not in dd-mm-yYyy format) | Fail |
| | | | Blank Input values | Not Applicable |

## 9.3.1 REGISTRATION SUCCESSFUL

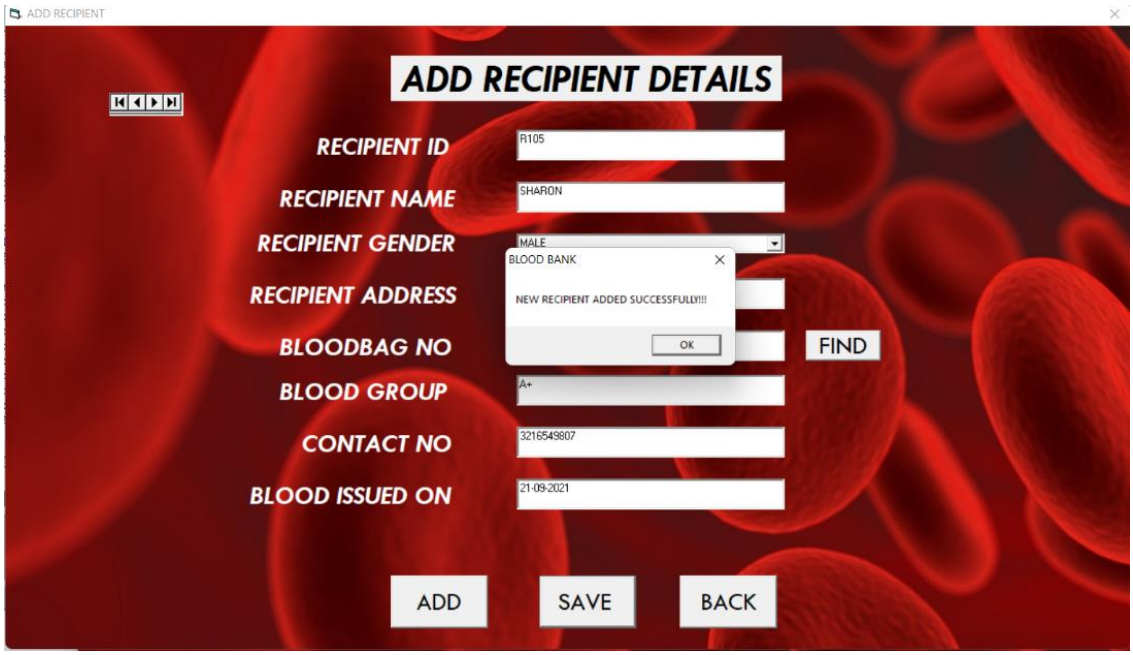



Fig 3.1

Recipient added successfully if all the details are correct and Valid.

## 9.3.2 INVALID CONTACT NUMBER

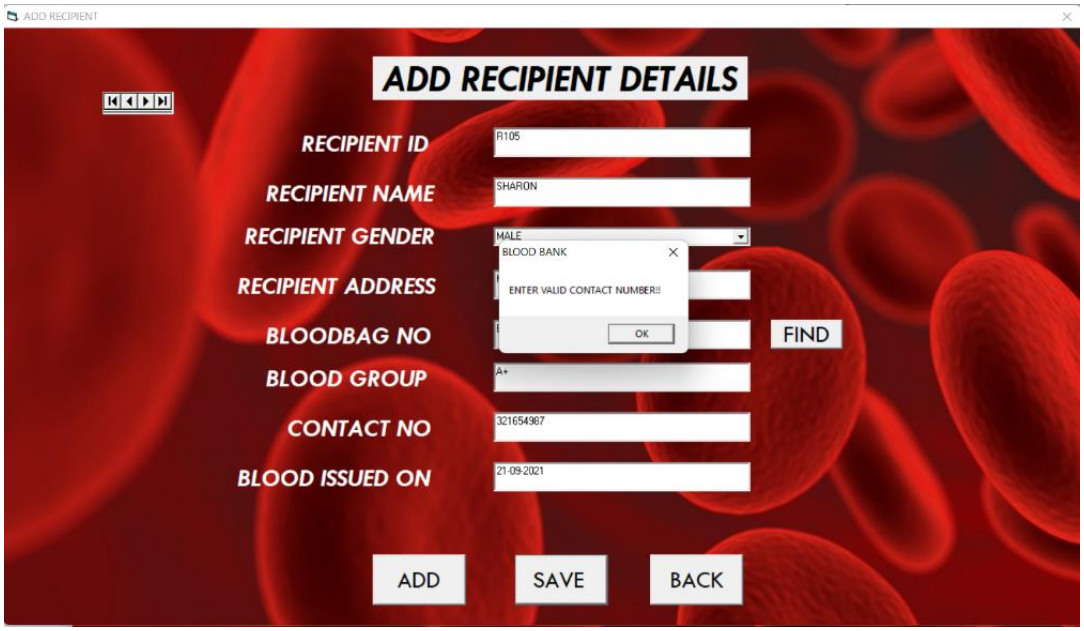



Fig 3.2

Error Message pops up if Contact No. entered is more than 10 digits

### 9.3.3 BLANK INPUT VALUES

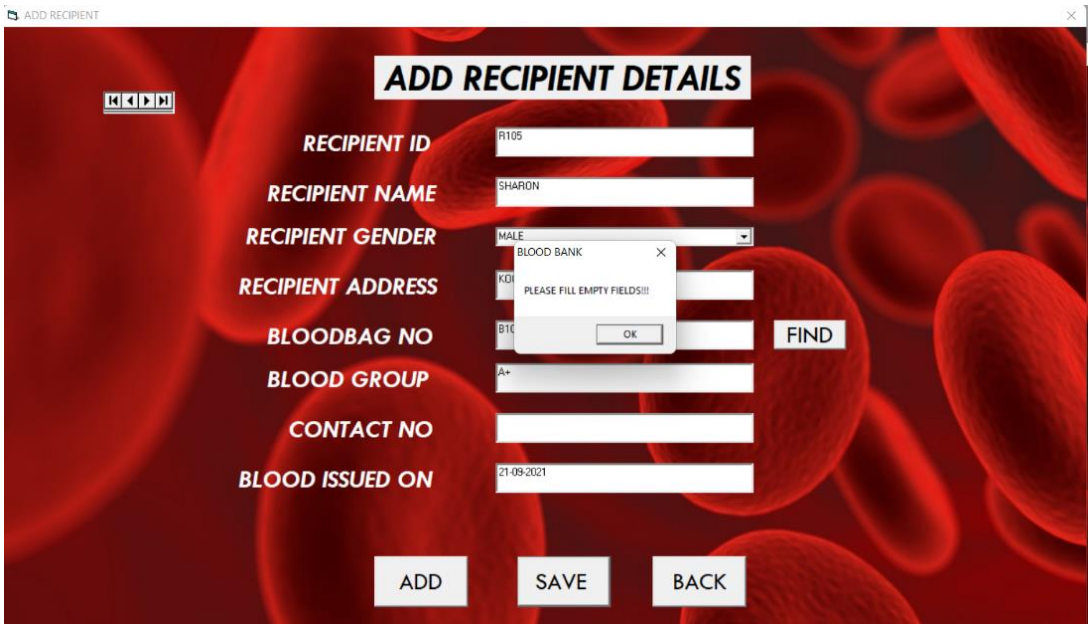



Fig 3.3

Details Not added if any input field is left empty.

# 10. CONCLUSION

The project “Blood bank Management System” is developed to maintain the Donor order details with the blood details and recipient details. This application helps the admin to add, update, delete and search these details from the database.

All these details are entered and retrieved manually, as there are many disadvantages like time consuming, inaccuracy of data. To overcome these problems, this application is introduced, which is a computerised version of the existing system which provides easy and quick access of data and helps in reliable maintenance of records.

# 11. FUTURE ENHANCEMENTS

This project works at an administrative end where only an admin or a staff can access the application and manage the records.

In future, This Standalone application can be converted into an Internet Based or online application.

- Enabling the Donors and recipient do the registration process through online without any assistance from the admin or a blood bank staff.
- And also, recipients can search for a specific blood by giving blood group and location which helps them to check the availability without wasting their time by physically going to a blood bank.

# 12. BIBLIOGRAPHY

1. Visual Basic 6: The Complete Reference by Noel Jerke
2. Visual Basic 6 Black book by Steven Holzner
3. SQL: The Complete Reference, Third Edition by James Groff
4. Introducing Microsoft SQL Server 2016 by Dennis Cherry
5. Beginning SQL Modelling 2008 by Bart Weller
6. www.tutorialspoint.com
7. www.stackoverflow.com