information if necessary. The administrator is also responsible for responding to

the hospital’s blood requests and checking the stocks in the blood bank’s inventory

information if necessary. The administrator is also responsible for responding to

the hospital’s blood requests and checking the stocks in the blood bank’s inventory

information if necessary. The administrator is also responsible for responding to

the hospital’s blood requests and checking the stocks in the blood bank’s inventory

2. Statement of the Problem

The following problem arises when using a typical blood bank’s existing system

2. Statement of the Problem

The following problem arises when using a typical blood bank’s existing system

**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

* The project blood bank management system is known to be a pilot
* project that is designed for the blood bank to gather blood from
* various sources and distribute it to the needy people who have high Requirements for it.
* The Software is designed to handle the daily transaction of the blood bank and Search the details when required.
* It also helps to register the details of donors, blood collection details as well as blood issued reports.
* The software Application is designed in such a manner that it can suit the needs of all the blood bank requirements in the course of future.
* It will help us to find the Blood group with its most efficient time to take care of the blood and it is more easy to hand over the blood to the hospital to help people to get blood on time.
* This all thing is been stored and been seen in this blood bank management system. To help more people trying best to do so.
  + 1. **General Overview of the Problem**

The problems are:

* Tracking the database was complicated when the details are maintained manually.
* It is time consuming and space consuming.
* Scarcity of rare bloods.
* Unavailability of blood during emergency.
* Less awareness among people about blood donation and blood transfusion.
  1. **BENEFITS OF BLOOD BANK MANAGEMENT SYSTEM**

A blood bank management system (BBMS) offers a wide range of benefits for both blood banks and the healthcare system at large. Here are some key advantages:

* Real-time data: Track blood units, their types, expiry dates, and location in real-time, preventing shortages and waste.
* Reduced wastage: Monitor expiry dates and prioritize use of older units, minimizing wastage of precious blood products.
* Optimal stock levels: Analyse historical data and predict future demand to maintain optimal stock levels and avoid critical shortages.
* Faster access to blood: Locate compatible blood units quickly and efficiently, reducing patient waiting times and improving care.
* Reduced transfusion errors: Track blood units throughout the chain of custody and minimize the risk of mismatched transfusions.
* Improved patient outcomes: Ensure timely and safe access to blood products for improved patient outcomes and reduced mortality rates.

**FUTURE LOOK**

The Future of Blood Bank Management Systems: A Glimpse into a Smoother, Smarter, and Safer Tomorrow

**1.3.1 AI-powered Inventory Management:**

Predictive analytics: Imagine a system that not only tracks current blood stocks but also predicts future demand based on factors like seasonal trends, disease outbreaks, and surgical schedules. This would allow for proactive inventory management, preventing critical shortages and ensuring optimal preparedness.

* + 1. **Blockchain-enabled Transparency and Traceability**:

Unbreakable chain of custody: Block chain technology can create a secure and transparent digital ledger for every blood unit, tracking its journey from donor to recipient. This would enhance traceability, minimize the risk of errors, and build trust in the blood supply chain.

* + 1. **Connected Devices and the Internet of Medical Things (IoMT):**
* Real-time monitoring: Smart sensors embedded in blood storage units and refrigerators could provide real-time data on temperature, humidity, and other critical parameters.
* This would allow for remote monitoring and immediate alerts in case of any deviations, ensuring optimal storage conditions and preserving blood quality.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1. EXISTING SYSTEMS**

**2.1.1 Blood Bank Management System**

**2.1.1.1 Introduction**

India suffers from an annual deficit of two million units, as only 1% of the Indian population donates blood as stated by the World Health Organization (WHO). Due to substandard medical facilities and practices in many parts of the country, there have been cases of transmission of infectious diseases like AIDS. The need for blood is increasing along with its importance for treating various medical conditions. There are three main components of blood; plasma, platelet, and RBC/WBC. Especially during this covid pandemic, we’re seeing a huge spike in the requirement of blood plasma from the patients who were recovered from covid-19 as their Convalescent Plasma now contains covid-19 antibodies.

**2.1.1.2 Methodology:**

The methodology chosen to develop the Blood bank system is the Rational Unified Process (RUP) from IBM developer works. RUP is a multi-layered adaptive process designed for software project teams that use their process elements as they scale up.

**2.1.1.3 Problem:**

* + Cannot identify all the factors that affect blood. Like power failure, natural disasters, transport accidents.
* It is difficult to predict the uncertainty in the requirements for major road accidents.
* Large qualities of blood are being wasted. 4 2.1.2 Enhancing blood transfusion safety through the use of online blood bank management system -Shinas College of technology.

**2.1.2. Enhancing blood transfusion safety through the use of online blood bank management system**

**2.1.2.1 Introduction:**

Blood transfusion safety remains an important public health concern in Oman. The availability of blood products of all blood types and the provision of its safety ensure public trust of its excellent healthcare system. However, lack of availability of these blood products and provision of unsafe blood products still impacts morbidity and mortality in the Sultanate. Through the use of online blood bank management system, blood transfusion safety is expected to be enhanced or improved. Risks on improper blood donors’ documentation and misplaced records can be minimized or totally avoided.

**2.1.2.2 Methodology:**

The researchers used both descriptive research and experimental research methodology. The study was descriptive because it describes the nature of the situations it exists at the time of study.

**2.1.2.3 Problems:**

Blood donors and patients or recipients of blood donation are not system user, their registration or information will be encoded by the blood bank receptionists.

**2.1.3 The Prospect and Significance of Lifeline: E-Blood Bank System-**

**2.1.3.1 Introduction**

The requirements for the blood are an important factor in the contemporary medicine and the health care. For every second there will be an individual who needs blood to save life. Blood transfusion is a lifesaving intervention that has n essential role in the total patient management within health care system. Over 4 million people are affected with infected with human immune virus by unsafe blood transfusion, 99% of 500,000 women die yearly with haemorrhage during pregnancy of childbirth Methodology: Data were collected using a self-administered survey distributed through the internet.

**2.1.3.2 Problem:**

* Integration of some additional functionality in the system for better services.

**CHAPTER 3**

**FEASIBILITY STUDY**

**3.1 FEASIBILITY STUDY**

Feasibility study aims to uncover the strength and weaknesses of the proposed project objectively and rationally. A feasibility study analyses the viability of a project to determine whether the project or venture is likely to succeed. The study is also designed to identify potential issues and problems that could arise while pursuing the project.

**3.1.1 Technical Feasibility**

* Available Technology: Are the necessary technologies readily available and compatible?
* Implementation Complexity: Can the system be implemented within budget and timeframe constraints?
* Scalability: Is the system designed to handle future growth and increased demand?
* Data Security: Will the system ensure the safety and privacy of sensitive patient and donor information?

**3.1.2. Operational Feasibility**

* Staffing: Are qualified personnel available to operate the system effectively?
* Regulations and Compliance: Does the system meet all relevant blood banking regulations and standards?
* Workflow and Processes: Are the system's workflows efficient and optimized for smooth operation?
* Sustainability: Can the system be sustained financially through funding or revenue generation?

**3.1.3 Financial Feasibility**

* Costs: Estimate the initial investment and on-going operational costs.
* Funding Sources: Identify potential funding sources such as grants, donations, or revenue from service fees.
* Financial Projections: Forecast the potential revenue and cost-savings to assess the system's financial viability.
* Break-even Analysis: Determine the number of blood donations or service transactions needed to cover the costs.

**3.1.4 Schedule Feasibility:**

* Needs Analysis: Interview stakeholders (hospital staff, donors, patients) to understand needs and requirements.
* System Design: Design the software and hardware architecture, workflow, and user interface.
* Technology Acquisition: Procure necessary hardware and software licenses.

**3.1.5 Legal Feasibility:**

* Data Privacy: Ensure compliance with relevant data privacy regulations (e.g., HIPAA, GDPR).
* Security: Implement robust security measures to protect sensitive patient and donor information.
* Regulatory Compliance: Meet all applicable blood banking regulations and standards (e.g., FDA, accreditation requirements).
* Licensing and Permits: Obtain necessary licenses and permits for operating a blood bank.
* Contractual Agreements: Develop standard contracts for blood supply agreements with hospitals and other partners.
* Liability and Insurance: Understand and mitigate potential legal risks through insurance coverage and clear policies.

**3.2 FUNCTIONAL REQUIREMENT:**

* FR1: User Login

Administrator shall be able to log in to the system using their credentials.

* FR2: Donor

The system shall maintain a database of donor profiles, including their medical history and donation history.

Administrators shall be able to add and update donor profiles.

* FR3: Search Donor

Administrators shall be able to Search donor through appropriate address and blood group.

* FR4: Delete Donor

Administrators shall be able to delete donor profiles by just searching donor Id.

* FR5: Logout

Administrators shall be able to Logout from the application.

**3.3 NON-FUNCTIONAL REQUIREMENTS**

* NFR1: Performance

The system shall handle a maximum number of simultaneous users.

* NFR2: Security

User data shall be encrypted during login transmission and can be only edited by administrator..

* NFR3: Availability

The system shall be available 24/7 with a planned downtime of no more than 2 hours per month for maintenance.

* NFR4: Usability

The user interface shall be intuitive and easy to navigate.

* NFR5: Compliance

The system shall comply with all relevant medical and data protection regulations.

**3.4 HARDWARE SPECIFICATIONS**

* Processor must be i3 or more
* RAM must be 4GB or more
* Windows must be of latest version i.e. Windows 11 or 10.

**3.5 SOFTWARE SPECIFICATIONS**

* For frontend development, Visual Studio 2022 platform.
* For backend development, Microsoft SQL server management studio.

**CHAPTER 4**

**DESIGNED MODULES**

**4.1 USER LOGIN**

Administrator shall be able to log in to the system using their credentials.

**4.2 DONOR**

The system shall maintain a database of donor profiles, including their medical history and donation history. Administrators shall be able to add and update donor profiles.

**4.3 SEARCH DONOR**

Administrators shall be able to Search donor through appropriate address and blood group.

**4.4 DELETE DONOR**

Administrators shall be able to delete donor profiles by just searching donor Id.

**4.5 LOG OUT**

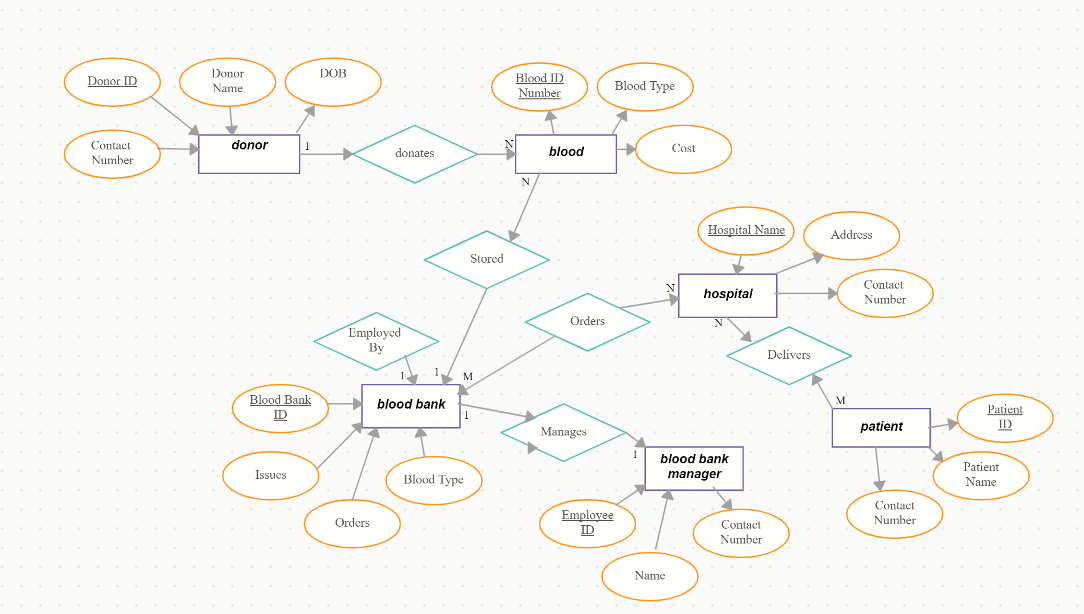
Administrators shall be able to LogOut from the application.

**CHAPTER 5**

**DATABASE DESIGN**

**5.1 ER DIAGRAM:**

This is just a basic ER diagram that can be customized to meet the specific needs of your blood bank management system. You may need to add additional entities, relationships, and attributes to reflect your system's functionality and data requirements.

****Fig 5.1 ER Diagram

**5.2 USED CASE DIAGRAM:**

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures

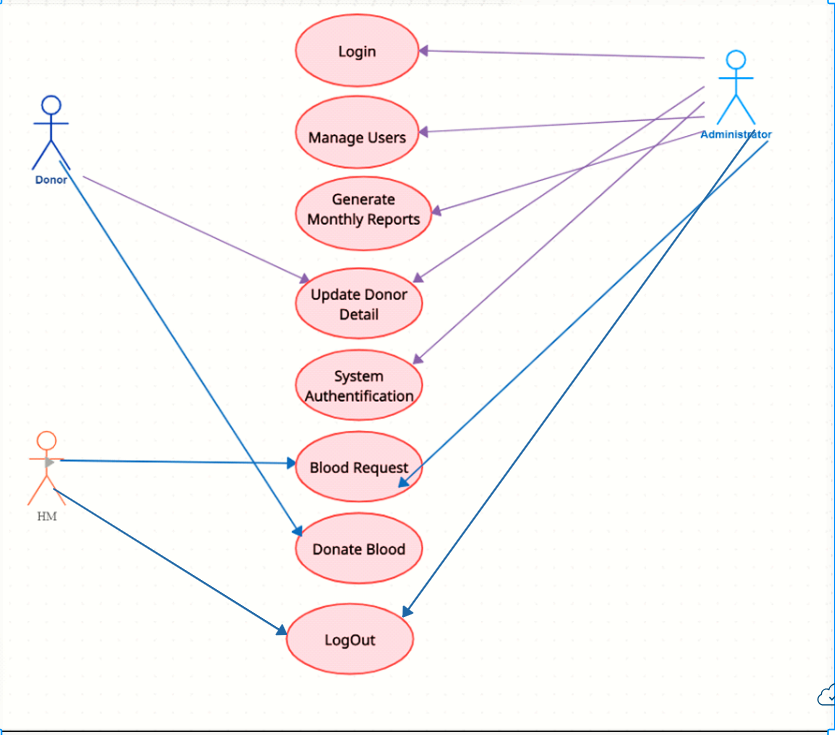


Fig 5.2 Used Case Diagram

**5.3 DATA FLOW DIAGRAM:**

**5.3.1 Level 0:**

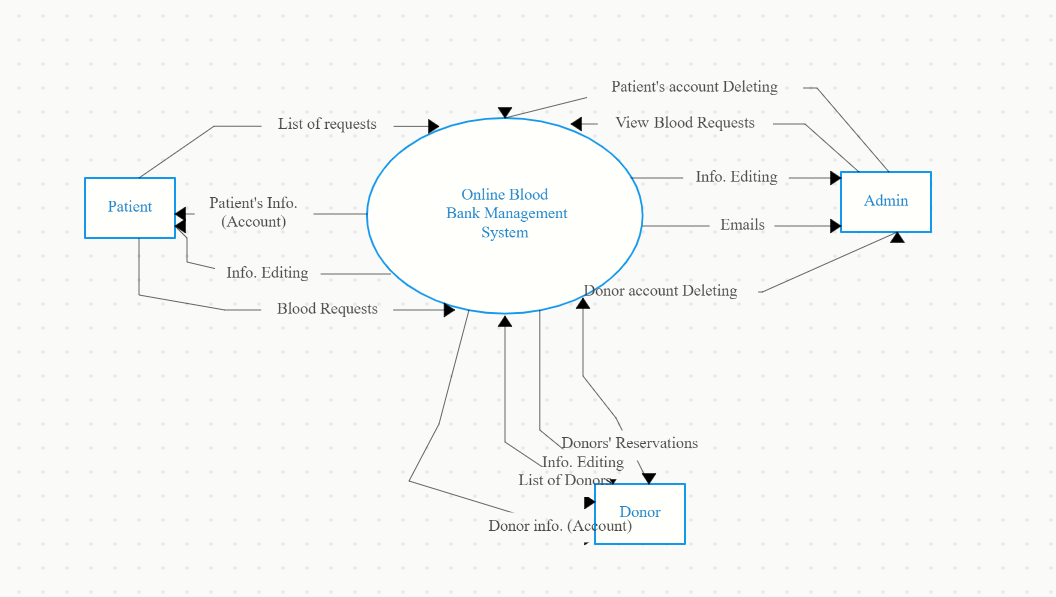
****

Fig 5.3.1 DFD (Level 0)

**5.3.2 Level 1:**

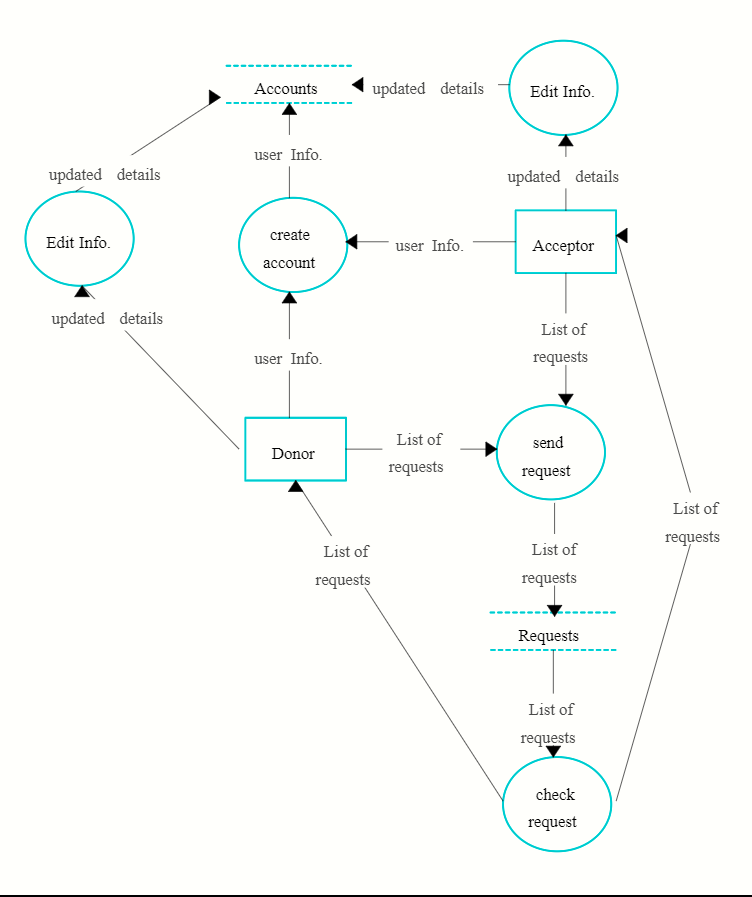
****

Fig 5.3.2 DFD (Level 1)

**5.3.3 Level 2:**

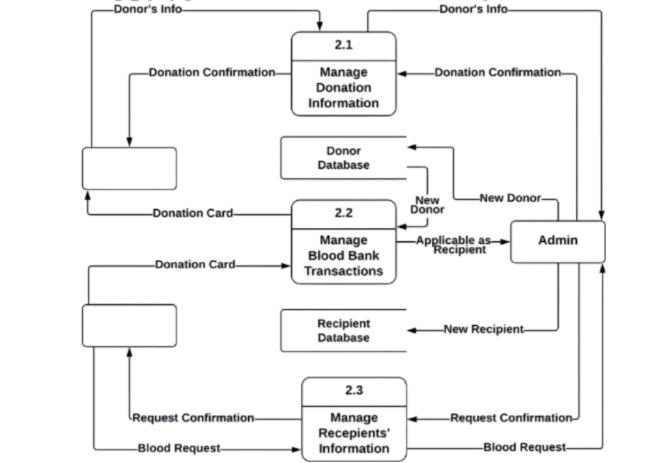
****

Fig 5.3.3 DFD (level 2)

**5.4 ACTIVITY DIAGRAM**

An event is created as an activity diagram encompassing a group of nodes associated with edges. To model the behavior of activities, they can be attached to any modeling element. It can model use cases, classes, interfaces, components, and collaborations. It mainly models processes and workflows. It envisions the dynamic behavior of the system as well as constructs a runnable system that incorporates forward and reverse engineering. It does not include the message part, which means message flow is not represented in an activity diagram.

****

Fig 5.4 Activity Diagram

**CHAPTER 6**

**DESIGN**

**Table 6.1 newDonor**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Data** **type** | **Length/size** | **Description** |
| D\_id | identity | 1,1 | Donor Id (Primary Key ,Not Null) |
| D\_name | varchar | 250 | Donor name (Not Null) |
| F\_name | varchar | 250 | Donor’s Father’s name (Not Null) |
| M\_name | varchar | 250 | Donor’s Mother’s name (Not Null) |
| DOB | varchar | 50 | Date Of Birth (Not Null) |
| Mobile | bigint | - | Donor’s Mobile Number (Not Null) |
| Gender | varchar | 25 | Donor’s Gender (Not Null) |
| Email | varchar | 250 | Donor’s Email (No Null) |
| Blood group | varchar | 250 | Donor’s Blood Group (No Null) |
| City | varchar | 150 | Donor’s City (No Null) |
| D\_Address | varchar | 550 | Donor’s Address (No Null) |

**6.2 LOGIN PAGE**

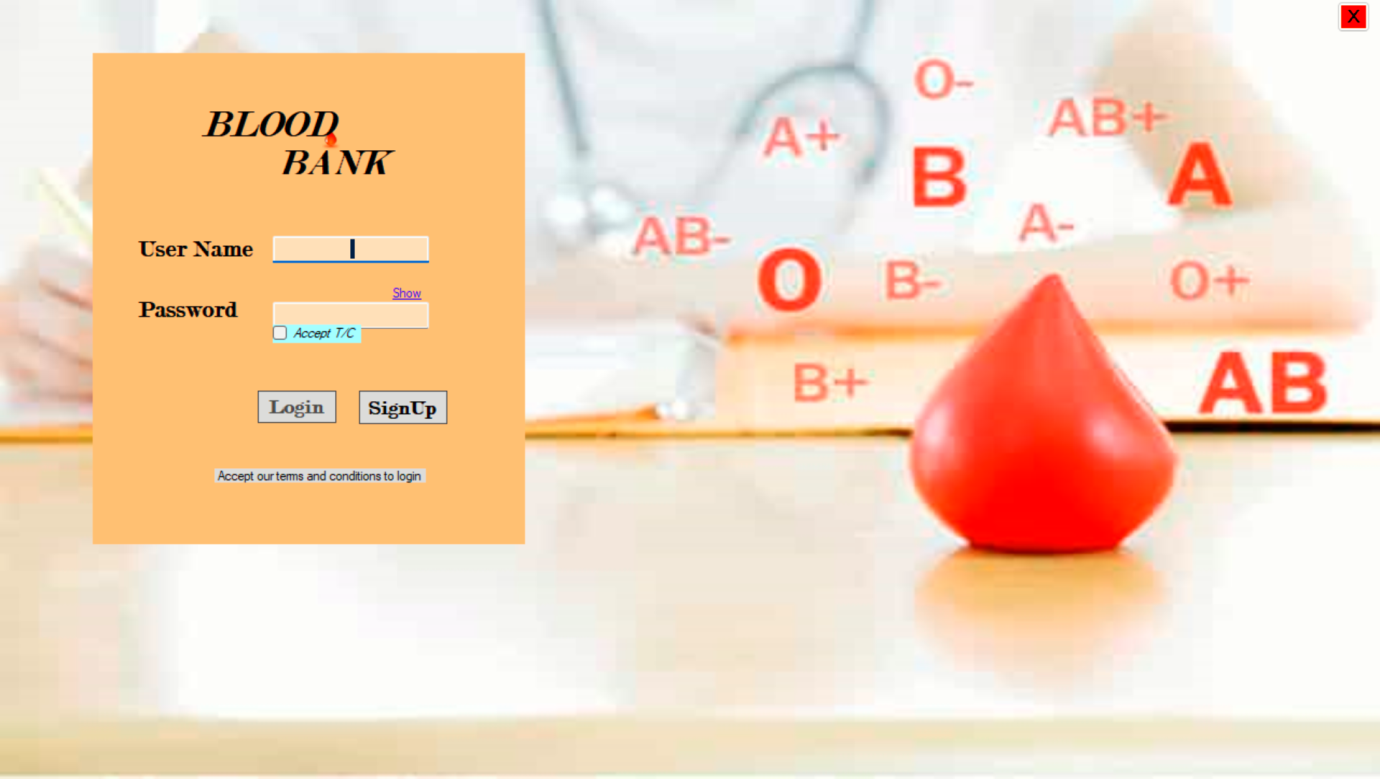


Fig 6.2 Login Page

**6.3 Dashboard**

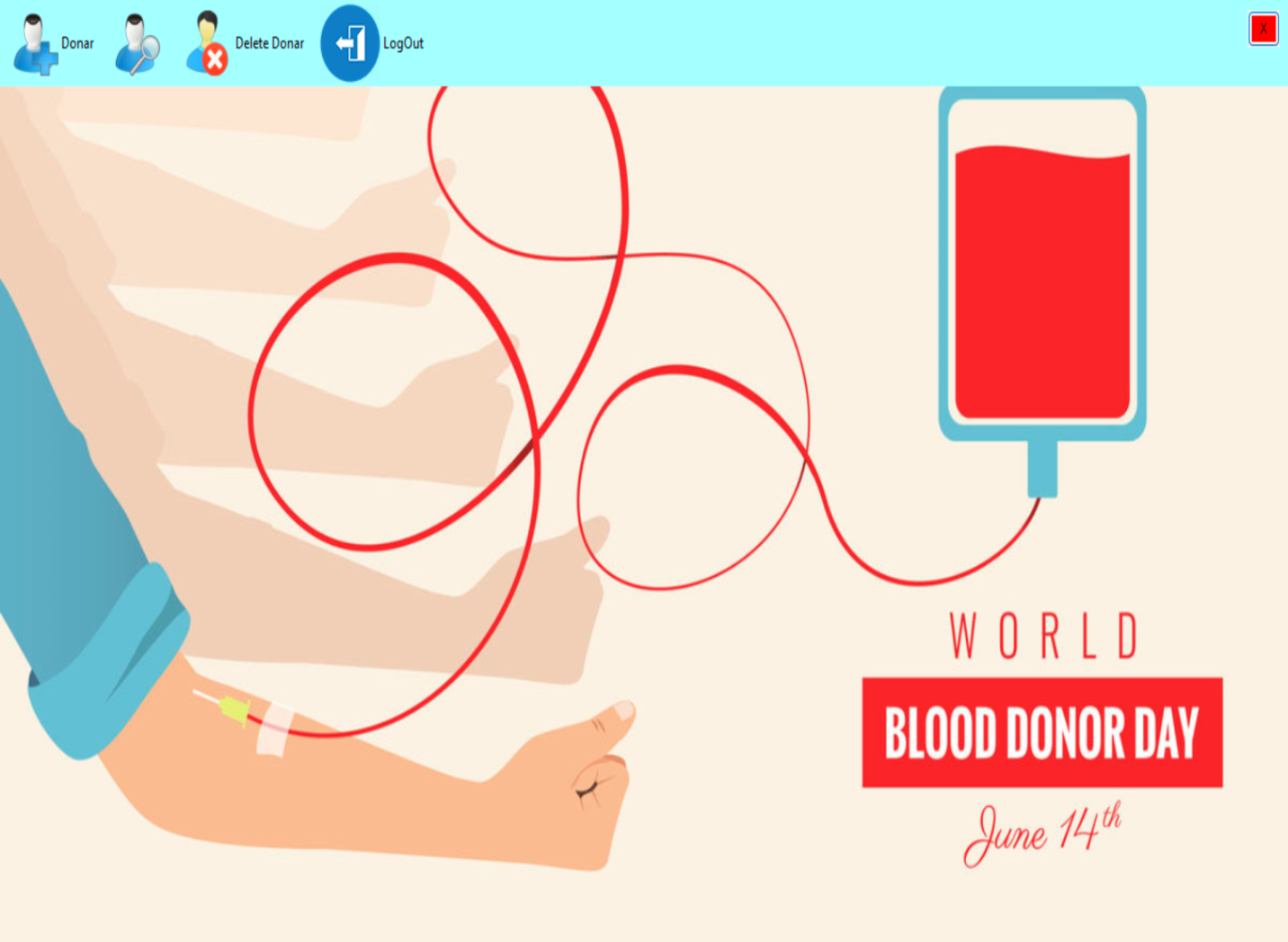


Fig 6.3 Dashboard

6.4 DONOR

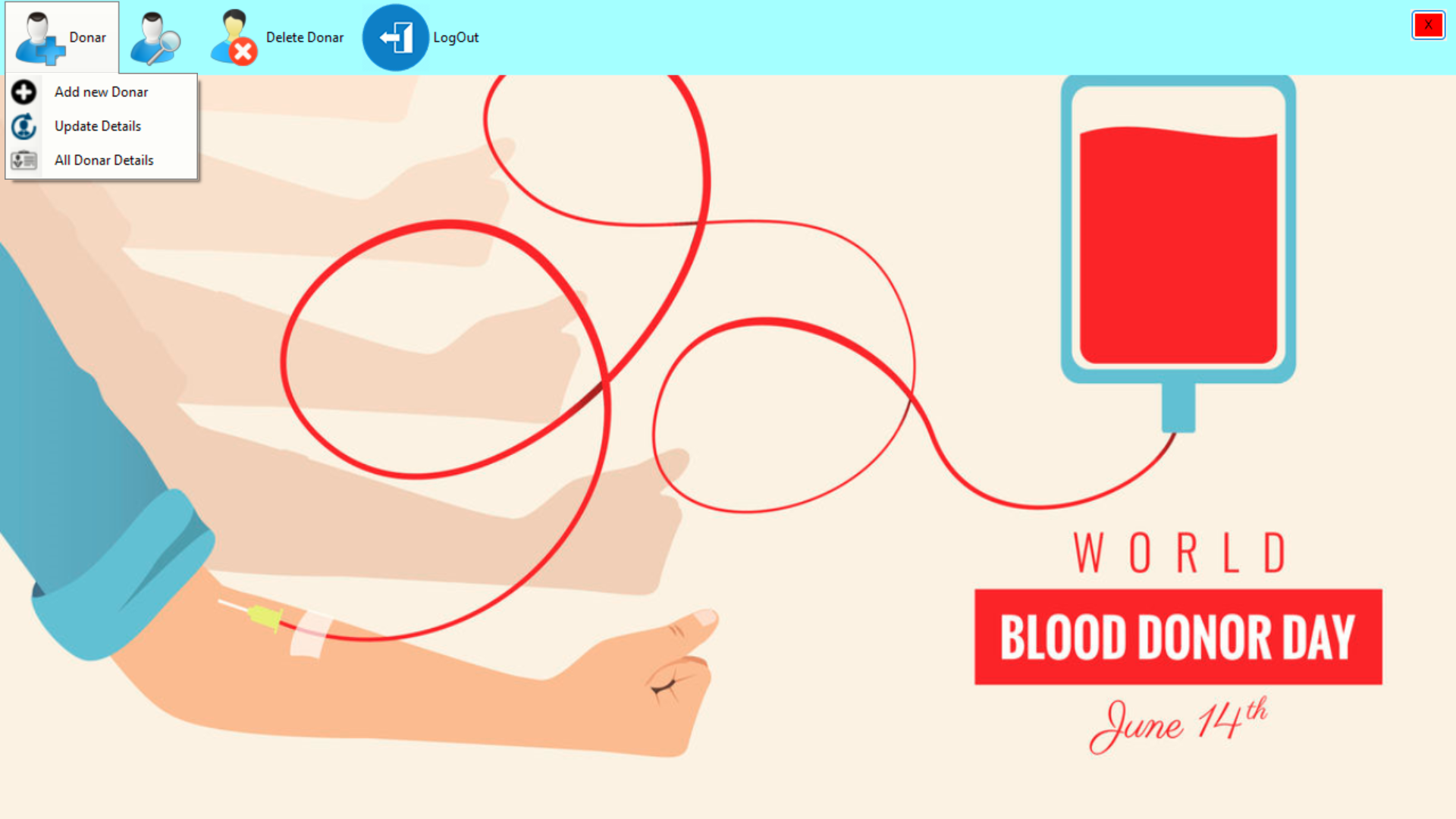


Fig 6.4 Donor

**6.5 ADD NEW DONOR**

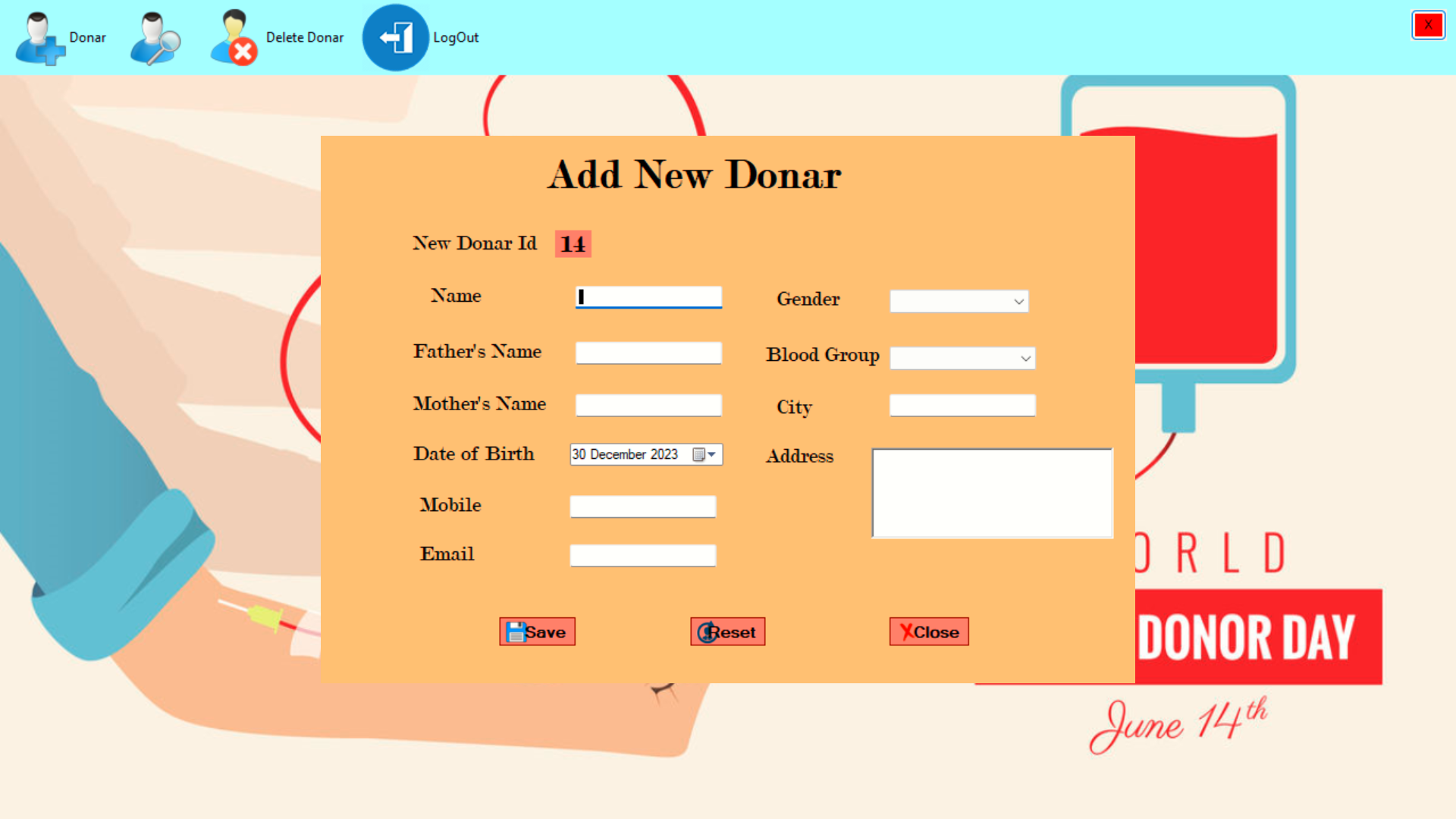


Fig 6.5 Add new Donor

**6.6 UPDATE DETAILS**



Fig 6.6 Update Details

**6.7 ALL DETAIL**



Fig 6.7 All Details

6.8 SEARCH DONOR (ADDRESS)

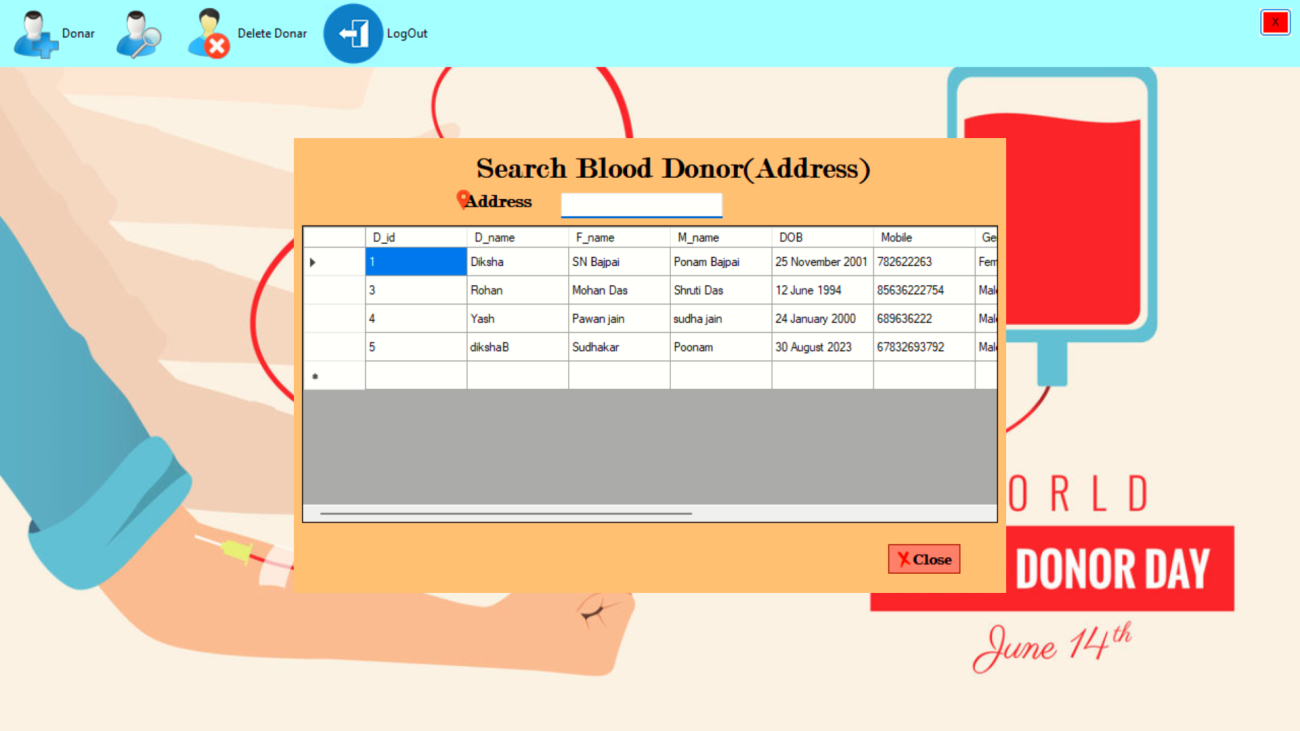


Fig 6.8 Search Donor (Address)

6.9 SEARCH DONOR (BLOOD GROUP)

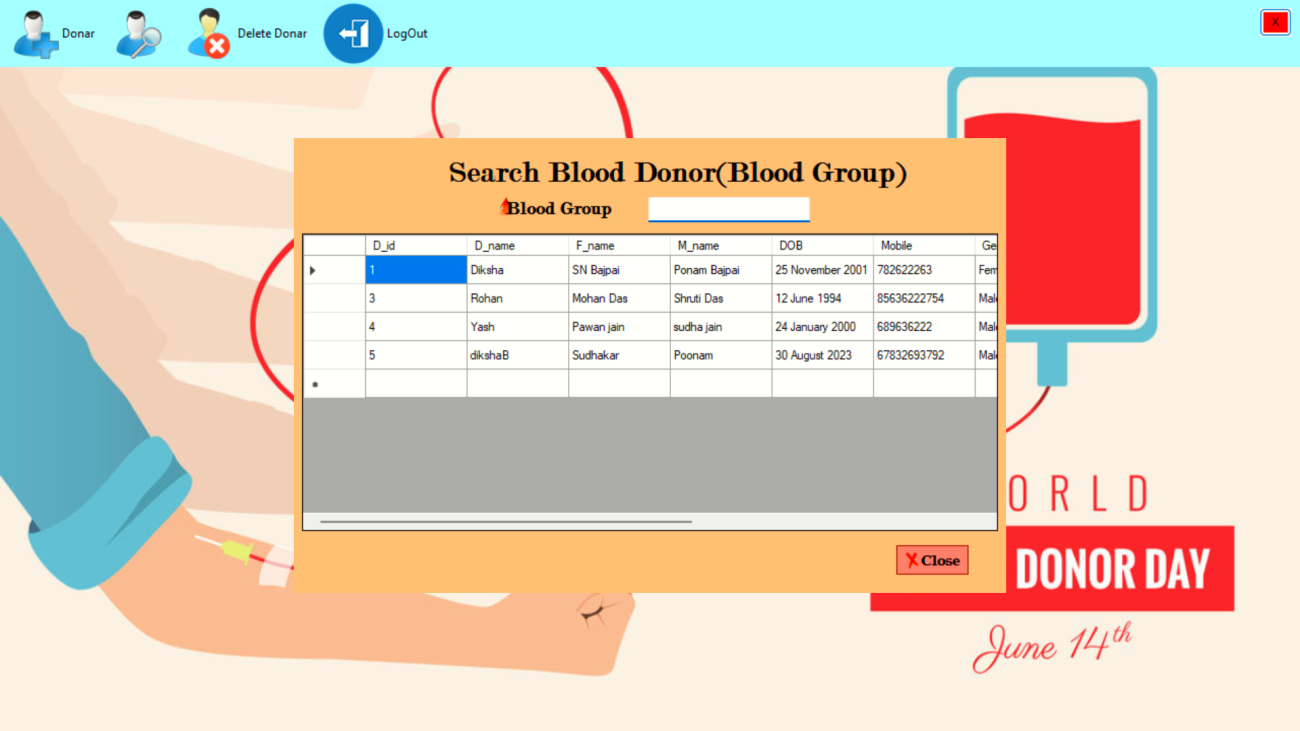


Fig 6.9 Search Donor (Blood Group)

6.10 DELETE DONOR

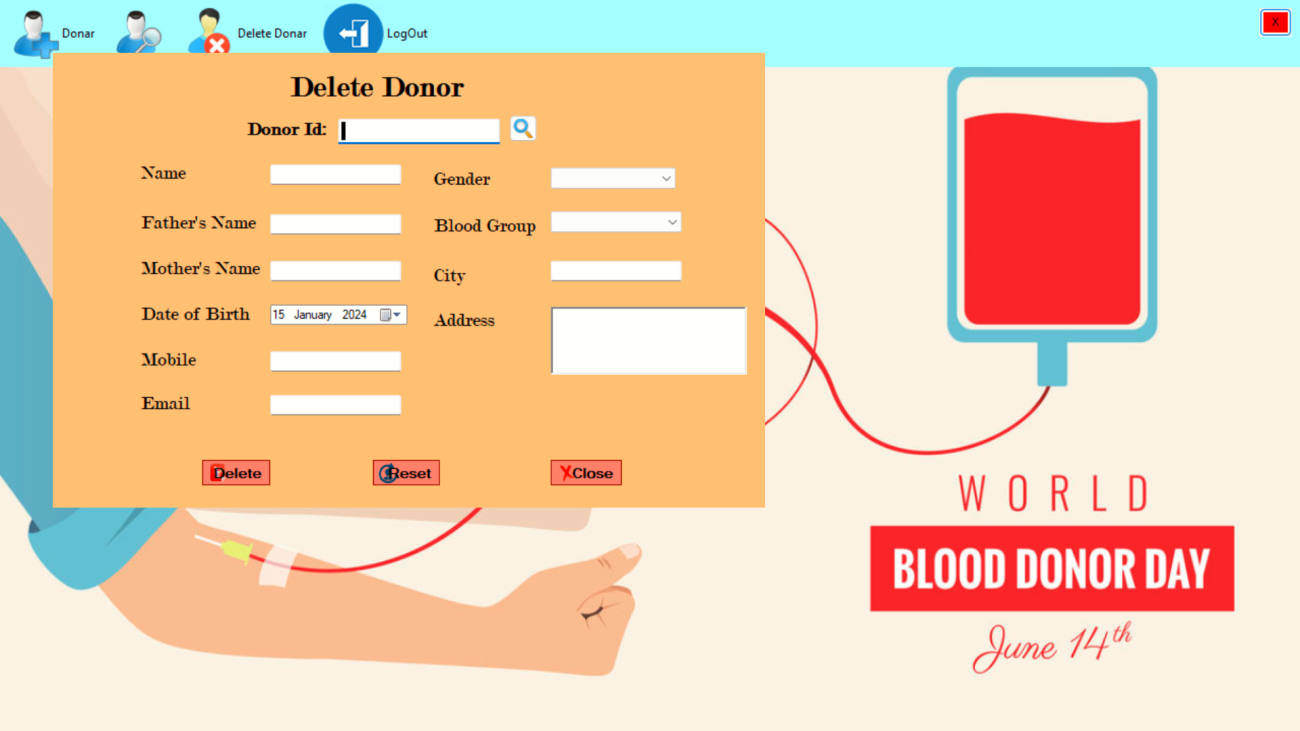


Fig 6.10 Delete donor

**CHAPTER 7**

**SOURCE CODE**

7.1 LOGIN PAGE

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BloodBankManagement

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void Showlb\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

if (Showlb.Text == "Show")

{

Showlb.Text = "Hide";

textBox2.PasswordChar = '\0';

}

else

{

Showlb.Text = " Show";

textBox2.PasswordChar = '\*';

}

}

private void checkBox\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox.Checked == true)

{

Loginbtn.Enabled = true;

}

else

{

Loginbtn.Enabled = false;

MessageBox.Show("Please accept terms and conditions");

}

}

private void Form1\_Load(object sender, EventArgs e)

{

Loginbtn.Enabled = false;

}

private void Loginbtn\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "diksha" && textBox2.Text == "pass" || textBox1.Text == "Admin" && textBox2.Text == "12345")

{

Dashboard db = new Dashboard();

db.Show();

this.Hide();

}

else

{

MessageBox.Show("Enter valid username or password", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

}

private void signupbtn\_Click(object sender, EventArgs e)

{

SignUp sn = new SignUp();

sn.Show();

this.Hide();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

7.2 DASHBOARD

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BloodBankManagement

{

public partial class Dashboard : Form

{

public Dashboard()

{

InitializeComponent();

}

private void button1\_Click\_1(object sender, EventArgs e)

{

Application.Exit();

}

private void bloodGrouToolStripMenuItem\_Click(object sender, EventArgs e)

{

SearchDonor\_byBloodGroup bg = new SearchDonor\_byBloodGroup();

bg.Show();

}

private void logOutToolStripMenuItem\_Click(object sender, EventArgs e)

{

Form1 fm = new Form1();

fm.Show();

this.Hide();

}

private void addNewDonarToolStripMenuItem\_Click(object sender, EventArgs e)

{

Add\_new\_Donar ad = new Add\_new\_Donar();

ad.Show();

}

private void updateDetailsToolStripMenuItem\_Click(object sender, EventArgs e)

{

Update udd = new Update();

udd.Show();

}

private void allDonarDetailsToolStripMenuItem\_Click(object sender, EventArgs e)

{

All\_Details Ad = new All\_Details();

Ad.Show();

}

private void locationToolStripMenuItem\_Click(object sender, EventArgs e)

{

SearchBloodDonor sb = new SearchBloodDonor();

sb.Show();

}

private void deleteDonarToolStripMenuItem1\_Click(object sender, EventArgs e)

{

DeleteDonor dd = new DeleteDonor();

dd.Show();

}

}

}

7.3 ADD DONOR

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Text.RegularExpressions;

using System.Diagnostics.Eventing.Reader;

using System.Xml.Linq;

namespace BloodBankManagement

{

public partial class Add\_new\_Donar : Form

{

function fn = new function();

//string emailpattern = "^[a - z0 - 9!#$%&'\*+/=?^\_`{|}~-]+(?:\\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*@(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?$";

public object Integer { get; private set; }

public Add\_new\_Donar()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void Add\_new\_Donar\_Load(object sender, EventArgs e)

{

String query = "Select max(D\_id) from newDonor";

DataSet ds = fn.getData(query);

int count = int.Parse(ds.Tables[0].Rows[0][0].ToString());

lbId.Text = (count +1).ToString();

}

private void btnSave\_Click(object sender, EventArgs e)

{

if (txtName.Text != "" && txtfName.Text != "" && txtMname.Text != "" && txtEmail.Text != "" && txtAddress.Text != "" && cbGender.Text != "" && cbBloodGroup.Text != "" && txtCity.Text != "" && txtMobile.Text != "")

{

String Dname = txtName.Text;

String Fname = txtfName.Text;

String Mname = txtMname.Text;

String Email = txtEmail.Text;

String Address = txtAddress.Text;

String Gender = cbGender.Text;

String BloodGroup = cbBloodGroup.Text;

String City = txtCity.Text;

Int64 Mobile =Int64.Parse(txtMobile.Text);

String DOB = dateTimePicker1.Text;

String query = "Insert into newDonor(D\_name,F\_name,M\_name,DOB,Mobile,Gender,Email,BloodGroup,City,D\_Address) values('"+Dname+"','"+Fname+"','"+Mname+ "','"+DOB+ "','"+Mobile+ "','"+Gender+"','"+Email+"','"+BloodGroup+"','"+City+ "','"+Address+"')";

fn.setData(query);

}

else

{

MessageBox.Show("Fill all the Details", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void txtMobile\_TextChanged(object sender, EventArgs e)

{

if(txtMobile.TextLength == 10)

{

txtMobile.ForeColor = Color.Black;

}

else

{

txtMobile.ForeColor = Color.Red;

}

}

private void txtMobile\_KeyPress(object sender, KeyPressEventArgs e)

{

if(!char.IsDigit(e.KeyChar)&& !char.IsControl(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Error, A Phone number cannot contain letters");

}

}

private void txtEmail\_TextChanged(object sender, EventArgs e)

{}private void txtEmail\_Leave(object sender, EventArgs e)

{} private void txtEmail\_KeyPress(object sender, KeyPressEventArgs e)

{}

private void txtEmail\_Validating(object sender, CancelEventArgs e)

{

System.Text.RegularExpressions.Regex rEmail = new System.Text.RegularExpressions.Regex(@"^[a-zA-Z][\w\.-]{2,28}[a-zA-Z0-9]@[a-zA-Z0-9][\w\.-]\*[a-zA-Z0-9]\.[a-zA-Z][a-zA-Z\.]\*[a-zA-Z]$");

if(txtEmail.Text.Length>0)

{

if (!rEmail.IsMatch(txtEmail.Text))

{

MessageBox.Show("Enter a valid Email Id","Error",MessageBoxButtons.OK, MessageBoxIcon.Error);

txtEmail.SelectAll();

e.Cancel = true;

}

}

}

private void btnReset\_Click(object sender, EventArgs e)

{

txtName.Clear();

txtfName.Clear();

txtMname.Clear();

txtEmail.Clear();

txtAddress.Clear();

cbGender.Text = string.Empty;

cbBloodGroup.Text = string.Empty;

txtCity.Clear();

txtMobile.Clear();

}

private void txtName\_Validating(object sender, CancelEventArgs e)

{

if (string.IsNullOrEmpty(Name)) {

Console.WriteLine("Name can't be empty!");

}

}

}

}

7.4 DELETE DONOR

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace BloodBankManagement

{

public partial class DeleteDonor : Form

{

function fn = new function();

public DeleteDonor()

{

InitializeComponent();

}

private void txtDonorId\_TextChanged(object sender, EventArgs e)

{

if(txtDonorId.Text == ""){

txtName.Clear();

txtfName.Clear();

txtMname.Clear();

dateTimePicker1.ResetText();

txtMobile.Clear();

cbGender.ResetText();

txtEmail.Clear();

cbBloodGroup.ResetText();

txtCity.Clear();

txtAddress.Clear();

}

}

private void btnClose\_Click(object sender, EventArgs e)

{

this.Close();

}

private void btnSearch\_Click(object sender, EventArgs e)

{

if (txtDonorId.Text != "") {

String query = "select \* from newDonor where D\_id = '" + txtDonorId.Text + "'";

DataSet ds = fn.getData(query);

if (ds.Tables[0].Rows.Count != 0)

{

txtName.Text = ds.Tables[0].Rows[0][1].ToString();

txtfName.Text = ds.Tables[0].Rows[0][2].ToString();

txtMname.Text = ds.Tables[0].Rows[0][3].ToString();

dateTimePicker1.Text = ds.Tables[0].Rows[0][4].ToString();

txtMobile.Text = ds.Tables[0].Rows[0][5].ToString();

cbGender.Text = ds.Tables[0].Rows[0][6].ToString();

txtEmail.Text = ds.Tables[0].Rows[0][7].ToString();

cbBloodGroup.Text = ds.Tables[0].Rows[0][8].ToString();

txtCity.Text = ds.Tables[0].Rows[0][9].ToString();

txtAddress.Text = ds.Tables[0].Rows[0][10].ToString();

}

}

else

{

MessageBox.Show("No record exist", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnDelete\_Click(object sender, EventArgs e)

{

if( MessageBox.Show("Are You Sure", "Delete", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) == DialogResult.OK)

{

String query = "delete from newDonor where D\_id = "+txtDonorId.Text+"";

fn.setData(query);

}

}

private void btnReset\_Click(object sender, EventArgs e)

{

txtDonorId.Clear();

}

}

}

**CHAPTER 8**

**TESTING**

* 1. **DONOR MANAGEMENT:**

**8.1.1 Registration:**

* Verify successful registration with valid and invalid data.
* Test duplicate registration attempts.
* Check if minimum age and weight requirements are enforced.
  + 1. **Medical Screening:**
* Test various scenarios for pre-donation questionnaires and health checks.
* Simulate positive and negative results for different blood tests.
* Verify deferral and disqualification rules are applied correctly.
  + 1. **Blood Donation:**
* Check recording of successful and unsuccessful donations.
* Verify volume calculation and minimum/maximum limits.

**8.2 INVENTORY MANAGEMENT:**

**8.2.1 Blood Product Storage:**

* Test registration and tracking of different blood types and components.
* Verify temperature and expiration date monitoring.
* Simulate blood product spoilage and removal from inventory.
  + 1. **Blood Product Ordering:**
* Test creation and processing of blood orders from hospitals.
* Verify inventory availability and compatibility checks.
* Simulate order fulfillment and dispatch.
  1. **SYSTEM FUNCTIONALITY**:

**8.3.1 User Access Control:**

* Test login and access levels for different user roles (admin, staff, donor).
* Verify secure access to donor and patient information.
  + 1. **Search and Reporting:**
* Test searching for donors based on various criteria.
* Verify generation of reports on blood inventory, donations, and transfusions.
  + 1. **Error Handling:**
* Simulate network connectivity issues and server errors.
* Test system behavior for invalid data input and unexpected scenarios.

**CHAPTER 9**

**CONCLUSION**

**9.1 CONCLUSION**

We have successfully designed and developed a proven website to make it easy for any user or person in need of blood to request the blood they need. Those who wish to donate blood can also register to donate blood on the website. Blood Bank management system will provide an effective way of managing the different types of blood available. It manages all the information about the donors in a systematic way so there is no data redundancy. Web based blood bank management system provides convenience, efficiency and security to the users and blood bank compared to manual system. It was found out that manual system has many disadvantages that disappoint and dissatisfy the users. Indeed, online blood bank management system make work easy, and ensure fast retrieval of data when needed. The system will eliminate all the problems encountered in the manual way which will help the hospital or the blood banks to work on a better way. The system was implemented using web-based technologies which include HTML, CSS, JS, Bootstrap for frontend and for backend I have used MySQL and PHP.

**9.2 REFERENCES**

* Tuan, T. G. Online Blood Donation Reservation and Management System, 2006.
* Development of a Blood Bank Management System Sumazly Sulaimana,\*, Abdul Aziz K.Abdul Hamida , Nurul Ain Najihah Yusria.
* Vikas Kulshreshtha, Dr. Sharad Maheshwari, “Blood Bank Management Information System in India,” in International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 1, Issue 2, pp.260-263
* Ravi Kumar, Shubham Singh, V Anu Ragavi, “Blood Bank Management System,” IJARIIE-ISSN(O)- 2395- 4396, Vol-3 Issue-5 2017
* Blood donor selection. Guidelines on assessing donor suitability for blood donation. Annex 3. Geneva: World Health Organization; 2012. [17 August 2012]. http://www.who.int/bloodsafety/voluntary\_donation/blood\_donor \_selection\_counselling/en/ [PubMed]

**BIBLIOGRAPHY**

Bibliography of Blood Bank Management Systems (BBMS):

Journal Articles:

* Pah Essah, M. S., & Said Ab Rahman, N. A. (2011). A web-based blood bank management system (BBMS). International Journal of Engineering and Technology (IRJET), 8(10), 132-138.
* Rajendra, S. S., & Bhalchandra, S. V. (2019). Online blood bank management system (BBMS). International Journal of Engineering Science and Computing, 9(2), 19849-19851.
* Gupta, D., & Saxena, N. (2015). A comprehensive study of blood bank management system. International Journal of Computer Science and Information Technologies, 6(1), 155-160.
* Kapoor, S., & Saxena, N. (2017). Design and implementation of a web-based blood bank management system. International Journal of Computer Applications, 168(12), 24-29.
* Nair, H. K., & Sahu, S. (2017). A secure and efficient blood bank management system using blockchain technology. International Journal of Computer Science and Information Security, 15(1), 1-8.
* Books and Conference Proceedings:
* Kumar, A. (2012). Blood bank management system: A technical guide. CRC Press.
* Proceedings of the International Conference on Healthcare Informatics (ICHI) (2018-present). (Contains numerous papers on BBMS).
* Proceedings of the Conference on Information Systems in Healthcare (CISH) (2018-present). (Contains numerous papers on BBMS).
* Websites and Online Resources:
* World Health Organization (WHO): Blood Transfusion Safety <https://www.who.int/news-room/fact-sheets/detail/blood-safety-and-availability>
* International Society of Blood Transfusion (ISBT): BBMS Resources <https://www.isbtweb.org/>
* Journal of Blood Management: BBMS Articles <https://pubmed.ncbi.nlm.nih.gov/37839660/>
* This bibliography is not exhaustive and only includes a selection of relevant resources.